ABSTRACT

Title of dissertation: A VISUAL ANALYTICS APPROACH
TO COMPARING COHORTS
OF EVENT SEQUENCES

Sana Malik, Doctor of Philosophy, 2016

Dissertation directed by: Professor Ben Shneiderman
Department of Computer Science

Sequences of timestamped events are currently being generated across nearly every domain of data analytics, from e-commerce web logging to electronic health records used by doctors and medical researchers. Every day, this data type is reviewed by humans who apply statistical tests, hoping to learn everything they can about how these processes work, why they break, and how they can be improved upon.

To further uncover how these processes work the way they do, researchers often compare two groups, or cohorts, of event sequences to find the differences and similarities between outcomes and processes. With temporal event sequence data, this task is complex because of the variety of ways single events and sequences of events can differ between the two cohorts of records: the structure of the event sequences (e.g., event order, co-occurring events, or frequencies of events), the attributes about the events and records (e.g., gender of a patient), or metrics about the timestamps themselves (e.g., duration of an event). Running statistical tests to cover all these

cases and determining which results are significant becomes cumbersome.

Current visual analytics tools for comparing groups of event sequences emphasize a purely statistical or purely visual approach for comparison. Visual analytics tools leverage humans' ability to easily see patterns and anomalies that they were not expecting, but is limited by uncertainty in findings. Statistical tools emphasize finding significant differences in the data, but often requires researchers have a concrete question and doesn't facilitate more general exploration of the data.

Combining visual analytics tools with statistical methods leverages the benefits of both approaches for quicker and easier insight discovery. Integrating statistics into a visualization tool presents many challenges on the frontend (e.g., displaying the results of many different metrics concisely) and in the backend (e.g., scalability challenges with running various metrics on multi-dimensional data at once). I begin by exploring the problem of comparing cohorts of event sequences and understanding the questions that analysts commonly ask in this task. From there, I demonstrate that combining automated statistics with an interactive user interface amplifies the benefits of both types of tools, thereby enabling analysts to conduct quicker and easier data exploration, hypothesis generation, and insight discovery. The direct contributions of this dissertation are: (1) a taxonomy of metrics for comparing cohorts of temporal event sequences, (2) a statistical framework for exploratory data analysis with a method I refer to as high-volume hypothesis testing (HVHT), (3) a family of visualizations and guidelines for interaction techniques that are useful for understanding and parsing the results, and (4) a user study, five long-term case studies, and five short-term case studies which demonstrate the utility and impact of these methods in various domains: four in the medical domain, one in web log analysis, two in education, and one each in social networks, sports analytics, and security.

My dissertation contributes an understanding of how cohorts of temporal event sequences are commonly compared and the difficulties associated with applying and parsing the results of these metrics. It also contributes a set of visualizations, algorithms, and design guidelines for balancing automated statistics with user-driven analysis to guide users to significant, distinguishing features between cohorts. This work opens avenues for future research in comparing two or more groups of temporal event sequences, opening traditional machine learning and data mining techniques to user interaction, and extending the principles found in this dissertation to data types beyond temporal event sequences.

A VISUAL ANALYTICS APPROACH
TO COMPARING COHORTS
OF EVENT SEQUENCES

by

Sana Malik

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Ben Shneiderman, Chair/Advisor
Dr. Catherine Plaisant, Co-Advisor
Professor Margret Bjarnadottir
Professor Hector Corrada-Bravo
Professor Niklas Elmqvist

Dedication

To Zaan, Mahum, Ismael, and Humza

Acknowledgments

I love too many people.

❤️ ❤️ ❤️ ❤️ ❤️ ❤️ ❤️ ❤️ ❤️ ❤️ ❤️

I would first like to thank my advisors, Dr. Ben Shneiderman and Dr. Catherine Plaisant, for their continued support throughout the last three years. Thank you, Ben, for your endless patience and optimism – I could not have asked for a more positive and encouraging advisor. Catherine, thank you for always being practical, available, and willing to provide feedback on everything I've asked. I've learned so much from you both, not just about research, but about being part of a team and always remaining positive.

I'd also like to thank the members my proposal and dissertation committees who made my work considerably stronger: Hector Corrado Bravo, Margret Bjarnadottir, Niklas Elmqvist, and Alan Sussman. Thank you for providing feedback throughout the entirety of my research. Thank you also to all my case study partners for putting up with countless bugs, usability issues, and confusing errors and still providing valuable feedback: Rachel Webman, Randall Burd, Leah MacFadyen, Eberechukwi Onukwugha, Jim Gardner, Eunyee Koh, and Sean Barnes.

I would like to thank Fan Du, for being a wonderfully reliable collaborator and friend; I am so glad I've had you by my side for the past two years. Megan Monroe, Cody Dunne, and John Alexis Guerra-Gomez: thank you for your guidance and mentorship. I'll always look up to you! I am so grateful for the entire HCIL and each of its members. Thank you for always being a bright, friendly place where ideas come to grow and practice talk standards are unreal.

I don't even know how to begin this next group. The past five years are when I've found "my people." I'm so grateful for all the friends I've made, for countless game nights where we spent more time learning the rules than playing the game, for trivia Thursdays, and for always teaching me new things. Philip and Robin (and Ada) Dasler, Cody Buntain, Leigh Cook, Steve Bach, Matt Mauriello, Jay Pujara, Alex Malozemoff. You guys are the best. Brenna McNally. I didn't include you in the previous list, because you are too special (no offense, everyone else). Thank you for doing too much for me. For baby-sitting me and forcing me to rehearse talks and write when I did. not. want. to. For cleaning the apartment without me noticing and for having enough energy to spare some of yours for me. And lastly, thank you, Steven Lee, for always seeing the silver lining and for always encouraging me.

To the CCL: Anam A., Amina, Annya, Asema, and Anam R.: who would I even be without you? Thanks for accepting me even though my name doesn't begin with an A and for countless birthday dinners, text messages, and road trips over the past 10 years.

Lastly, I'd like to thank my family. My parents, for being the hardest working people I know and my siblings for never letting me feel alone.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Sequences of timestamped events are currently being generated across nearly every domain of data analytics. Consider a typical e-commerce site tracking each of its users through a series of search results and product pages until a purchase is made. Or consider a database of electronic health records containing the symptoms, medications, and outcomes of each patient who is treated. Every day, this data type is reviewed by humans who apply statistical tests, hoping to learn everything they can about how these processes work, why they break, and how they can be improved upon.

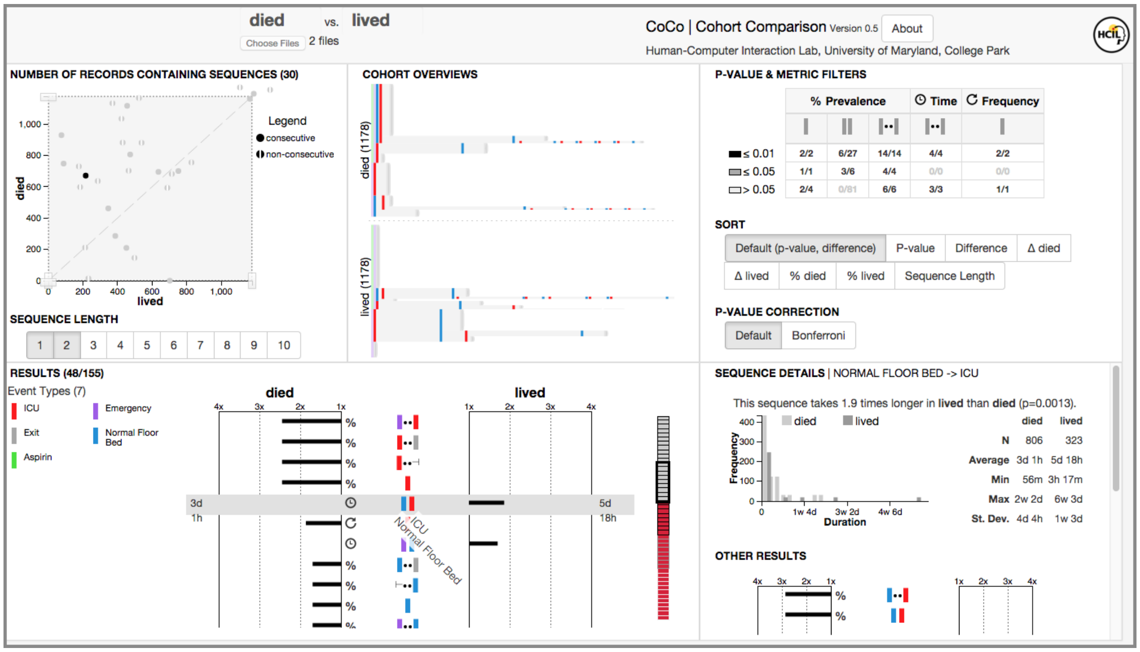
Human eyes and statistical tests, however, reveal very different things. Statistical tests show metrics, uncertainty, and statistical significance. Human eyes see context, confirm what they already know, and discover patterns that are unexpected.

Visualization tools strive to capitalize on these latter, human strengths. For example, the EventFlow visualization tool [1] supports exploratory, visual analyses over large datasets of temporal event sequences. This support for open-ended exploration, however, comes at a cost. The more that a visual analytics tool is designed around open-ended questions and flexible data exploration, the less it is able to effectively integrate automated, statistical analysis. Automated statistics can provide answers, but only when the questions are known.

The opportunity to combine these two approaches lies in the middle ground. By all accounts, the goal of open-ended questions is to generate more concrete questions. As these questions come into focus, so too does the ability to automatically generate the answers. I introduce a visual analytics tool, CoCo (for “Cohort Comparison”, Figure 1.1), that is designed to capitalize on one such scenario.

Consider again the information that is tracked on an e-commerce site. From a business perspective, the users of the site fall into one of two groups: people who bought something and people who did not. If the goal is to convert more of the latter into the former, it is critical to understand how these two groups, or cohorts, are different. Did one group look at more product pages? Or spend more time on the site? Or have some clear demographic identifier such as gender, race, or age? Similar questions arise in the medical domain as well. Which patients responded well to an experimental medication? How did their treatment patterns differ from the patients who received the standard treatment?

Although comparing two groups of data is a common task, with temporal event sequence data in particular, the task of running many statistical tests becomes complex because of the variety of ways the cohorts, sequences, and events can differ. In addition to the structure of the event sequences (e.g., order, co-occurrences, or frequencies of events), the attributes about the events and records (e.g., gender of a patient), and the timestamps themselves (e.g., an event’s duration) can be distinguishing features between the cohorts. For this reason, running statistical tests to cover all these cases and determining which results are significant becomes



	% Prevalence			◷ Time	↻ Frequency
	|	||	|••|	|••|	|
■ ≤ 0.01	2/2	6/27	14/14	4/4	2/2
▨ ≤ 0.05	1/1	3/6	4/4	0/0	0/0
□ > 0.05	2/4	0/81	6/6	3/3	1/1

This sequence takes 1.9 times longer in **lived** than **died** (p=0.0013).

	died	lived
N	806	323
Average	3d 1h	5d 18h
Min	56m	3h 17m
Max	2w 2d	6w 3d
St. Dev.	4d 4h	1w 3d

Figure 1.1: Two datasets, each containing about a thousand patients as they are transferred throughout a hospital, are being compared using CoCo: patients who lived and patients who died (demo dataset; no real data). Along the top are high-level overviews of each dataset: a scatterplot displaying the sequences in the dataset and how often they occur and each cohort is visualized as an EventFlow graph.The bottom panel displays a rich compact view of the results of high-volume hypothesis testing, ranked by significance with a legend pairing each event with a color. To the right of the list, details-on-demand for a selected hypothesis (comparing the average timing between the blue and red event) provides more details and context for the results. A set of control panels (top right panel) allows analysts to sort and filter the results by event sequence length, event types, sample size, significance, or metric.

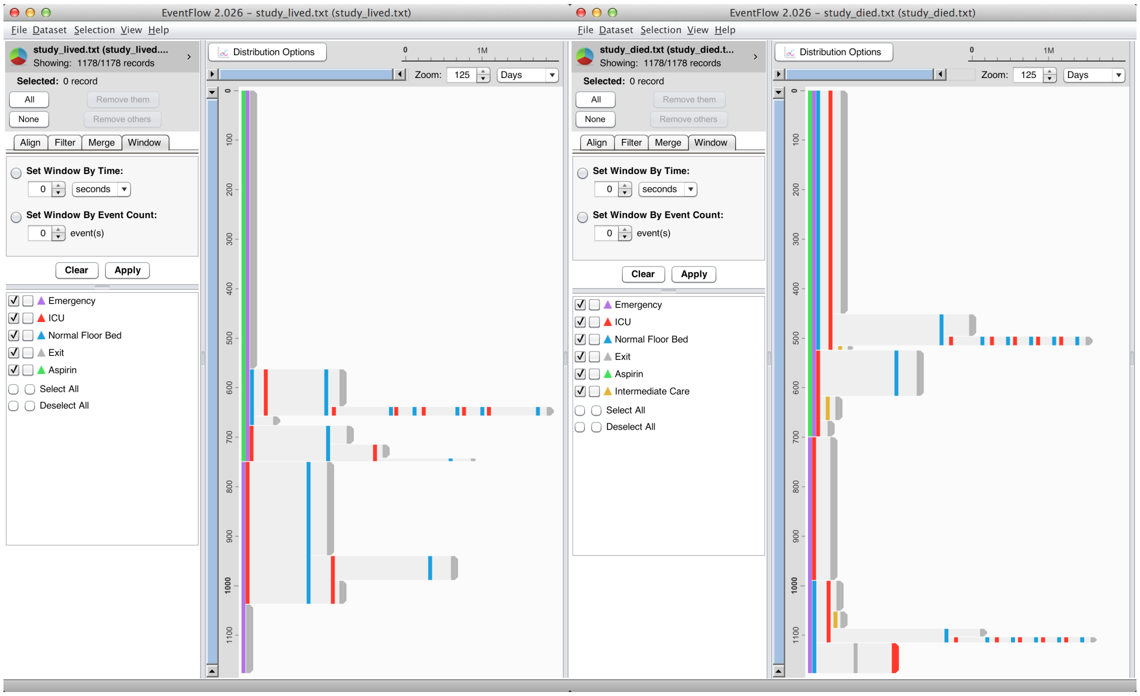


Figure 1.2: Current approaches to comparing event sequences involve putting two separate windows side-by-side for a visual comparison.

cumbersome. Based on three years of case studies, I present a taxonomy of metrics for comparing cohorts of event sequences. Additionally, the factor on which the cohorts are formed may call for different types of questions to be asked about the data. For example, in a set of medical records split by date (e.g., last month's trials vs. this month's), a researcher may be interested in how outcomes for the patients differ between the cohorts, whereas a dataset split by the patient's outcome (e.g., patients who die vs. those who live) would ignore such a metric.

Current tools for cohort comparison of temporal event data (Section 2) emphasize one of two strategies: 1) purely visual comparisons between groups (Figure 1.2), with no integrated statistics, or 2) purely statistical comparisons over one or more features of the dataset. By contrast, CoCo is designed to provide a more balanced integration of both human-driven and automated strategies.

Purely statistical methods of comparison would benefit from user intervention. With the sheer number of metrics, it is time consuming to run every metric ahead of time, especially when not every metric may be required for analysis. Users with domain knowledge about the datasets would ideally be able to select from the metrics and easily eliminate unnecessary metrics. Further, questions asked during cohort comparison may vary based on how the cohorts were divided. If the cohorts were divided by outcome (e.g., patients who lived versus patients who died), the sequence of events leading up to them becomes more important. Analysis might revolve around determining what factors (time or attributes) or events lead to the outcome by determining how the metrics differ between the groups. Conversely, if the cohorts were split based on an event type, questions may revolve around finding distinguishing outcomes (e.g., patients who took Drug A may result in more strokes than patients who took Drug B). Exploration of cohorts that are split by time (e.g., the same patients over two different months) may be more open-ended and require all metrics. The cohorts can be distinguished by time factors, event attributes, or events themselves (sequences of events or outcomes).

Results from purely statistical methods can also be difficult to parse and understand. Analysts may have different priorities and questions, which require different methods for sorting the results. For example, analysts may be interested in *any* difference between the datasets, regardless of the direction of the difference, whereas other analysts may be interested only in results that occur more frequently in Cohort A. Integrated interaction techniques would allow analysts to specify their priorities when viewing results.

The contribution of this thesis is to enable researchers to be far more flexible in examining cohorts and facilitate human intervention where it can save time and effort. Because of the pre-defined problem space of comparing temporal event sequences, analysts can save time by having answers to common questions readily available and giving them a starting point for their exploration. It is important to note that CoCo is intended for exploratory data analysis which will reveal areas of interest to analysts, not as a means of displaying final statistical results, so more complex controls are left for future work. Analysts are expected to conduct follow-up (and more controlled) tests after they have identified possible hypotheses - such as clinical trials in the medical domain or A/B testing in the-commerce domain.

Purely visual tools for temporal event sequences are a good starting point for developing analysis tools for cohort studies, but can be improved by the inclusion of the statistical tests used in automated approaches. For example, EventFlow assumes that each patient record consists of time-stamped point events (e.g. heart attack, vaccination, first occurrence of symptom), temporal interval events (e.g. medication episode, dietary regime, exercise plan), and patient attributes (e.g. gender, age, weight, ethnic background, etc.).

In multiple case studies with EventFlow, the researchers repeatedly observed users visually comparing event patterns in one group of records with those in another group. In simple terms the question was: what are the sequences of events that differentiate one group from the other? A common aspiration is to find clues that lead to new hypotheses about the series of events that lead to particular outcomes, but many other simple questions also involved comparisons. Epidemiologists analyzing the patterns of drug prescriptions [2] tried to compare the patterns of different classes of drugs. Hospital administrators looking at patient journeys through the hospital compared the data of one month with the previous month. Researchers analyzing task performance during trauma resuscitation [3] wanted to compare performance between cases where the response team was alerted of the upcoming arrival of the patient or not alerted. Transportation analysts looking at highway incident responses [4] wanted to compare how an agency handled its incidents differently from another. Their observations suggest that some broad insights can be gained by visually comparing pairs of EventFlow displays (e.g., analysts could see if the patterns were very similar overall between one month and the next) or very different (e.g., a lot more red or the most common patterns were different) but analysts repeatedly expressed the desire for more systematic ways to compare cohorts of records.

My research aims to bridge the gap between statistical and visual analyses to enable more efficient insight discovery and hypothesis generation. With this comes many practical challenges of implementing a high-volume hypothesis testing framework and presenting its result set in an understandable and useful way. On the backend, I consider the scalability of automatically running metrics on complex event sequences. The problem with scalability is two-fold: first, as the number of events grows, the number of possible event sequences grows exponentially. Second, with large numbers of metrics, developers must think about how to efficiently and simultaneously apply many metrics to the dataset at once. On the frontend are considerations with displaying various metrics in a unified way so analysts can understand and parse the results. Additionally, with the sheer number of results, analysts must be given intelligent interaction techniques for parsing, filtering, organizing, and sorting the results.

On a broader level, my dissertation contributes an understanding of how cohorts of temporal event sequences are commonly compared and the difficulties associated with applying and parsing the results of these metrics. It also contributes a set of visualizations, algorithms, and design guidelines for balancing automated statistics with user-driven analysis to guide analysts to significant, distinguishing features between cohorts. This work opens avenues for future research in comparing two or more groups of temporal event sequences, opening traditional machine learning and data mining techniques to user interaction, and extending the principles found in this dissertation to data types beyond temporal event sequences. With the enormous amount of temporal data being collected in medical trials, consumer web logs, and sensor-based technologies, the opportunities for gaining insights are vast. With a tool like CoCo, analysts will be able to improve analysis that will lead to more efficient processes in medical health, business, education, and many other areas.

I begin by showing that the task of cohort comparison is specific enough to support automatic computation against a bounded set of potential questions and objectives, a method I refer to as High-Volume Hypothesis Testing (HVHT). From this starting point, I demonstrate that the diversity of these objectives, both across and within different domains, as well as the inherent complexities of real world datasets, still require human involvement to determine meaningful insights. I explore how visualization and interaction better support the task of exploratory data analysis and understanding HVHT results (how significant they are, why they are meaningful, and whether the entire dataset has been exhaustively explored). Through interviews and case studies with domain experts, I iteratively design and implement visualization and interaction techniques in a visual analytics tool, CoCo, which is used by real-world analysts performing cohort comparison on their own datasets.

1.1 Contributions

The contributions of this dissertation are:

A taxonomy of metrics for comparing cohorts of temporal event sequences. Through a systematic literature review of EventFlow and other case studies, I identified common questions that analysts ask when comparing two or more groups of event sequences and organized these questions in a taxonomy of metrics.

A statistical framework for exploratory data analysis. I implement a subset of the metrics introduced in the taxonomy and identify and solve the major practical challenges of applying thousands of statistical tests, a method I refer to as high-volume hypothesis testing (HVHT),

A family of visualizations and guidelines for interaction techniques. Through an iterative design process with case study partners, I develop and implement visualizations and interaction techniques that are useful for understanding and parsing large sets of hypothesis results.

Evaluations to demonstrate the utility and impact of these methods. I preform three types of evaluation through the development of CoCo:

- a preliminary user study comparing CoCo to EventFlow for the task of cohort comparison,
- six long-term case studies, and
- five short-term case studies.

1.2 Dissertation Organization

This dissertation is organized in the following parts: Chapter 2 discusses related work in event sequence visualization, statistics for comparing cohorts, machine learning techniques for identifying meaningful event sequences, and methods for efficient computation on multi-dimensional data. Chapter 3 discusses the taxonomy of cohort comparison metrics. Chapter 4 discusses the challenges and solutions on the backend for implementing a high-volume hypothesis framework and Chapter 5 discusses the challenges and solutions on the frontend, including a set of visualization design guidelines. Chapter 6 details the evaluation . Chapter 7 concludes the dissertation and discusses avenues for future work.

Chapter 2: Background and Related Work

2.1 Event Sequence Visualization and Comparison

Work on visualization of sequential data is described here in two parts: visualizations of a single group of event sequences and visualizations comparing two or more sequences.

2.1.1 Single Groups

EventFlow [1] (Figure 2.1) and OutFlow [5] (Figure 2.2) visualize a simplified view of collections of event and interval sequences. Both tools aggregate a single cohort and the whole sequences of records. EventFlow allows users to explore the underlying dataset through this visualization. However, they only support visualizing a single group of records, though comparison can be facilitated by using multiple instances of the visualize. In this case however, the tools do not provide statistical information about the differences. CoCo borrows some event icon motifs from EventFlow (such as using colored markers to represent events).

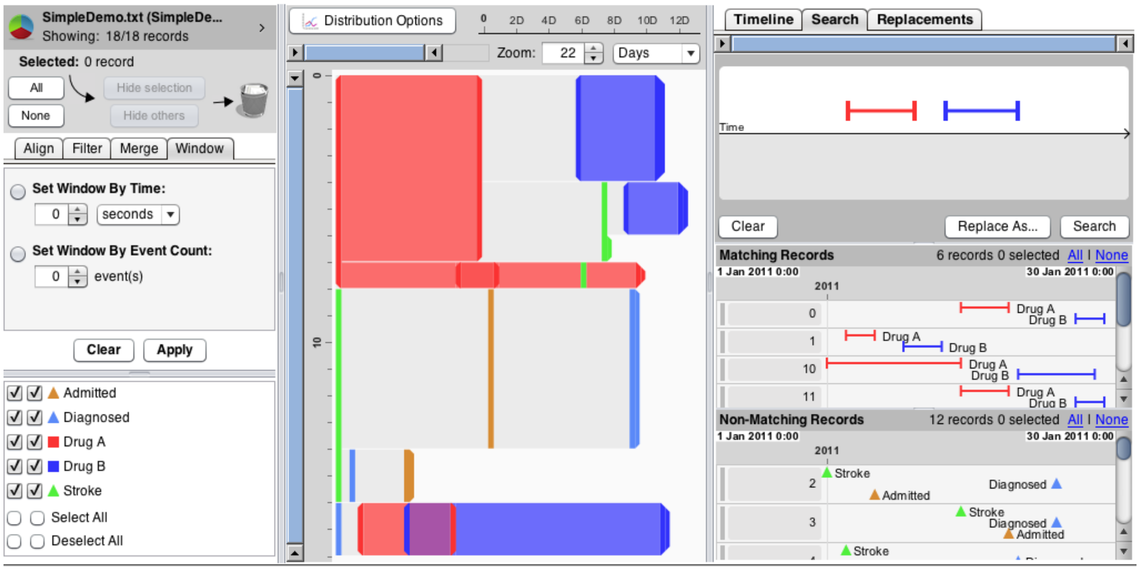


Figure 2.1: EventFlow visualizes an aggregated view of a single group of event sequences. CoCo borrows event icon representations from EventFlow.

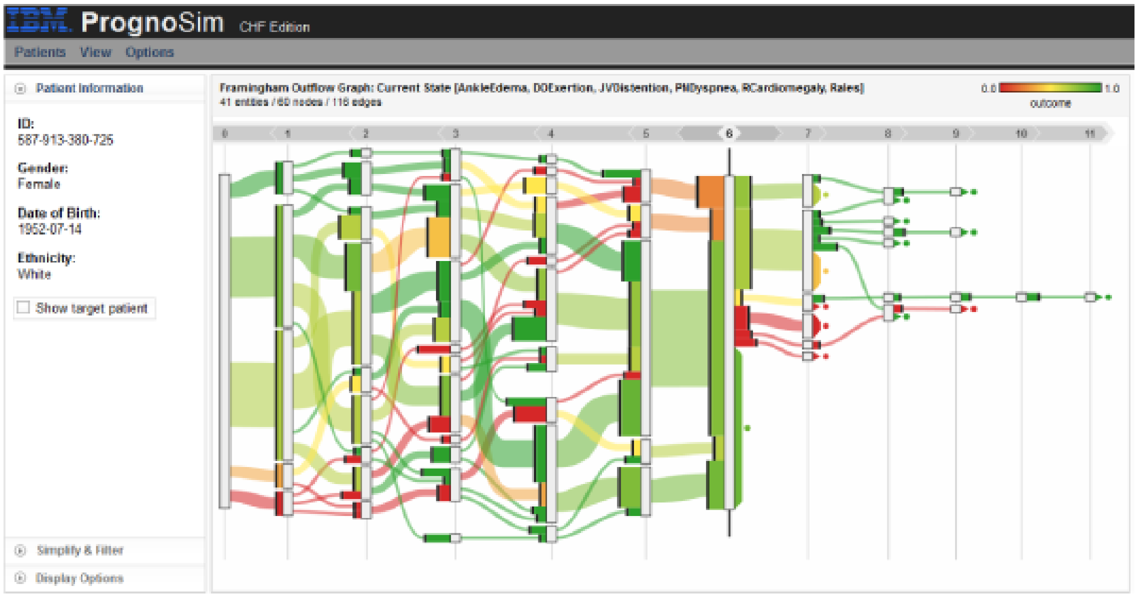


Figure 2.2: Outflow visualizes groups of temporal event sequences for outcome analysis.

2.1.2 Visual Comparison

Gleicher et al. [6] provide an extensive survey of visual comparison techniques classified into three categories and combinations thereof: juxtaposition, superposition, and explicit encoding. This characterization was used as a framework for exploring designs for visualizing comparison results. Though many visualization tools have been designed for event sequence visualization [1, 7] there has been little research on visualizing event sequence comparison until recently. Zhao et al. [8] design MatrixWave, a visualization designed to compare the flow of users in click-stream datasets. MatrixWave focuses on differences in the occurrence of immediate, pairwise steps in the event stream, whereas CoCo generalizes to differences in single events and sequences of any length, as well as differences dealing with time.

Besides finding differences in datasets, event sequence comparison has been explored in the context of finding similarities. Vrotsou et al. [9] introduce a set of event sequence similarity measures. They explore using visualization and interactive data mining to cluster similar groups of event sequences. While CoCo focuses on difference metrics, this work can be extended to applicable similarity measures.

2.1.3 Event Sequence Comparison

Solutions for comparing sequential data have been explored in many different fields, including comparative genomics, text mining, and tree comparison. They are discussed here in the context of event history data and discrete-time models [10].

I draw first on methods to compare collections of general sequences without

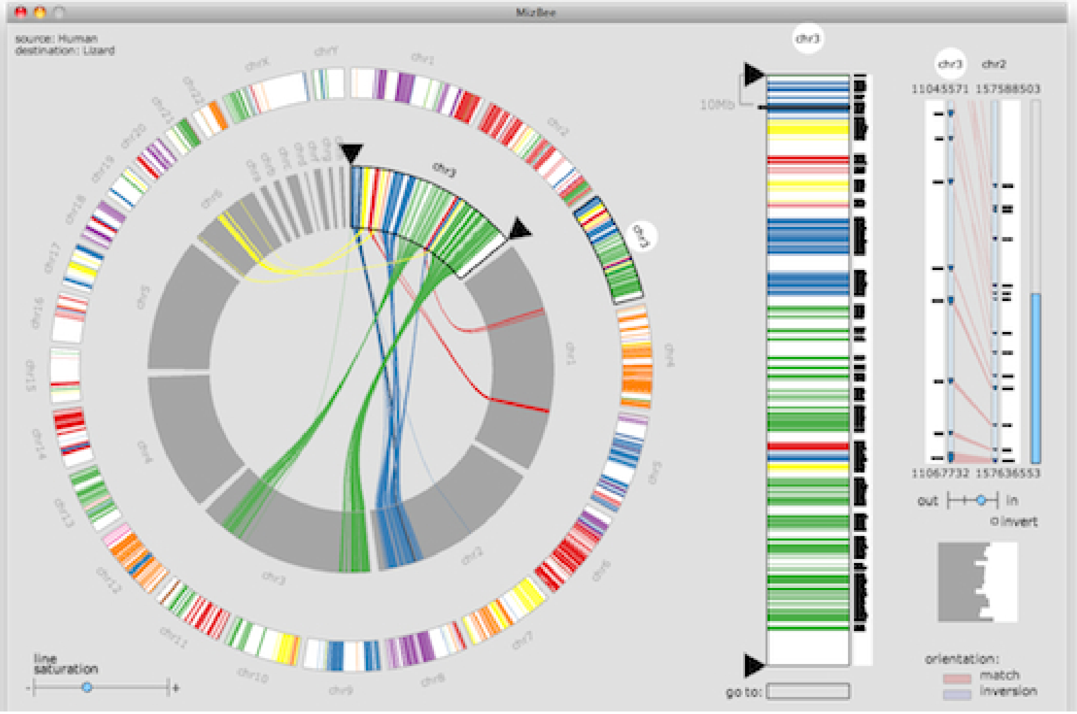


Figure 2.3: MizBee measures the similarity between genomes by visualizing regions of shared sequences.

the notion of time, most notably the fields of comparative genomics and text mining, where the data is ordered with respect to some index [11].

Genome browsers [12–17] have been developed to visualize genome sequences. They compare genomes by visualizing the position of each nucleotide, and consider a genome as a long and linear sequence of nucleotides. Scientists also compare genomes at the gene level. However, most of the existing tools are only able to compare either only the similarities or only the differences of collections of gene sequences. For example, MizBee [18] (Figure 2.3) measures the similarity between genomes by visualizing the regions of shared sequences. Variant View [19] (Figure 2.4), cBio [20] and MuSiC [21] only support displaying sequence variants. Further, genome sequences

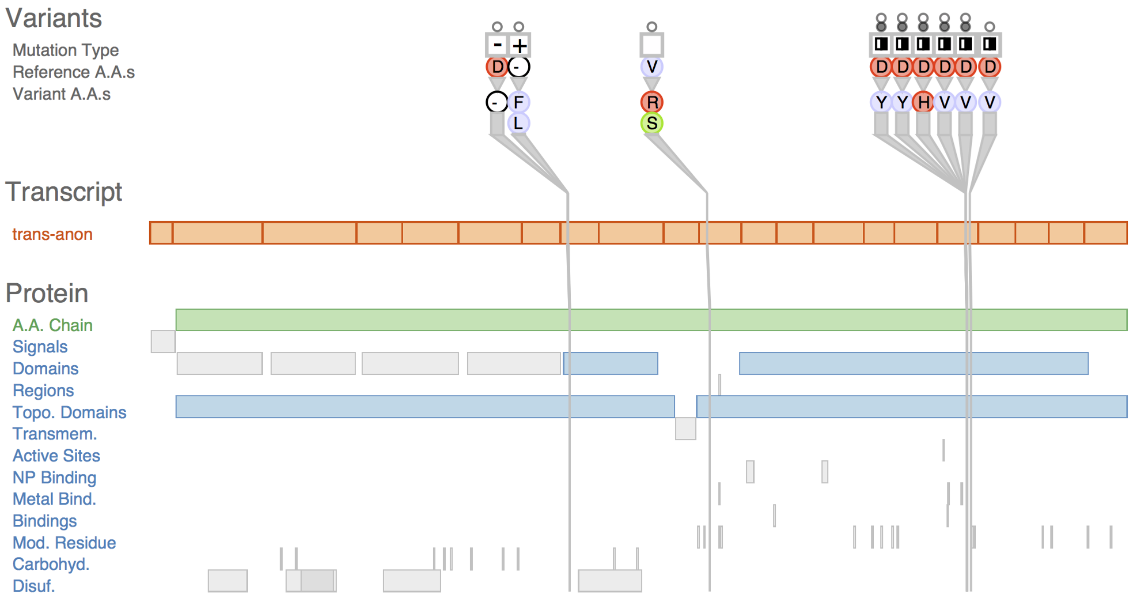
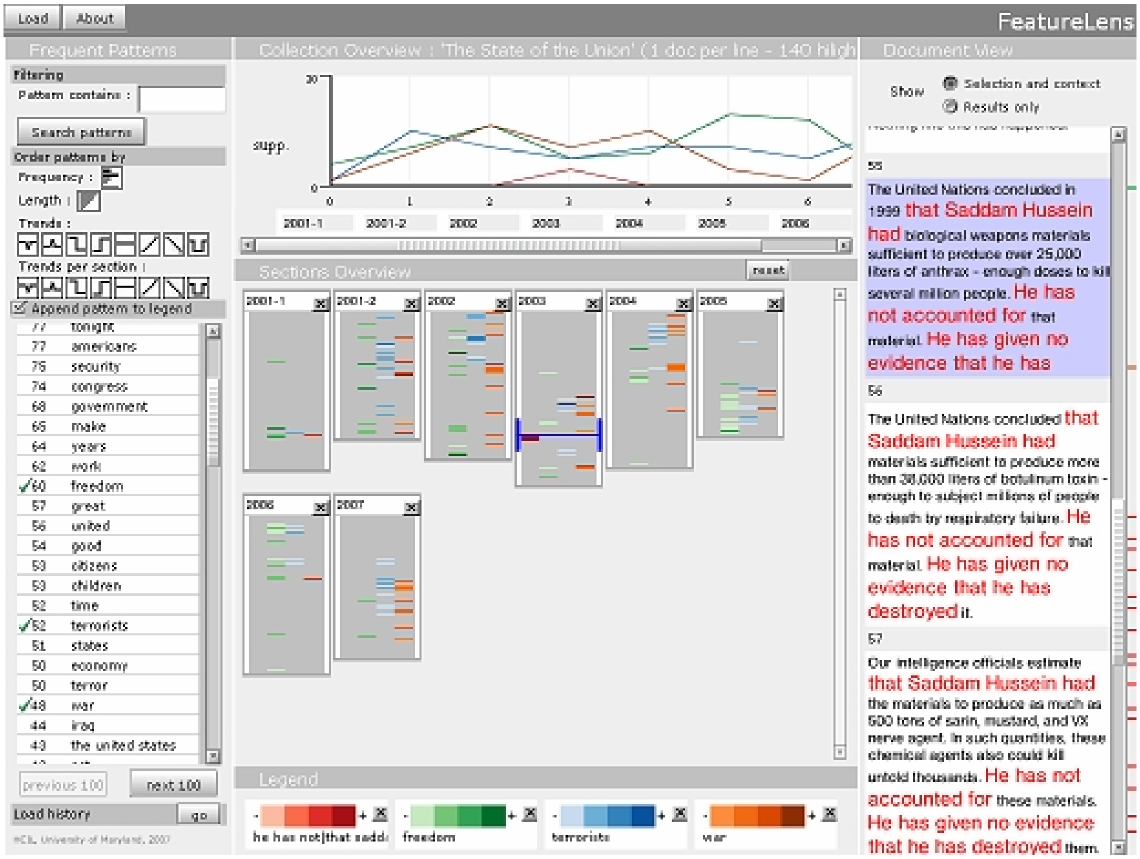


Figure 2.4: Variant View is a genome browser tool that aligns sequences by similarity.

are often compared as a sequence of linear *positions*, which does not lend itself to distinctions between point events versus interval durations.

Texts are often compared by extraction of frequent n-grams [22]. FeatureLens (Figure 2.5) by Don et al. [23] defined n-gram as a contiguous sequence of words and used a visualization approach to compare the co-occurrences of frequent n-grams of text. However, it only supports comparison among sections of a single document. Jankowska et al. [24] proposed to convert documents into vectors of frequent character n-grams and designed a relative n-gram signature to encode the distance between n-gram vectors. Viégas et al. presented history flow [25] (Figure 2.6) to visually compare between versions of a document. Their approach assumes that the later version of a document is developed based on the earlier one, which is not applicable to event history data.



77	tonight
77	americans
76	security
74	congress
68	government
65	make
64	years
62	work
✓60	freedom
57	great
56	united
54	good
53	citizens
53	children
52	time
✓52	terrorists
51	states
50	economy
50	terror
✓48	war
44	iraq
43	the united states

55

The United Nations concluded in 1999 that Saddam Hussein had biological weapons materials sufficient to produce over 25,000 liters of anthrax - enough doses to kill several million people. He has not accounted for that material. He has given no evidence that he has

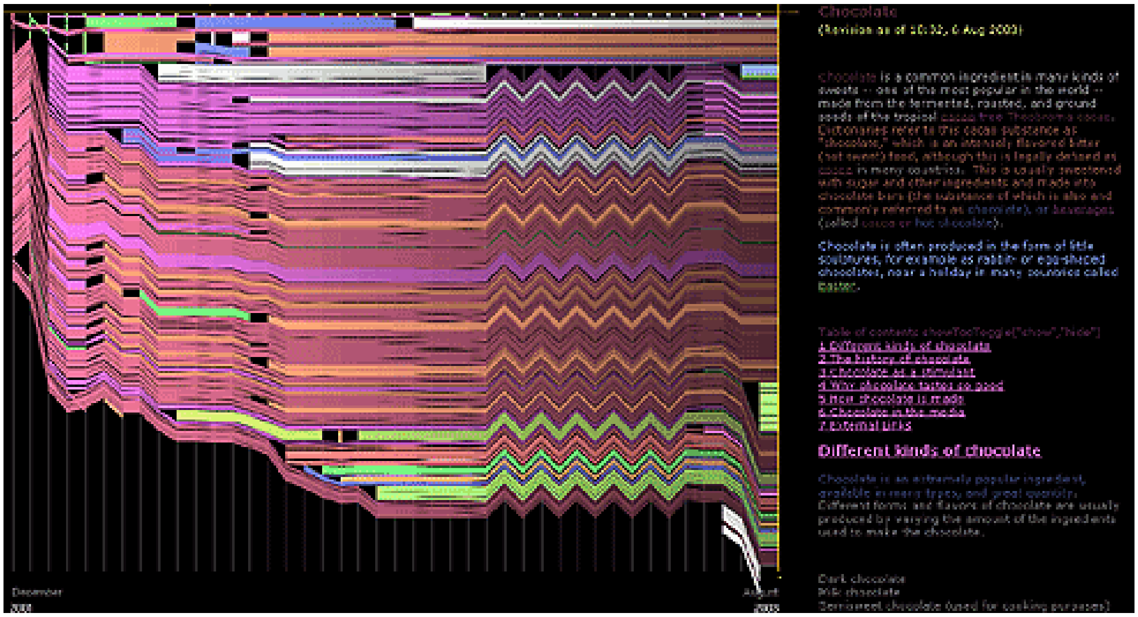
56

The United Nations concluded that Saddam Hussein had materials sufficient to produce more than 38,000 liters of botulinum toxin - enough to subject millions of people to death by respiratory failure. He has not accounted for that material. He has given no evidence that he has destroyed it.

57

Our intelligence officials estimate that Saddam Hussein had the materials to produce as much as 500 tons of sarin, mustard, and VX nerve agent. In such quantities, these chemical agents also could kill untold thousands. He has not accounted for these materials. He has given no evidence that he has destroyed them.

Figure 2.5: FeatureLens visualizes frequent patterns in text collections.



Chocolate

(Revision as of 10:32, 6 Aug 2003)

chocolate is a common ingredient in many kinds of sweets -- one of the most popular in the world -- made from the fermented, roasted, and ground seeds of the tropical cacao tree Theobroma cacao. Dictionaries refer to this cacao substance as "chocolate," which is an intensely flavored bitter (not sweet) food, although this is legally defined as cocoa in many countries. This is usually sweetened with sugar and other ingredients and made into chocolate bars (the substance of which is also and commonly referred to as chocolate), or beverages (called cocoa or hot chocolate).

Chocolate is often produced in the form of little sculptures, for example as rabbit- or egg-shaped chocolates, near a holiday in many countries called Easter.

Table of contents showTocToggle("show","hide")

1 Different kinds of chocolate
2 The history of chocolate
3 Chocolate as a stimulant
4 Why chocolate tastes so good
5 How chocolate is made
6 Chocolate in the media
7 External links

Different kinds of chocolate

Chocolate is an extremely popular ingredient, available in many types and great quantity. Different forms and flavors of chocolate are usually produced by varying the amount of the ingredients used to make the chocolate.

Dark chocolate
Milk chocolate
Semisweet chocolate (used for cooking purposes)

Figure 2.6: History Flow visualizes changes between versions of the same document.

Most of the techniques mentioned above (in both genomics and text mining) only provide a visual comparison between single long sequences, whereas event history data consists of many, short transactional sequences.

Temporal event sequences are often represented as trees. While many comparison techniques exist for trees, many do not take into account values or attributes of nodes and none are specifically designed for temporal data. Munzner presented the TreeJuxtaposer [26] (Figure 2.7) system to help biologists explore structural details of phylogenetics, but focuses only on structural differences in the trees and not any attributes about the nodes (such as timestamps). Bremm [27] studied the comparison of phylogenetic trees in a more statistical way by extending the algorithms of TreeJuxtaposer to compare more than two trees and considers "edge length" which could be generalized to durations of gaps between sequential events. Holten [28] presented an interactive visualization method to compare different versions of hierarchically organized data. He proposed two methods of tree comparison: icicle plot and hierarchical sorting, but does not propose any statistical comparison technique, and focuses more on "leaf-to-leaf" matching, which considers whole paths (or sequences) only.

TreeVersity2 [29] (Figure 2.8) compares by tree structure and the node values. Though TreeVersity2 is general to all trees, it leaves out temporal-specific analysis such as duration of or between interval events. TreeVersity2 compares two datasets over time, but assumes these time periods are disjoint. CoCo does not assume that the datasets are split by a time attribute and treats time of the nodes as another comparable attribute in the dataset. TreeVersity2 also includes a textual reporting

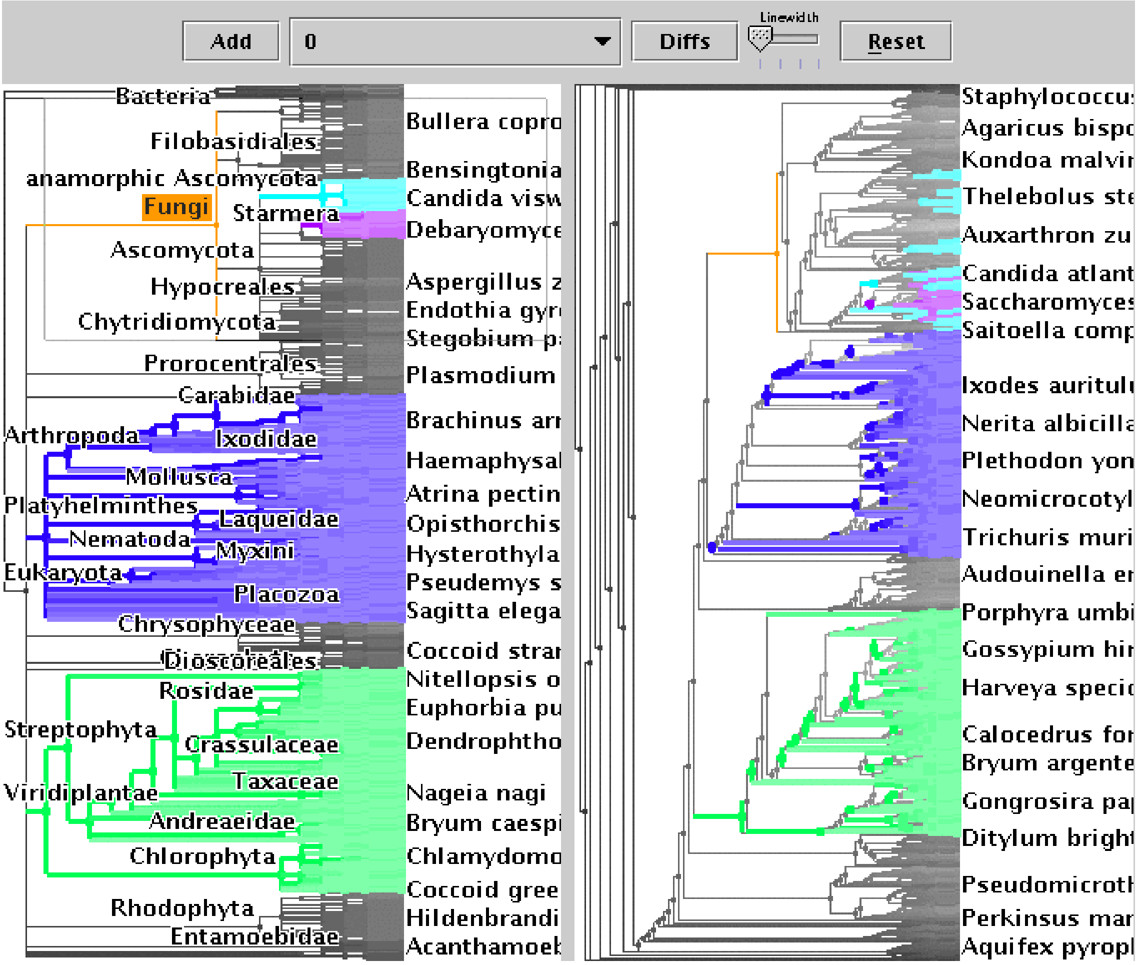
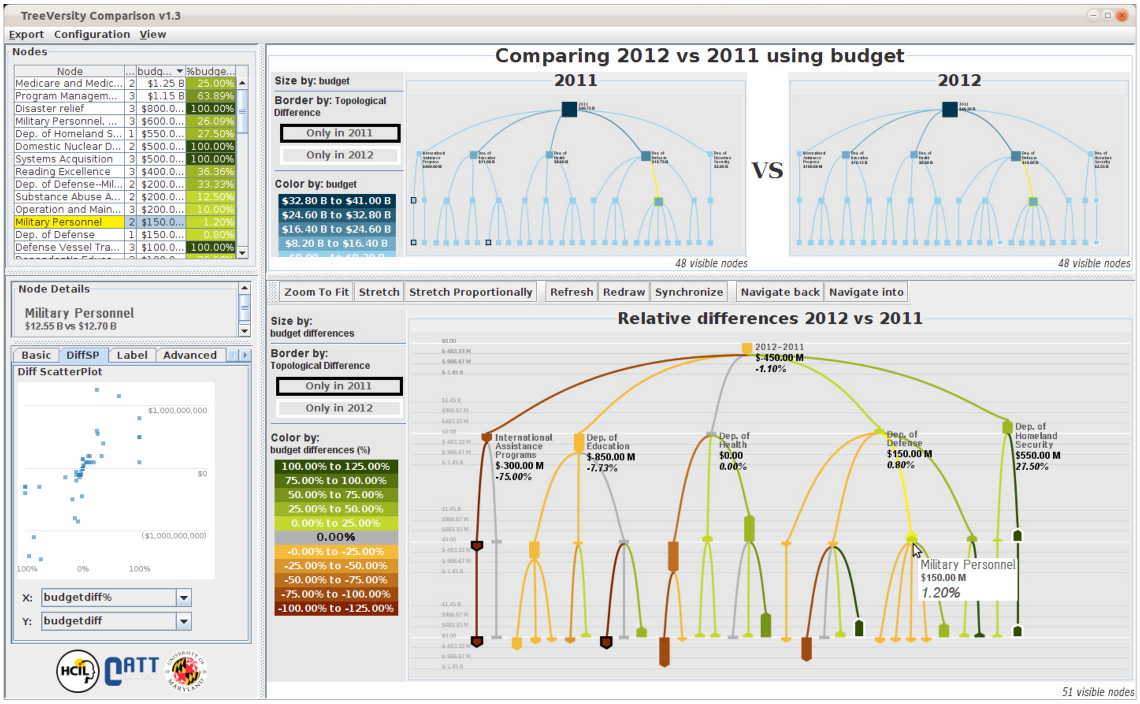


Figure 2.7: The TreeJuxtaposer system to help biologists explore structural details of phylogenetics and focuses only on structural differences in the trees but not any attributes about the nodes (such as timestamps).



TreeVersity Comparison v1.3

Nodes

Node	...	budg...	%budge...
Medicare and Medic...	2	$1.25 B	25.00%
Program Managem...	3	$1.15 B	63.89%
Disaster relief	3	$800.0...	100.00%
Military Personnel, ...	3	$600.0...	26.09%
Dep. of Homeland S...	1	$550.0...	27.50%
Domestic Nuclear D...	2	$500.0...	100.00%
Systems Acquisition	3	$500.0...	100.00%
Reading Excellence	3	$400.0...	36.36%
Dep. of Defense--Mil...	2	$200.0...	33.33%
Substance Abuse A...	2	$200.0...	12.50%
Operation and Main...	3	$200.0...	10.00%
Military Personnel	2	$150.0...	1.20%
Dep. of Defense	1	$150.0...	0.80%
Defense Vessel Tra...	3	$100.0...	100.00%

Node Details

Military Personnel
$12.55 B vs $12.70 B

Diff ScatterPlot

X: budgetdiff%

Y: budgetdiff

Comparing 2012 vs 2011 using budget

Size by: budget

Border by: Topological Difference

Color by: budget

Relative differences 2012 vs 2011

Size by: budget differences

Border by: Topological Difference

Color by: budget differences (%)

Figure 2.8: TreeVersity visually compares trees with similar structures.

tool that highlights outliers in the data.

Many of these comparison techniques also lack a statistical significance test for the comparisons. In this work, the comparison supports both visual and statistical approaches.

2.2 Statistics for Comparing Cohorts

In medical cohort studies, the most prevalent approach for comparison is survival analysis. In survival analysis, survival time is defined as the time from a defined point to the occurrence of a given event [30], and the Kaplan-Meier method is often used to analyze the survival time of patients on different treatments and to compare their risks of death [30–33]. Based on the Kaplan-Meier estimate, survival

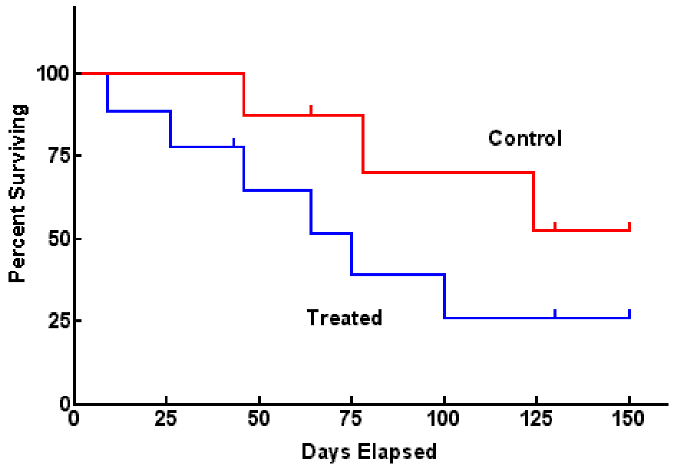


Figure 2.9: The Kaplan-Meier Estimator is used to compare the survival rates of groups of patients receiving different treatments. The estimator shows the maximum possible likelihood of survival (as a percentage) for each group as a function of time.

time of two groups of patients can be visualized (Figure 2.9) and compared with survival curves, which plot the cumulative proportion surviving against the survival times [30]. Also, the log-rank test is often used to statistically compare two survival curves by testing the null hypothesis [30]. Dupont et al. applied survival analysis in their clinical study [32]. Compared with survival analysis, the event sequences data used in this work is much more complicated, and requires a more advanced analysis model.

Currently tools that combine visualization and statistics for medical cohort analysis focus on single cohorts. CAVA [34] (Figure 2.10) is a visualization tool for interactively refining cohorts and performing statistics on a single group. Recently, Oracle published a visualization tool for cohort study [35]. Based on patients' clinical data, it supports interactive data exploration and provides statistics as well

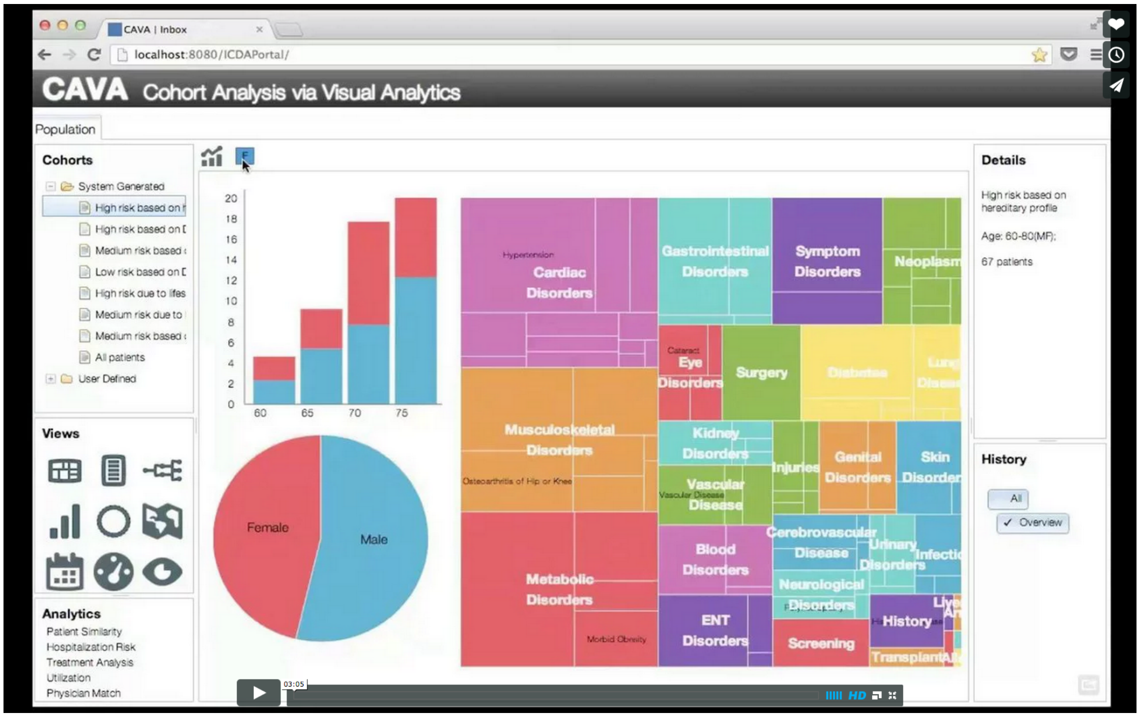


Figure 2.10: CAVA combines visual analytics and statistics by allowing users to interactively refining cohorts and perform statistics on a single group.

as visualization functionalities. These tools similarly focus on combining visualization with automated statistics and providing an interactive interface for selecting cohorts; however, both tools aim at grouping and identifying patient cohorts for further characterization, while my work focuses on comparing two existing cohorts based on their event histories.

2.3 Exploratory Hypothesis Testing

John Tukey describes statistical methods for summarizing data set characteristics in Exploratory Data Analysis [36], some of which are employed in CoCo. As event sequence datasets grow larger and larger, researchers are moving towards more exploratory methods for hypothesis generation and testing. The statistical implications of high-volume hypothesis testing (e.g., inevitable false positives) have been extensively researched [37,38]. CoCo treats each result independently, leaving the application of statistical corrections for future research.

Liu et al. [39] explore the statistical and technical implications of automatically generating and testing many hypotheses. Similar to this work, they find that interactive techniques such as sorting and filtering are necessary for parsing these result sets, but their display is largely textual. This work explores more visual methods for displaying both the hypothesis and results.

2.4 Temporal Data Mining

Automated hypothesis testing is closely related to big data mining. Previous work studying temporal data mining has mostly focused on discovering frequent temporal patterns and computing temporal abstractions of time-oriented data. Gupta et al. [40] provide a survey on outlier detection for temporal data sets.

There are many established algorithms for frequent sequence mining [41, 42] and association rule (itemset) mining [43]. The majority of data mining techniques focus on mining sequences in a single dataset and not comparing across two datasets. While two data mining techniques can be used in tandem to facilitate similar comparisons (e.g. comparing frequent sequence results across two datasets), more specialized methods are needed to answer "which sequences occur significantly differently between these datasets?" Bay and Pazzani introduce contrast mining sets [44], an algorithm for detecting differences between groups based on record attributes, such as age, gender, or occupation. In addition to record attributes, CoCo also looks at differences in event sequences, based on both occurrence and timestamps.

Pattern discovery is an open-ended problem which aims to unearth all patterns of interest [11]. Much of the literature is concerned with developing efficient algorithms to automatically discover frequent temporal patterns and extract temporal association rules [45–51]. To constrain the search procedure, some algorithms [45,47] allow users to provide initial knowledge and rules. Many of the algorithms are generalized to any sequence of tokens, however some tools [52] modify existing sequence mining algorithms to incorporate temporal attributes as well. To show the results, Norén et al. [53] used a graphical approach to visualize temporal associations.

Temporal abstraction focuses on obtaining a succinct and meaningful description of a time series [54]. Klimov et al. [55] developed VISITORS to visualize patient records by grouping the event attribute values at different temporal granularities. Moskovitch et al. [54] aggregated values of point data by state and trend, to obtain its interval representation. Batal et al. [56] converted time series data into vectors of frequent patterns, which can be used with standard vector-based algorithms. However, most of the work in this topic only focused on the time and value dimensions of an event category (a concept), which is considered as event attributes in this work.

Typical data mining algorithms are a blackbox, allowing little user involvement during the process. Recent work has been done on interactive sequence mining [52, 57–59], though these system focus primarily on mining frequent patterns in a single dataset. Little work has been done on involving the user in mining differences between datasets.

2.5 Scalability in Visual Analytics

Scalability in visual analytics has two main components: scaling of the visualization itself when displaying large amounts of data and optimization of algorithms for processing and analyzing this data.

Approaches for visualizing a large volume of data include displaying only a sample of the data, providing interaction techniques to “drill-down”, or aggregating the display. Fishet et al. [60] simplify large data visualization by using random

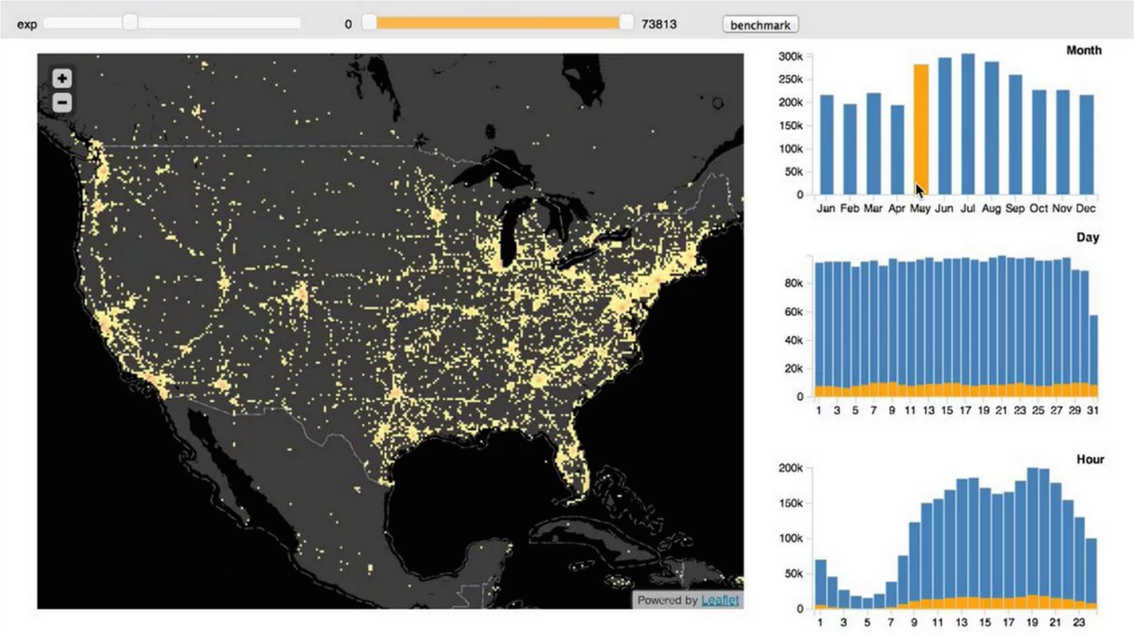
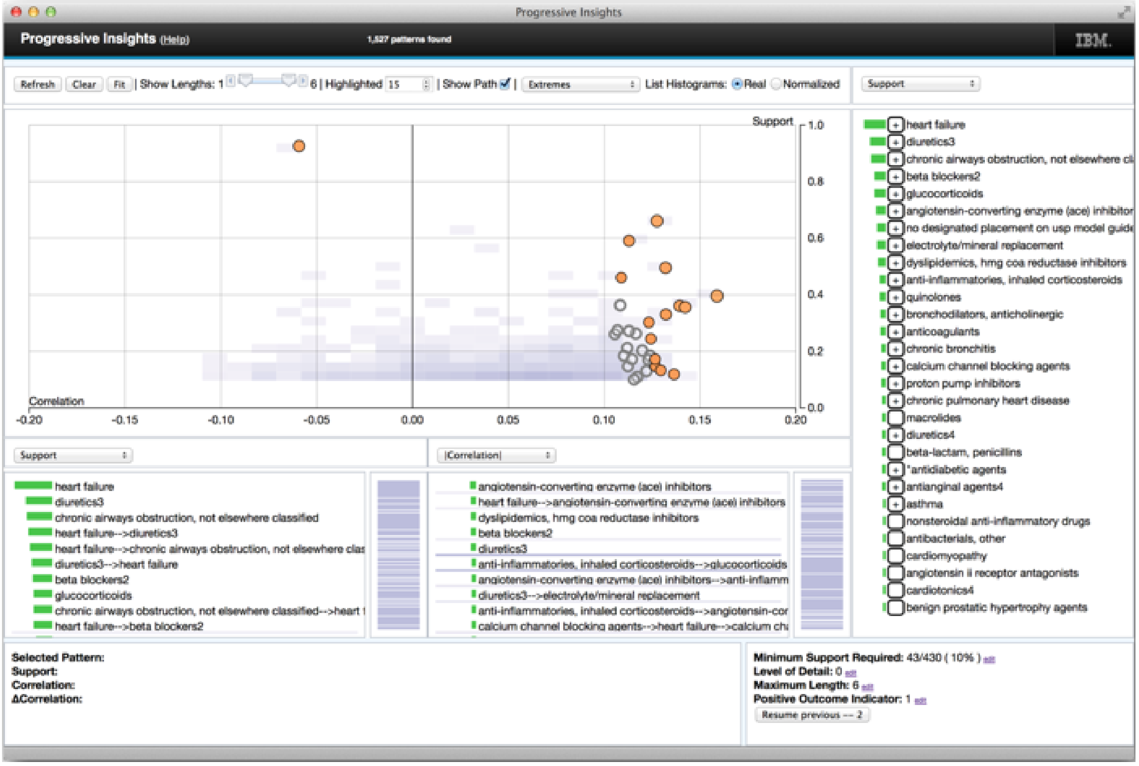


Figure 2.11: imMens uses aggregation to scale to large datasets.

sampling to incrementally display results to users. EventFlow [1] and imMens [61] (Figure 2.11) use aggregation techniques to display a large volumes of data.

There has been less work on optimizing the computation portion of visual analytics. Stolper et al. [7] introduce Progressive Insights (Figure 2.12), which allows users to see in-progress visualizations in order to allow users to guide the algorithm and ignore subspaces of the data that may not be relevant. In databases, multiple query optimization [62] is a technique to use the results of previous queries to reduce execution time on future, related query. However, a large part of this research falls under range queries, where the results of one query might be a subset of another.



- heart failure
- diuretics3
- chronic airways obstruction, not elsewhere classified
- heart failure-->diuretics3
- heart failure-->chronic airways obstruction, not elsewhere clas
- diuretics3-->heart failure
- beta blockers2
- glucocorticoids
- chronic airways obstruction, not elsewhere classified-->heart f
- heart failure-->beta blockers2

- angiotensin-converting enzyme (ace) inhibitors
- heart failure-->angiotensin-converting enzyme (ace) inhibitors
- dyslipidemics, hmg coa reductase inhibitors
- beta blockers2
- diuretics3
- anti-inflammatories, inhaled corticosteroids-->glucocorticoids
- angiotensin-converting enzyme (ace) inhibitors-->anti-inflamm
- diuretics3-->electrolyte/mineral replacement
- anti-inflammatories, inhaled corticosteroids-->angiotensin-cor
- calcium channel blocking agents-->heart failure-->calcium ch

- ⊞ heart failure
- ⊞ diuretics3
- ⊞ chronic airways obstruction, not elsewhere cl
- ⊞ beta blockers2
- ⊞ glucocorticoids
- ⊞ angiotensin-converting enzyme (ace) inhibitor
- ⊞ no designated placement on usp model guide
- ⊞ electrolyte/mineral replacement
- ⊞ dyslipidemics, hmg coa reductase inhibitors
- ⊞ anti-inflammatories, inhaled corticosteroids
- ⊞ quinolones
- ⊞ bronchodilators, anticholinergic
- ⊞ anticoagulants
- ⊞ chronic bronchitis
- ⊞ calcium channel blocking agents
- ⊞ proton pump inhibitors
- ⊞ chronic pulmonary heart disease
- □ macrolides
- ⊞ diuretics4
- □ beta-lactam, penicillins
- ⊞ "antidiabetic agents
- ⊞ antianginal agents4
- ⊞ asthma
- □ nonsteroidal anti-inflammatory drugs
- □ antibacterials, other
- □ cardiomyopathy
- □ angiotensin ii receptor antagonists
- □ cardiotonics4
- □ benign prostatic hypertrophy agents

Selected Pattern:
Support:
Correlation:
ΔCorrelation:

Minimum Support Required: 43/430 (10%) edit
Level of Detail: 0 edit
Maximum Length: 6 edit
Positive Outcome Indicator: 1 edit

Figure 2.12: Progressive Insights allows users to see in-progress visualizations in order to allow users to guide the algorithm and ignore subspaces of the data that may not be relevant.

2.6 Summary

This chapter covers the related work for cohort comparison. Event sequence cohort comparison lies at the intersection of event sequence visualization, statistical methods, exploratory hypothesis testing, temporal data mining and scalability. Though much work has been done toward supporting cohort comparison with regard to visualizing single groups of event sequences and the visual comparison of complex objects, the areas of event sequence comparison and balancing automated hypothesis testing with an interactive user interface are largely unexplored.

Chapter 3: A Taxonomy of Metrics for Comparing Cohorts

The first phase of my research was to explore the space of temporal event sequence comparison and to identify what questions analysts were asking when performing cohort comparison.

I conducted a literature review of seven case studies with EventFlow [63] and current methods for cohort comparison. Overwhelmingly, there was a disconnect between the questions that were being asked and the answers existing tools provided.

The results suggested that some broad insights can be gained by visually comparing pairs of EventFlow displays (e.g., analysts could see if the patterns were very similar overall between two groups) or very different (e.g., a lot more red or the most common patterns were different) but analysts repeatedly expressed the desire for more systematic ways to compare cohorts of records. However, existing tools were not designed to support exploration, but instead focused on answering a concrete hypothesis. For instance, analysts were asking simply "What patterns lead to two different outcomes?" where as the tool supported simple yes or no queries such as, "Does XYZ lead to a specific outcome?" The who, what, when, and why of the inquiry was difficult for the analyst to explore.

Following this observation, I looked at what insights analysts had discovered through their use of the tools and discovered that a number of common patters of inquiry existed. The most commonly explored aspect of sequence comparison focuses on the structure of the sequences (e.g., order of consecutive events, co-occurrences of non-consecutive events) and the frequency of sequences. However, event and record attributes (e.g., gender of a patient) and the timestamps themselves (e.g., duration of an event) can also be distinguishing features between the cohorts.

I constructed a taxonomy based on the observations made through the literature review and by observing analysts with three overarching goals: (1) support more open-ended questions that answer the who, what, when, and why when comparing event sequences, (2) organize these questions in a way that promotes *systematic* exploration, and (3) provide a more holistic comparison, beyond looking at only the structure of the sequences.

The taxonomy is organized in three parts: (1) summary metrics, (2) record metrics, and (3) event sequence metrics. Though this taxonomy can be applied to a variety of fields, the dataset used as an example for the remainder of this chapter consists of records of patients who were admitted to the emergency room and follows their movement through their stay at the hospital (Figure 3.1): being administered aspirin, being admitted into the hospital room, transferring between a normal floor bed and the intensive care unit (ICU), and ultimately being discharged either dead or alive. The dataset is split into two cohorts: patients who died and patients who lived.

While this taxonomy is derived from numerous case studies in seven domains and aims to show the complexity and variety of questions asked during cohort com-

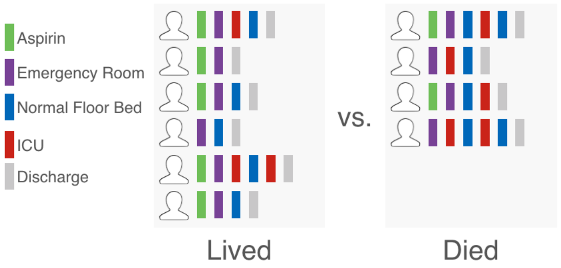


Figure 3.1: The dataset used as an example for the remainder of this chapter consists of records of patients who were admitted to the emergency room and follows their movement through their stay at the hospital: being administered aspirin, being admitted into the hospital room, transferring between a normal floor bed and the intensive care unit (ICU), and ultimately being discharged either dead or alive.

parison, it can be expanded upon with the inclusion of more metrics that may be required for alternate domains and situations (e.g., similarity metrics or metrics dealing with the absence of events).

3.1 Summary Metrics

Summary metrics deal with the cohorts as a whole and provide a high-level overview of the datasets.

Number of records. Raw number of records in each cohort (Figure 3.2).

Number of events. Raw number of events in each cohort (Figure 3.3).

Number of unique records. Total number of unique records in each cohort based on the sequence of events (timestamps are not considered).

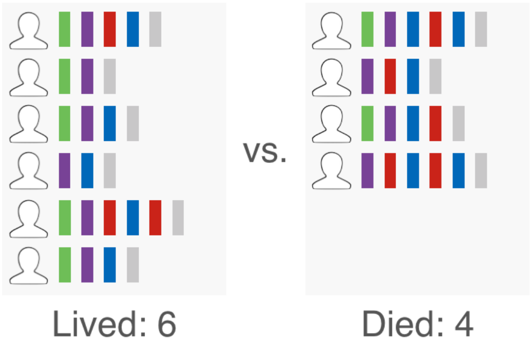


Figure 3.2: Because cohorts do not necessarily need to be the same size, it is important to report on the number of records in each cohort. An understanding of the number of records allows analysts to understand broad trends between the cohorts (e.g., is the selection criteria balanced?) In this example, there are only 4 patients who died versus 6 who lived.

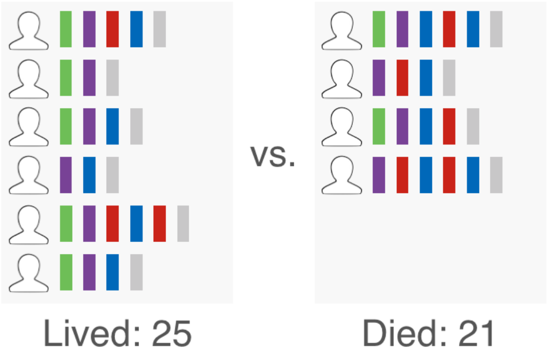


Figure 3.3: The number of events is the raw number of events in each cohort. Combined with the number of records metric, this can reveal interesting information about the frequency of events and the average length of records. In this example, though there are 50% more patients who lived than those who died, the number of events is only 20% greater, indicating that patients who died have longer sequences, on average, than those who live.

Number of each event. Total number of occurrences for each event cateogry per cohort.

Minimum, Maximum, and Average length of records. The length of a record is considered as the number of events in that record.

3.2 Record Metrics

Record-level attributes (such as patient gender or age) compare the cohorts as population statistics. General statistics across the entire dataset is a problem already tackled by analytics tools such as Spotfire [64] or Tableau [65], however these tools look at a single attribute. For example, they might compare the number of males versus females or patients on Wednesday versus Thursday. There may be implications about the *combinations* of record attributes (e.g., the women on Wednesday versus the women on Thursday versus the men on Wednesday versus the men on Thursday). In clinical trials, it is important that all patient attributes are balanced and currently no tools exist for visually confirming that all attribute combinations are balanced (Figure 3.4).

3.3 Sequence Metrics

Sequence metrics deal with hypotheses at a sequence-level and can refer to (1) the occurrence of sequences, (2) the timing of sequences, or (3) event-level attributes.

Sequences are differentiated by type and can refer to any number of types:

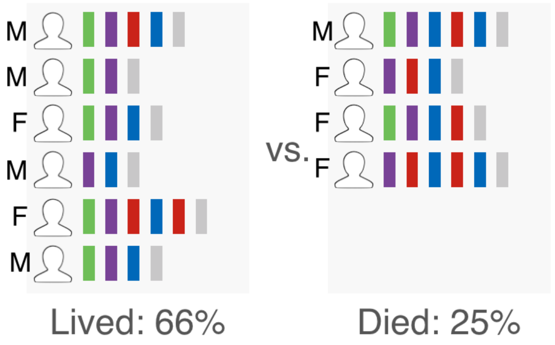


Figure 3.4: Prevalence of record attributes reports on the percent of records who have a particular value. This example is comparing the proportion of male and female patients between the two groups.

Sequence A record's entire history.

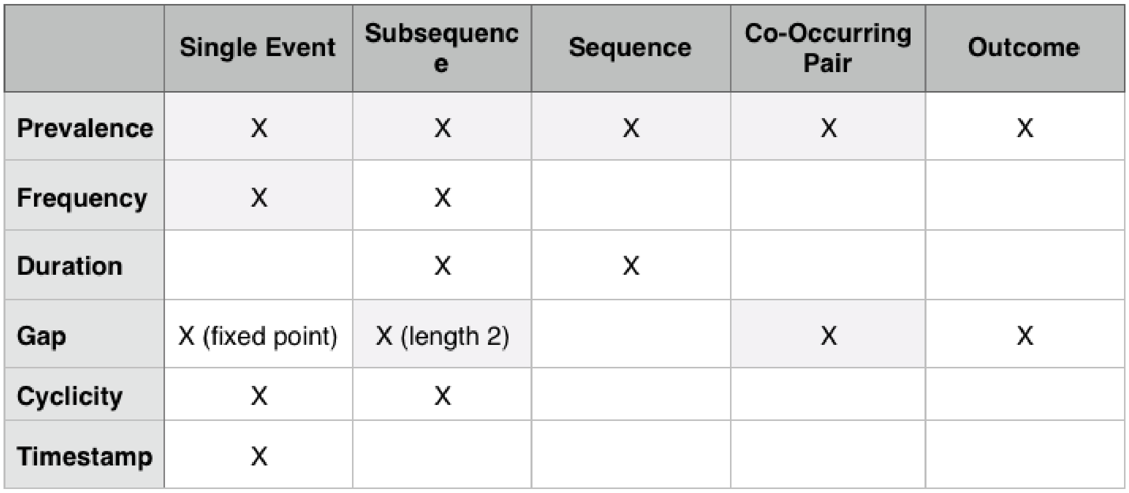
Subsequence A consecutive part of a record, consisting of two or more events.

Event A subsequence of length one, or a single event category.

Co-occurring pair Two events, that may occur non-consecutively within a single record.

Outcome The last event in a record.

Many of the metrics can be applied to multiple sequence types, but not to all. For example, metrics dealing with event gaps can only be applied to sequences of length 2 (consecutive or non-consecutive). Table 3.1 shows which matrix are applicable to which sequence types and the following sections describe each, organized by occurrence, time, and attribute metrics.



	Single Event	Subsequence	Sequence	Co-Occurring Pair	Outcome
Prevalence	X	X	X	X	X
Frequency	X	X			
Duration		X	X		
Gap	X (fixed point)	X (length 2)		X	X
Cyclicity	X	X			
Timestamp	X				

Table 3.1: This table shows the applicable metrics for each sequence type (denoted by an X). Metrics with shaded cells are those that were implemented in the final version of CoCo.

3.3.1 Occurrence Metrics

Prevalence of an event. The percent of records or total number of events that a particular event occurs in (Figure 3.5). **Implemented in CoCo.*

Prevalence of a subsequence. The percent of records in which the subsequence appears. For example, patients who lived are given aspirin before going to the emergency room more often than the patients who died (Figure 3.6). **Implemented in CoCo.*

Prevalence of a whole sequence. Percent of records with a given sequence. **Implemented in CoCo.*

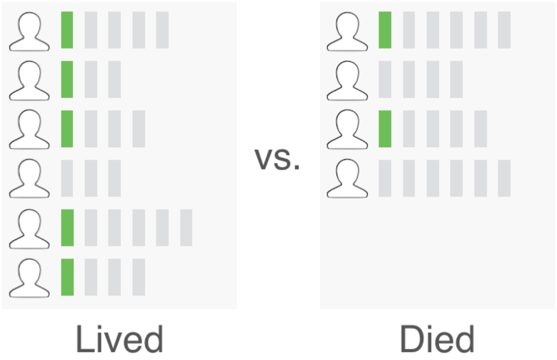


Figure 3.5: The prevalence of an event is calculated as a percentage of records that contain that particular event.

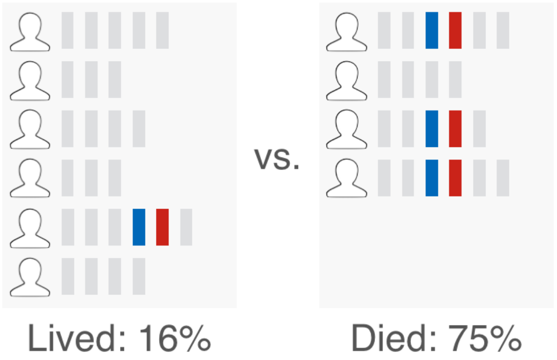


Figure 3.6: The prevalence of a subsequence is calculated as a percentage of records that contain that particular subsequence.

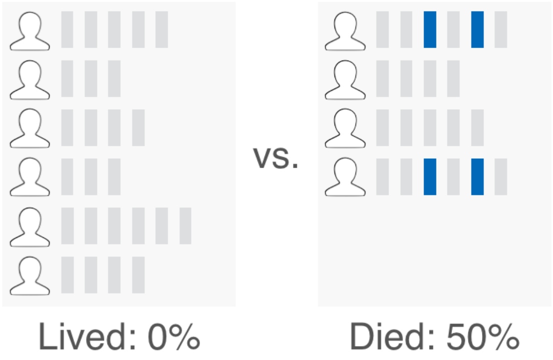


Figure 3.7: Co-occurring events are a pair of events which occur within a single record, and may or may not have other events between them.

Prevalence of Co-occurring Events. The percent of records containing both events A and B (with any number of events between them, Figure 3.7). **Implemented in CoCo.*

Prevalence of Outcomes. If a single event is prevalent as an "outcome" (i.e., the last event in the sequence). This metric in particular applies only to cohorts that are not already split on an outcome event.

Frequency of an event. The number of times per record an event occurs. Because this is a distributed numerical metric, the system can report on minimum, maximum, mean, median, mode, and the distribution as a histogram. **Implemented in CoCo.*

Order of consecutive events in a subsequence. The percent of records containing event A directly preceding event B versus B preceding A. For example, perhaps patients who go to the ICU before the floor are more likely to live than

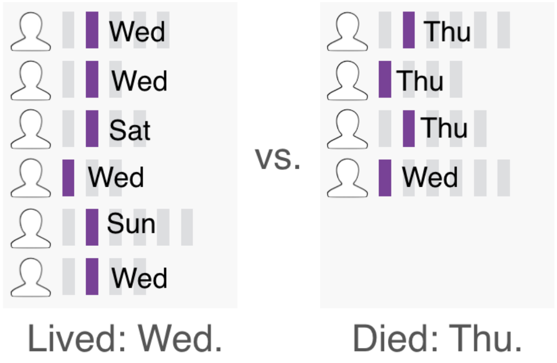


Figure 3.8: Absolute time metrics look at the timestamp of a particular event. For example, the prevalence of the day of the week can differ between the two cohorts.

patients who have these events in the reverse order. **Implemented in CoCo.*

3.3.2 Time Metrics

Time metrics deal with the timestamps at both the event and sequence levels – relative and absolute. All of these metrics result in distributed numerical values, so the system can report on minimum, maximum, mean, median, mode, and the distribution as a histogram for each.

Cyclicity The time between repeat occurrences of a sequence.

Gap The gap between two events.

Duration The duration a sequence takes to complete.

Absolute time of an event. Prevalence of a particular timestamp of an event or multiple events (e.g., if all events in one cohort occurred on the same day, Figure 3.8).

Duration from a fixed point in time. The length of time from a user-specified, fixed point – aligned by either a selected event or absolute date-time.

Duration of interval events. The duration of a particular interval event. For example, this can be the length of exposure to a treatment or the duration of a prescription.

Duration of a subsequence. The length of time from the beginning of the first event in a subsequence to the end of the last event in the subsequence.

Duration of overlap in interval events. The overlap (or lack thereof) of interval events. For example, the overlap of Drug A and Drug B could be more common in the cohort of patients who lived versus those who died.

Event Gap between consecutive events. The time between the end of one event and the beginning of the next. For example, the average length of time between hospital patients entering the emergency room and being transferred to the ICU is under two hours in patients who lived and over two hours in those who died. **Implemented in CoCo.*

Event Gap between co-occurring (non-consecutive) events. The length of time between non-consecutive events (two events with some number of other events occurring between them, Figure 3.10). **Implemented in CoCo.*

Cyclic events. The duration between cyclic events and sequences.

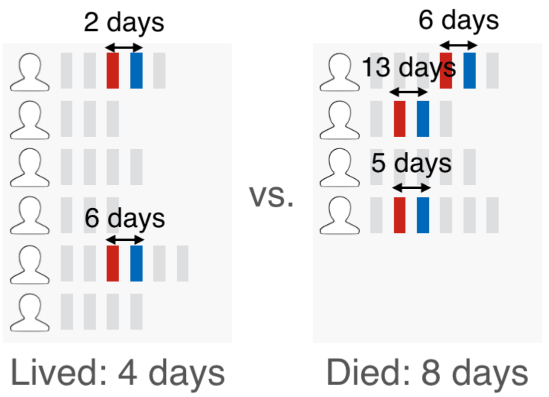


Figure 3.9: Relative time metrics involve comparing the average gap between two consecutive events.

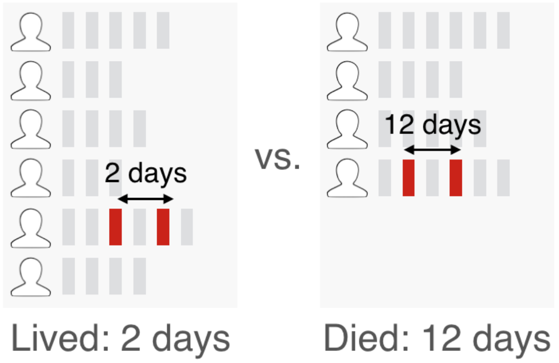


Figure 3.10: Relative time metrics involve comparing the average gap between two events.

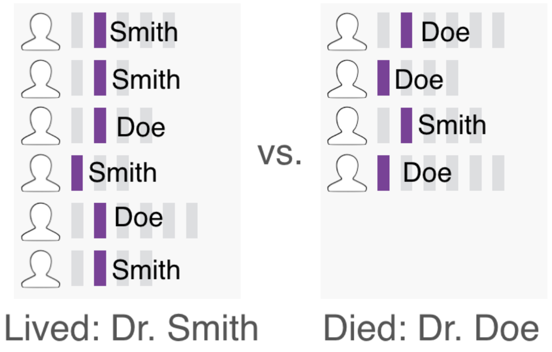


Figure 3.11: Attribute metrics are similar to other event metrics, but the events are further broken down by the attribute's value. In this example, the doctor that is on-call when the patient arrived at the emergency room is noted. Dr. Smith was on-call more often in patients who lived than those who didn't.

3.3.3 Event Attribute Metrics

Any of the above metrics can be applied over values of an attribute of the events instead of the event category itself. This can be done by swapping an event category by the values of a particular attribute. For example, in a medical dataset, analysts might be interested in seeing how a particular emergency room doctor might be related to the outcome of a patient. Analysts would then switch all events of "Emergency" with the value of its "doctor" attribute. If there are three doctors, this would create 3 new pseudo-event categories. Analysts can use the metrics from above to see the difference in event sequences, times, or prevalence of each doctor in either cohort (Figure 3.11).

3.4 Combining Metrics

The number of metrics is further multiplied because any combination of the above metrics is a new metric.

Survivor analysis. Survivor analysis is a common metric in cohort comparison studies in the medical field, for understanding how an event or sequence occurs or diminishes over time. This is equivalent to combining prevalence with time – how does the prevalence of an event occur over time.

3.5 Summary

This chapter presents Contribution 1: a taxonomy of metrics for comparing cohorts of event sequences. Although comparing two groups of data is a common task, with temporal event sequence data in particular, the task becomes complex because of the variety of ways the cohorts, sequences (entire records), subsequences (a subset of events in a record), and events can differ. Through a literature review of seven case studies of EventFlow and evaluation of cohort comparison, I work to understand how analysts perform cohort comparison and categorize their common questions. Though much work has been done in differentiating between the structure of the event sequences (e.g., order, co-occurrences, or frequencies of events), many analysts and tool miss the opportunity to explore the attributes about the events and records (e.g., gender of a patient), and the timestamps themselves (e.g., an event's duration) as distinguishing features between the cohorts. In this chapter, I present a taxonomy of 23 metrics of how cohorts can differ organized by cohort summary metrics, event sequences, and record attributes. While this taxonomy aims to be a holistic view of the cohort comparison space, there is the potential for expansion to many more metrics not mentioned here (Section 7.2). This taxonomy serves as an example of the complexity of questions that are possible when comparing cohorts of event sequences and to demonstrate that these questions can be asked systematically.

Chapter 4: Statistical Framework for High-Volume Hypothesis Testing

In any form of high-volume data analysis, wait times are a given, but this problem is especially prevalent when dealing with groups of event sequences because of the exponential number of unique sequences that exist in a single dataset. Consider the simple case of a dataset with only two events: A and B. Without considering repetitions, there are 5 unique event sequences that can occur:

$$\text{A} \quad \text{B} \quad \text{A}\rightarrow \text{B} \quad \text{B} \rightarrow \text{A} \quad \text{AB},$$

where AB represents two events occurring concurrently (at the same timestamp). When allowing repetition, the number of event sequences becomes infinite:

$$\text{A} \rightarrow \text{A} \quad \text{B} \rightarrow \text{B} \quad \text{A} \rightarrow \text{A} \rightarrow \text{B} \quad \text{AB} \rightarrow \text{B} \quad \ldots$$

Further, each event sequence can have multiple metrics applied to it. For example, with the sequence A → B, we can consider the prevalence among records (i.e., percent of records containing this sequence), frequency (i.e., average number of occurrences per record, duration (i.e., average time from A to B). When comparing cohorts, the application of each of these metrics to each cohort is equivalent to a

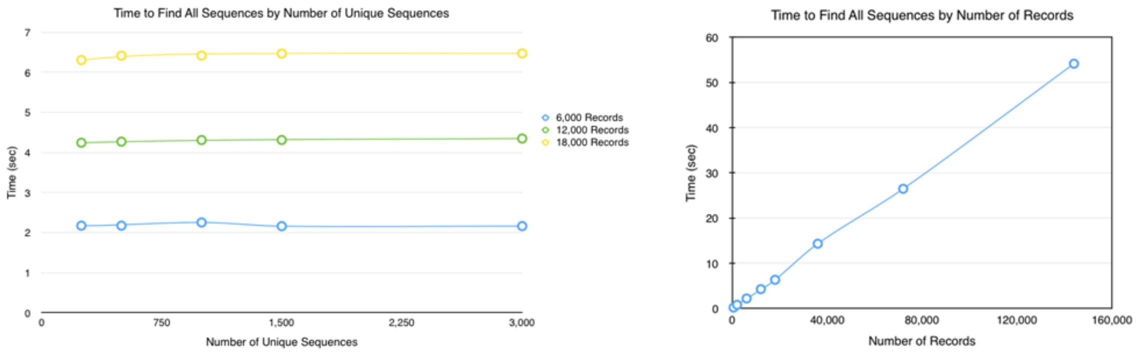


Figure 4.1: A chart of the average runtime to find all sequences (a) versus the number of unique sequences and (b) versus the number of records. Finding all subsequences within both datasets grows proportionally with the number of records in the dataset, whereas the number of unique sequences has no effect.

hypothesis. Does A → B occur similarly in both cohorts, or does it occur significantly more in one than the other? Is the duration of A → B the same in both cohorts, or is it longer in one than the other? Thus, a simple dataset with only five event categories can have hundreds of hypotheses applied to it and larger datasets quickly become challenging to process.

To provide a sense of timing, I conducted timing tests with the final version of CoCo with datasets of varying numbers of records (250, 500, 1000, 1500, 3000, 6000, 9000, 18,000, 36,000, and 72,000) and numbers of unique sequences (250, 500, 1000, 1500, 3000) in each cohort. I collected the runtimes for 100 runs each of finding all sequences within the datasets and calculated all the metrics. All tests were performed on a machine with an 2.2 GHz Intel Core i7 processor with 8 GB of memory. No multithreading was used.

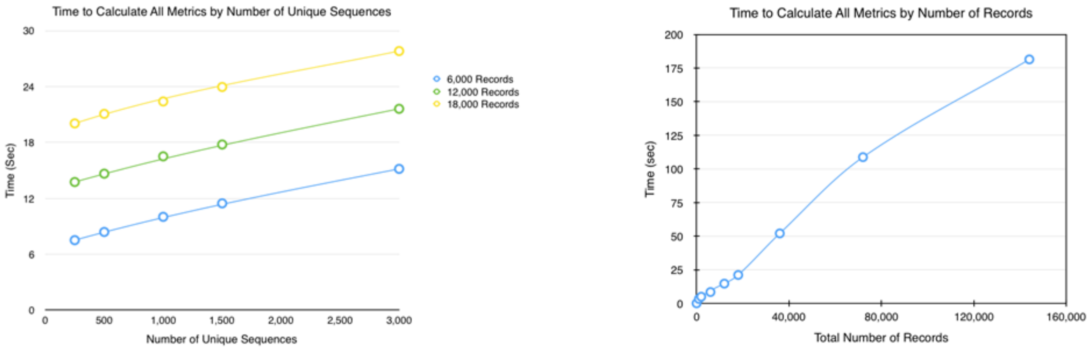
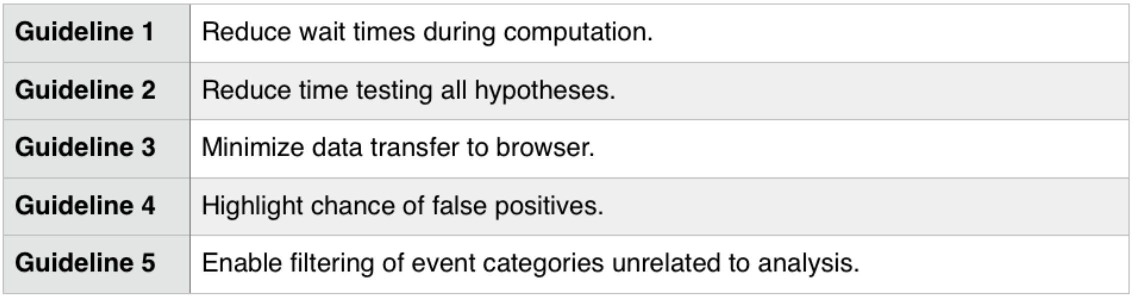


Figure 4.2: A chart of the average runtime to calculate all hypotheses (a) versus the number of unique sequences and (b) versus the number of records. Calculating all hypotheses depends both on the number of records and the number of unique sequences in the dataset.

Figure 4.1 provides a chart of the average runtime to find all sequences (a) versus the number of unique sequences and (b) versus the number of records. Finding all subsequences within both datasets grew proportionally with the number of records in the dataset, with the largest dataset (9000 records in each cohort) taking about 6.4 seconds to complete, regardless of the number of unique sequences. The number of unique sequences (and therefore, the number of events) did not have an effect on the time to find all sequences, because all sequences are mined based on the existing patterns in the dataset. That is, all sequences that are mined are a subsequence of each record as a whole, so every record must be checked.

Figure 4.2 provides the results for the average runtimes to calculate all hypotheses (a) versus the number of unique sequences and (b) versus the number of records. Calculating the metrics scaled proportionally with the number of unique sequences found, more rapidly than linearly. This is due to the fact that as for each new unique sequence, the number of new subsequences is potentially more than one, so every new sequence introduces at least two new subsequences. The effect is further multiplied by applying numerous metrics to a single sequence (e.g., prevalence AND time). Overall, the time to calculate hypotheses took much longer than the time to find all subsequences, with the largest dataset (9000 records and 3000 unique sequences in each cohort) taking over 27 seconds to calculate, due to the fact that approximating p-values requires the most time.

Because calculating the p-values is the most time-consuming aspect of hypothesis calculation, it becomes more important to present analysts with methods for reducing wait times, both on front- and back-ends. Through case studies with users, I identified five scalability guidelines for extending high-volume hypothesis testing to large datasets (Table 4.1).



Guideline 1	Reduce wait times during computation.
Guideline 2	Reduce time testing all hypotheses.
Guideline 3	Minimize data transfer to browser.
Guideline 4	Highlight chance of false positives.
Guideline 5	Enable filtering of event categories unrelated to analysis.

Table 4.1: The 5 Scalability Guidelines for extending high-volume hypothesis testing to large datasets.

In this chapter I describe the implementation of CoCo's statistical framework and how the components in its web-based client-server system are designed and organized. Lastly, I present five guidelines for performing high-volume hypothesis testing on event sequences to address these five issues.

4.1 System Overview: Backend

CoCo is a web application that uses the client-server method to divide the frontend and the backend. This section provides an overview of the architecture used in the backend, which was written in Python 2.7. Python was chosen because the extensive availability of packages, ability for fast development, and flexible, multi-paradigm nature. Flask [66] was chosen as the webserver framework for being easy-to-install and lightweight. For computing statistics, the well-known package SciPy [67] was used.

4.1.1 Code Structure and Organization

CoCo is organized into four main packages: (1) the main controller, (2) data processing, (3) metric computation, and (4) utilities. Figure 4.3 lays out each of the packages and the classes they contain.

1. Main controller. The main controller contains the necessary elements for setting up the main server, providing web endpoints for the frontend, and housing all variables needed on the server-side. All data is kept in memory with simple objects and native Python data structures, though it is organized such that future work may adapt it to a relational database backend (Section 7.2.4). A database backend was not implemented in the prototype in order to minimize the number of package dependencies when case study partners were installing CoCo on their machines. Additionally, the use of native Python lists and dictionaries allowed for

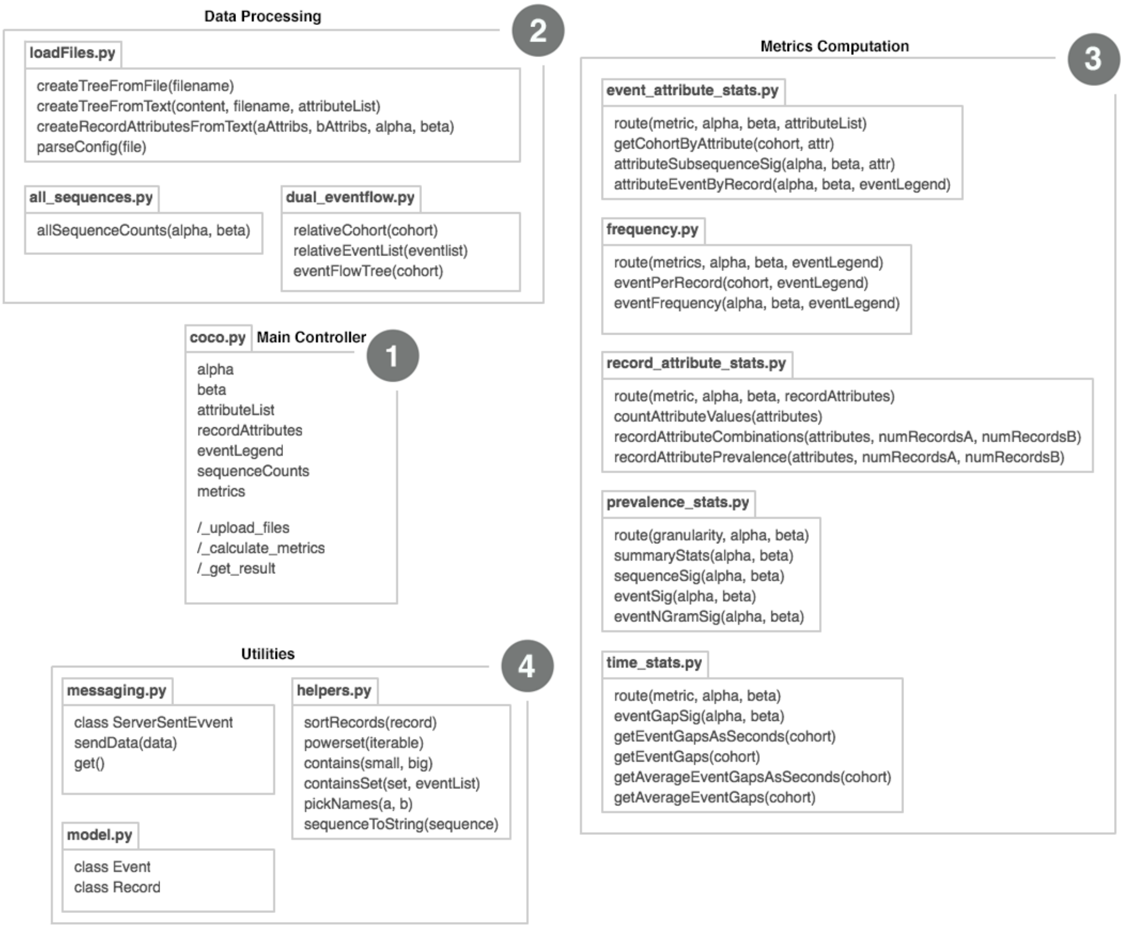

Data Processing
2
loadFiles.py
createTreeFromFile(filename)
createTreeFromText(content, filename, attributeList)
createRecordAttributesFromText(aAttribs, bAttribs, alpha, beta)
parseConfig(file)
all_sequences.py
allSequenceCounts(alpha, beta)
dual_eventflow.py
relativeCohort(cohort)
relativeEventList(eventlist)
eventFlowTree(cohort)
coco.py
Main Controller
1
alpha
beta
attributeList
recordAttributes
eventLegend
sequenceCounts
metrics
/_upload_files
/_calculate_metrics
/_get_result
Utilities
4
messaging.py
class ServerSentEvvent
sendData(data)
get()
model.py
class Event
class Record
helpers.py
sortRecords(record)
powerset(iterable)
contains(small, big)
containsSet(set, eventList)
pickNames(a, b)
sequenceToString(sequence)
Metrics Computation
3
event_attribute_stats.py
route(metric, alpha, beta, attributeList)
getCohortByAttribute(cohort, attr)
attributeSubsequenceSig(alpha, beta, attr)
attributeEventByRecord(alpha, beta, eventLegend)
frequency.py
route(metrics, alpha, beta, eventLegend)
eventPerRecord(cohort, eventLegend)
eventFrequency(alpha, beta, eventLegend)
record_attribute_stats.py
route(metric, alpha, beta, recordAttributes)
countAttributeValues(attributes)
recordAttributeCombinations(attributes, numRecordsA, numRecordsB)
recordAttributePrevalence(attributes, numRecordsA, numRecordsB)
prevalence_stats.py
route(granularity, alpha, beta)
summaryStats(alpha, beta)
sequenceSig(alpha, beta)
eventSig(alpha, beta)
eventNGramSig(alpha, beta)
time_stats.py
route(metric, alpha, beta)
eventGapSig(alpha, beta)
getEventGapsAsSeconds(cohort)
getEventGaps(cohort)
getAverageEventGapsAsSeconds(cohort)
getAverageEventGaps(cohort)

Figure 4.3: Code structure and organization.

fluid data transfer between the JavaScript-based frontend and the Python-based backend.

Variables. Each cohort is represented by a global variable as a dictionary mapping record IDs to `Record` objects, named `alpha` and `beta`. The dictionary allows for rapid looking for particular records, as well as quick collection of the records themselves. An attribute dictionary maps event attribute keys to a set of event attribute values. The record attribute list maps record attribute keys to another dictionary, for each cohort. The second dictionary maps the record ID to the record's attribute value. This allows for easy look-up by attribute, attribute value, cohort, and specific record. The event legend is a set of all event categories found in the datasets. The sequence counts for every sequence are stored in a list because it is the most accessed metric category. This datastructure is used by the sequence scatterplot, as well as to calculate all prevalence metrics, so it was important to store this data after it is calculated once.

Lastly, the metrics object is a dictionary which defines metrics by category and granularity. Each metric stores its title (e.g., "most differentiating events"), a description which is used in the tooltip, and results, which are empty to start. Each result object contains a list of hypothesis results, represented as 3-tuple containing value in cohort A, value in cohort B, and the p-value.

Web endpoints. The CoCo backend has four main endpoints:

- `/` – This is the index which renders the CoCo HTML template.

- `/_upload_files` – This endpoint is called when the analysts select and upload their data files. It is responsible for parsing all the input files into their respective data structures.

- `/get_sequence_counts` – After the data files are loaded and parse, the main JavaScript controller access this endpoint to retrieve the data for the sequence scatterplot.

- `/calculate_metrics` – This endpoint calls all necessary methods to automatically compute all hypotheses.

Other endpoints are used for shutting down the system cleanly (`/shutdown`) and serving and streaming server-sent events (`/stream`).

2. Data processing. This package contains classes dealing with processing the data in CoCo, including loading files and preparing data for visualizations.

File processing.. Within this package are methods for processing the files inputted by the user. Processing includes reading the data file, creating the cohort objects, assigning which cohort is which (left vs. right cohort), parsing the file name and giving it a human readable name, and processing any attribute and configuration files, if applicable.

Sequence extraction. This package computes the counts all the sequences and subsequences of length 1 to 10, and records how many times each sequence appears in each cohort record. The result is provided to the sequence scatterplot visualization.

EventFlow tree builder. This package processed data for the EventFlow visualization and returns an object that can displayed. This includes

3. Metric Computation. All methods dealing with computing metrics are grouped into this package, and organized by metric type: prevalence, time, frequency, record attributes, and event attributes.

Prevalence. There are three main methods for computing prevalence: consecutive sequences, single events, and non-consecutive pairs. This class also contains many helper functions which are used for counting each of these types of sequences in each dataset. All prevalence metrics are compared using chi-squared.

Time. This package calculates metrics dealing with time using a Wilcoxon rank-sum test to compare means.

Frequency. This package calculates metrics dealing with frequency of events using a Wilcoxon rank-sum test to compare means.

Attributes. This package uses chi-squared to run metrics dealing with record and event attributes.

4. Utilities.

Models. CoCo defines two simple classes, as explained in the previous section: `Event` and `Record`. The `Event` class contains:

`event` – the event category,

`time` – the timestamp of the event,

`recordID` – the ID of the record which this event is a part of, and

`attributes` – a dictionary mapping event attributes to values.

`Record` contains:

`recordID` – the record's unique identifier,

`eventList` – a nested list of `Event` objects, sorted by time. Each new event is inserted into its sorted position. Each inner list is a group of events with the same timestamp, which represent concurrent events, and

`attributes` – a dictionary mapping record attributes to values.

Helpers. This package contains various helper functions that are used throughout the system.

Messaging. This package contains the class and methods for creating and sending server-sent events (SSEs), which allows the server to communicate with the frontend.

4.1.2 Data Processing Pipeline

This section presents an overview of the data processing pipeline for CoCo (Figure 4.4).

CoCo reads in two files (one for each cohort) in the same format as EventFlow: 5-column, tab-delimited text file where each column is as follows:

1. Record ID. The ID of the record to which the event belongs. Each record

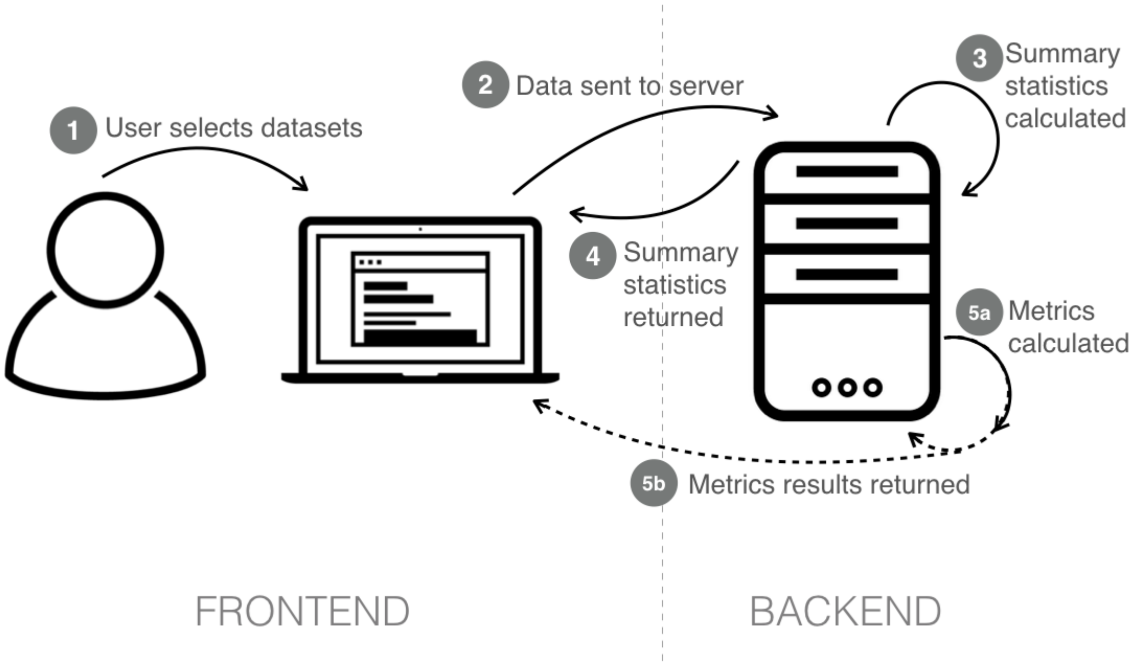

1 User selects datasets
2 Data sent to server
3 Summary statistics calculated
4 Summary statistics returned
5a Metrics calculated
5b Metrics results returned
FRONTEND
BACKEND

Figure 4.4: CoCo data processisning pipeline. CoCo processes data in five major steps: (1) Analysts select two datasets from the interface. (2) The data files are sent to the server. (3) Sequences and counts are extracted. (4) The results for the sequence counts are sent back to the client. (5) CoCo begins (a) calculating metric results and (b) sends them back as they are completed, until all metrics have been calculated.

should have a unique ID. Record IDs do not have to be unique between the two cohorts.

2. Event Category. The type of event.

3. Start Time. The start timestamp of the event.

4. End Time (optional). The end time of an interval event, left blank if a point event.

5. Attribute List (optional). Attributes for this event. Each attribute is semicolon separated and defined as “attribute=value”

After the analysts load both datasets, CoCo identifies all sequences of lengths 1 to 10. The sequences are mined using n-grams. Next, all pairs of non-consecutive events are found in each sequence. The sequences are stored in a dictionary that also counts the number of times each sequence occurs in the datasets.

Next, the metrics are applied to each of the sequences. As each metric is completed, the server returns the set of results to the frontend using Server-Sent Events (SSEs). Though the result list is not shown until all metrics have finished calculating, a table shows progress of which metrics have been calculated, and overview visualizations are shown.

4.2 Guidelines for Scaling HVHT to Large Event Sequence Datasets

Due to the complexity of mining sequences in multiple event sequence cohorts and running hypothesis tests on all of them, many challenges arise dealing with wait times, result set size, and statistical errors. In this section, I describe the five major guidelines for scaling a statistical framework to larger datasets.

Guideline 1: Reduce wait times during computation.

I applied two different methods for computing the hypothesis tests: (1) performing all calculations ahead of time and providing results only when all results are complete, and (2) calculating hypothesis tests by category (e.g., single event frequency, sequence frequency, time gaps, etc) and allowing analysts to see results as they are available.

The first method resulted in long wait times, but allowed the results to be ranked in a more meaningful way. That is, by waiting for all results to be completed, the most "differentiating" or significant results can be displayed first, thus offering more guidance to the analysts about which results are important.

The second method allows analysts to see partial results as soon as they are ready. When a metric is fully calculated, the analysts can select that metric to see all results in that category. In early case studies, this enabled analysts to narrow their focus, although they found that they weren't necessarily interested in specific metrics, just the most major differences – regardless of metric type. The metrics were structured to first calculate the most simple metrics first, e.g., single event metrics, which enabled analysts to understand their datasets on a broader level before going into detailed sequence metrics. Additionally, the metrics should be calculated in an order conducive to both the analysis and optimal time. Through case studies, users common process models were observed. Combined with timing tests to determine the results which are quickest to compute, a recommended computation order is suggested:

- Prevalence – Events
- Prevelence – Non-consecutive pairs
- Time – Consecutive pairs
- Time – Non-consecutive pairs
- Frequency – Events
- Prevalence – Consecutive subsequences
- Record Attributes – prevalence

Guideline 2: Reduce time testing all hypotheses.

Long wait times can cause an analyst to lose concentration and incur more time recalling their task. In an effort to minimize long waits, I implemented a sequence length limit on the sequence mining step in the CoCo pipeline.

The original version of CoCo counted every sequence that appeared in the loaded datasets. However, in the weblog clickstream data, there were some records that had as many as 320 events. Based on observations of the previous three case studies, the analysts often did not look at results of sequences of length more than four. Typically, longer sequences were more obscure and analysts were not able to derive meaningful insights from them. Sequences are limited to length 10, to be adequately long enough for the longer sequences found in clickstreams. In the use of this new limited version, analysts still looked mostly at sequences of length 45 at most, so there was no need to extend the range beyond length 10. The limit was not further reduced because performance at this stage was reasonable. Limiting the sequence length offered a speed up of about 15x. If future datasets would benefit from a shorter limit, I leave determining the ideal limit for future work.

Guideline 3: Minimize data sizes when transferring to browser

Larger datasets require more hypotheses to be tested, thus larger result sets to return to the browser. Aside from the computation time, this results in a much large space requirement. Due to some browser limitations, it is not possible to send data over a certain size. Thus, to reduce the volume of the result set, those sequences that occur in less than 1% of the records are automatically filtered. Additional hypothesis results can be loaded on demand.

Guideline 4: Highlight chance of false positives.

The potential for false positives are highlighted by providing the distribution of p-values to the analysts in a filterable table. Two statistical experts that I consulted with suggested this, because with any statistical test that is applied many times to a single dataset, there is some likelihood of false positives. By providing the analysts the distribution of the resulting p-values, the analysts can see if the actual distribution of p-values is what would be expected by random chance or if it is in fact affected by the content of the dataset.

The chance of statistical uncertainty is further highlighted by providing context to the analysts about each result. For example, by providing related statistic results for the same sequence and showing the prevalence of subsequence results.

Lastly, we place an emphasis on effect size, rather than p-value ,by primarily showing and sorting by the difference and grouping the p-values into broad ranges.

Guideline 5: Enable filtering of event categories unimportant to analysis.

By default, CoCo starts by showing only the results for single event categories (sequences length 1) so analysts can make informed decisions about which events occur frequently and which might be important to the analysis. After determining if any events can be dismissed, analysts can filter out those events that they deem unrelated or unimportant to their questions. In our example case study, the analysts were able to reduce their number of events from over 100 events to under 20.

4.3 Summary

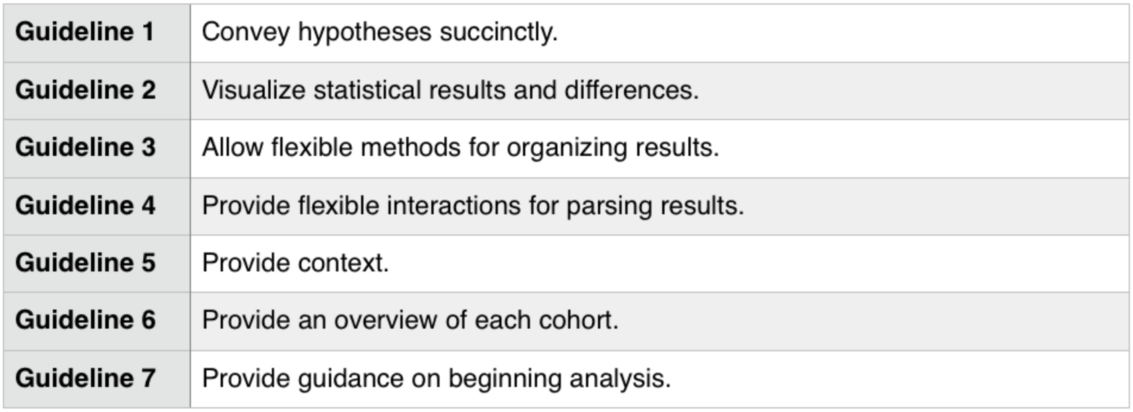
This chapter presents Contribution 2: a statistical framework for performing high-volume hypothesis testing when comparing cohorts of event sequences. The nature of event sequences present unique challenges, including an exponential volume of sequences, increased chance of false positives and statistical errors, and long wait times for analysts to begin analysis. Towards addressing these challenges, I designed a system to better support statistical event sequence comparison tasks and presented an overview of its implementation (Section 4.1). Through case studies, I confirmed the usefulness of these techniques and present the lessons learned as five guidelines for scaling HVHT to large datasets (Section 4.2).

Chapter 5: Design of CoCo: Frontend

After running thousands of hypothesis tests, analysts must then be able to parse through the large result set. In doing so, there are two main challenges: 1) the sheer volume of the result set makes it difficult to identify meaningful and significant results, and 2) when running a large number of statistical tests on a single dataset, the chance of false positives increases.

Through a user study and ten case studies, I aimed to solve these challenges and understand how to leverage the benefits of user-guided exploration to parse high-volume hypothesis results. To develop the initial design and icons, I conducted interviews with three analysts experienced with event sequence visualization: a medical researcher from a local hospital, a graduate student at the University of Maryland, and a business school professor. All had used EventFlow [1] extensively and had active research projects comparing cohorts of patients. An initial version was implemented. After the three analysts had used the initial version with their own data and analytic goals, I interviewed them during a period of a month to collect feedback on the benefits and pitfalls of the initial version, and analysts' needs when reviewing hypothesis results. Feedback was also collected from a eighteen other short-term detailed demonstrations, some of which lead to long-term case studies.

I distill lessons learned into seven design guidelines for balancing automated high-volume hypothesis testing with integrated visualization and interaction (Table 5.1).



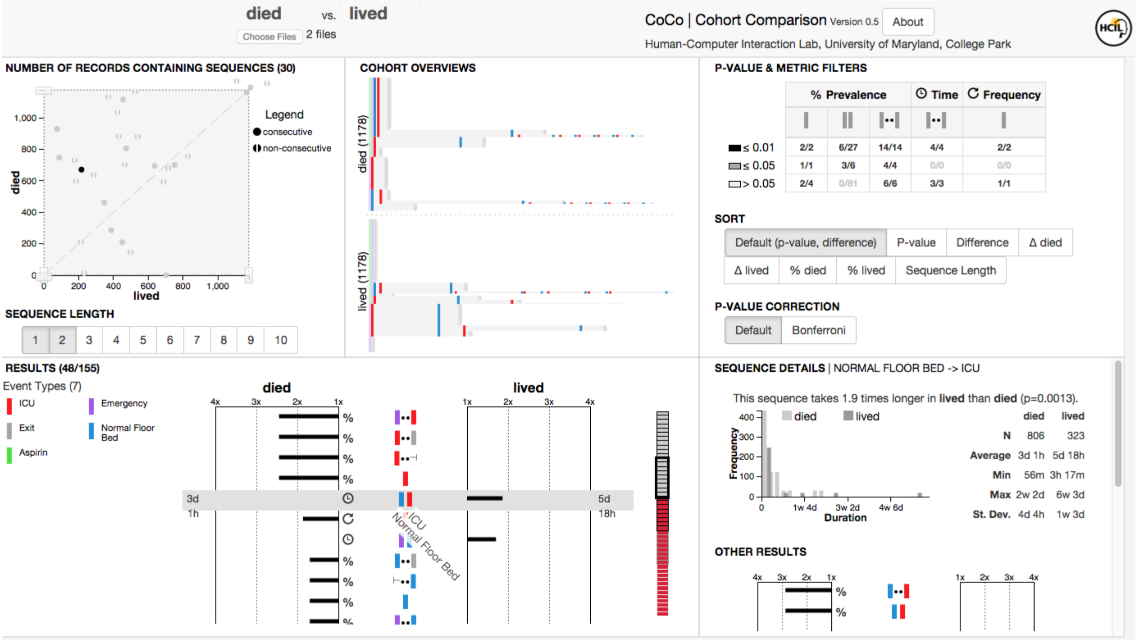
Guideline 1	Convey hypotheses succinctly.
Guideline 2	Visualize statistical results and differences.
Guideline 3	Allow flexible methods for organizing results.
Guideline 4	Provide flexible interactions for parsing results.
Guideline 5	Provide context.
Guideline 6	Provide an overview of each cohort.
Guideline 7	Provide guidance on beginning analysis.

Table 5.1: The 7 Design Guidelines for balancing automated high-volume hypothesis testing with integrated visualization and interaction (Section 5.2).

In this chapter, I provide an overview of the final interface of CoCo. From there, I describe seven design guidelines learned from the design process and the rationale behind the design decisions that led to CoCo's final design. Appendix 8 provides details on previous versions of CoCo and the changes between each itertation.

5.1 Description of the User Interface

CoCo (Figure 5.1) is comprised of five main panels: sequence scattergram, cohort overviews, result filters, results panel, and sequence details.



	% Prevalence			⏲ Time	↻ Frequency
	▮	▮▮	▮••▮	▮••▮	▮
■ ≤ 0.01	2/2	6/27	14/14	4/4	2/2
▨ ≤ 0.05	1/1	3/6	4/4	0/0	0/0
□ > 0.05	2/4	0/81	6/6	3/3	1/1

This sequence takes 1.9 times longer in **lived** than **died** (p=0.0013).

	died	lived
N	806	323
Average	3d 1h	5d 18h
Min	56m	3h 17m
Max	2w 2d	6w 3d
St. Dev.	4d 4h	1w 3d

Figure 5.1: CoCo is comprised of five main panels: sequence scattergram and filters, cohort overviews, result filters, results panel, and sequence details.

5.1.1 Sequence Scattergram and Sequence Filters

The first panel provides an overview of all the sequences in the dataset, as well as a method for filtering the sequences by length and by type (consecutive, non-consecutive). Each dot represents a sequence that is found in the datasets and is placed on the axes according to the number of records that contain that sequence in each cohort. Sequences of length 1 are single event categories. A *consecutive* sequence is one that occurs in the dataset with no events between them, and a *non-consecutive* sequence may contain extra events within it. Consecutive sequences are indicated with a solid black circle and non-consecutive sequences are represented by a circle divided by a white line. Analysts can filter the results based on the sample sizes to exclude rows with very low or very high sample sizes. This can be used as a method for quality control (e.g., removing results with an insufficient sample size) or as a method for segmenting the results into more manageable pieces. For example, analysts may want to evaluate more frequent sequences first (e.g., sequences with 50% or more record coverage), before moving viewing less frequent sequences.

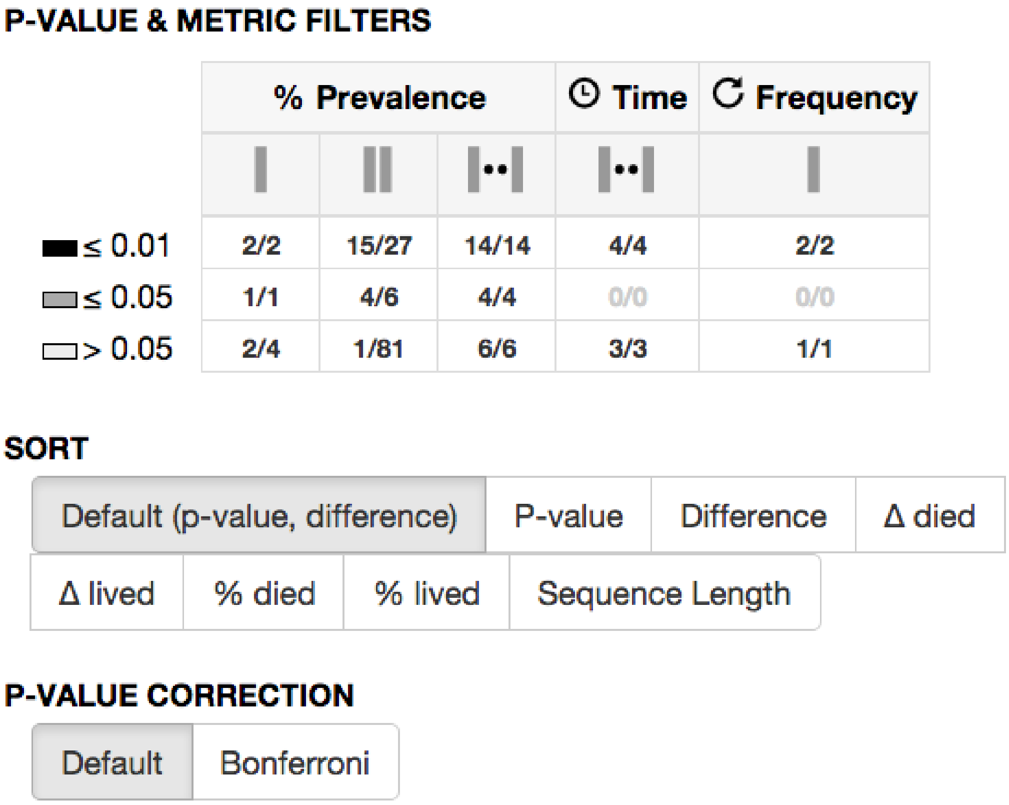
5.1.2 Cohort Overviews

The second panel provides a high-level overview of the sequences contained in each cohort using an EventFlow [1] display. The heights of the cohorts are adjusted in proportion with the number of records in each dataset (e.g., the dataset with more records will take up more vertical space).

5.1.3 Result Filters

The third panel (Figure 5.2) provides methods for filtering, sorting, and correcting the hypothesis test results.

Results can be filtered by two ways using a table that shows the number of hypotheses that were tested according to metric and sequence type. Analysts can choose to see only time, frequency, or prevalence metrics. Similarly, analysts might be interested in only single events, whole record histories, or partial subsequences. The table further breaks down the results based on p-value into three groups: ≤ 0.01, ≤ 0.05, and > 0.05. Each cell contains the total number of hypotheses currently shown out of the total number of hypotheses testing for that metric, sequence type,



P-VALUE & METRIC FILTERS

	% Prevalence			🕒 Time	↻ Frequency
	❙	❙❙	❙••❙	❙••❙	❙
■ ≤ 0.01	2/2	15/27	14/14	4/4	2/2
▨ ≤ 0.05	1/1	4/6	4/4	0/0	0/0
□ > 0.05	2/4	1/81	6/6	3/3	1/1

SORT

[✓ Default (p-value, difference)] | P-value | Difference | Δ died | Δ lived | % died | % lived | Sequence Length

P-VALUE CORRECTION

[✓ Default] | Bonferroni

Figure 5.2: Methods for sorting and filtering the result set. Results can be filtered by two ways using a table that shows the number of hypotheses that were tested according to metric and sequence type. The table further breaks down filtering the results based on p-value into three groups: ≤ 0.01, ≤ 0.05, and > 0.05.

and p-value group.

Analysts can sort the results based on what they find most important:

- Ratio and significance. Sort first by the significance level (p-value in three groups: ≤ 0.01, ≤ 0.05, and > 0.05) then within each group, by magnitude of the difference (descending). This is the default sorting option.
- Significance only. Sorted by the raw p-value (descending).
- Ratio only. Sorted by the absolute ratio (ascending or descending)
- Alpha value. Sorted by the absolute value in alpha (descending).
- Beta value. Sort by the absolute value in beta (descending).

Lastly, analysts can apply a Bonferroni correction [68] to the results.

5.1.4 Results Panel

The main results panel (Figure 5.3) displays all the results of the hypothesis tests according to the sorting and filtering preferences set by the analyst. To the left is a legend which each event category that is found in the dataset, assigned a color.

Each result is encoded as a row, where the center shows the hypothesis that was tested. Colored bars in the center indicate the sequence that the hypothesis refers to and the icons to the left indicate the corresponding metric. Depending on the value of the result, a bar grows out from the center in the direction where the value is larger, on a ratio scale. The bar is then colored by the p-value of the result:

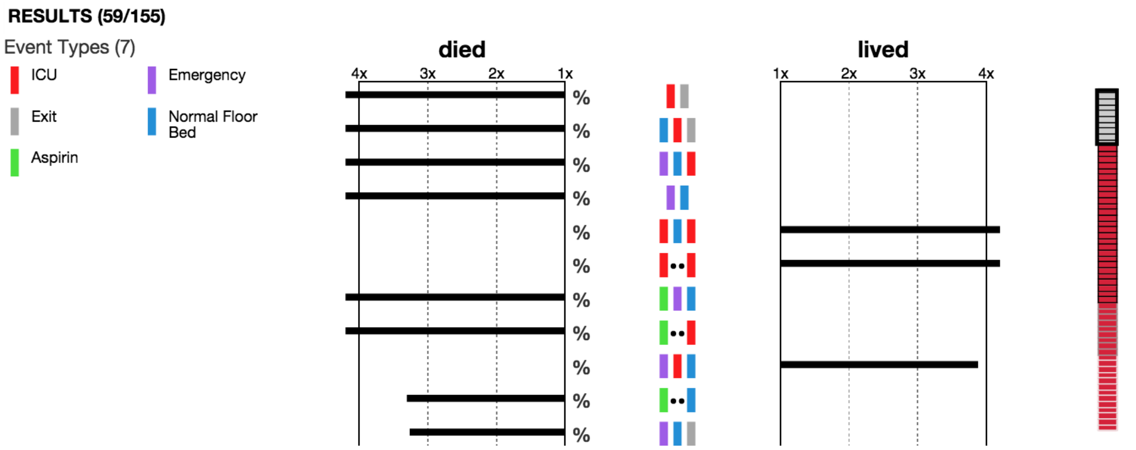


Figure 5.3: The main results panel (Figure 5.3) displays all the results of the hypothesis tests according to the sorting and filtering preferences set by the analysts. To the left is a legend which each event category that is found in the dataset, assigned a color. Each result is encoded as a row, where the center shows the hypothesis that was tested. Colored bars in the center indicate the sequence that the hypothesis refers to and the icons to the left indicate the corresponding metric. Depending on the value of the result, a bar grows out from the center in the direction where the value is larger, on a ratio scale. The bar is then colored by the p-value of the result.

- Black indicates a p-value ≤ 0.01.
- Grey indicates a p-value ≤ 0.05.
- White indicates p-value > 0.05.

When reviewing a large list of results, it is unclear to analysts when everything has been reviewed, especially when they use filtering methods to view smaller pieces of the results at a time. A simple progress bar at the right of the results shows the analysts progress through the result set. It is a heatmap where each result is a single line and color indicates:

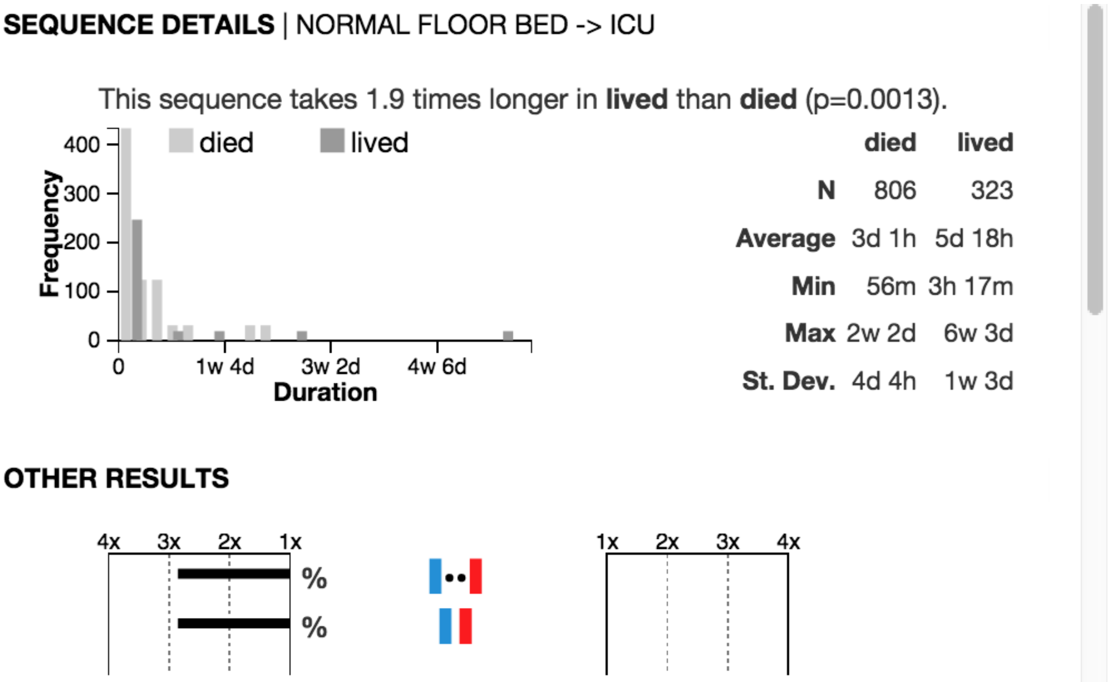
- Grey: result has been reviewed.
- Red: result has been calculated and is not reviewed.

To make it more obvious that the analysts has not missed potentially significant results, CoCo also encodes the p-value using a colored border, matching the above p-value colors.

The progress bar serves as the scrollbar and minimap for the result set. Analysts can page through the data by scrolling along the progress bar. A thickened border indicates the portion of the data that is currently being viewed. The order in the progress bar matches the order of the detailed results and is determined by the analyst, based on the sort options provided.

5.1.5 Sequence Details

Context is given using details on demand. Analysts are able to see the underlying data for a selected result. Depending on the type of metric, analysts will see different information. Because metrics dealing with prevalence are only a matter of percentage, all this data is shown in the result snapshot and the details on demand don't show any additional information. For metrics that show an average (e.g., all time metrics and frequency metrics), the details on demand show the exact distribution for all values (Figure 5.9). Additionally, the details on demand show high-level statistics about the distribution: sample size (n), average, minimum, maximum, and standard deviation.



SEQUENCE DETAILS | NORMAL FLOOR BED -> ICU

This sequence takes 1.9 times longer in **lived** than **died** (p=0.0013).

	died	**lived**
N	806	323
Average	3d 1h	5d 18h
Min	56m	3h 17m
Max	2w 2d	6w 3d
St. Dev.	4d 4h	1w 3d

OTHER RESULTS

Figure 5.4: Analysts can view details about a result by clicking it. Results that correspond to comparing averages (such as average duration or average frequency) will show the distributions of all the values and statistics about the average, minimum, maximum, and standard deviation in both cohorts.

5.2 Design Guidelines for HVHT Visual Analytics Tools

Guideline 1: Convey hypotheses succinctly.

In an initial implementation, the LifeLines2 [69] triangle scheme was used to display event sequences and organized results by metric (e.g., all results dealing with the occurrence of sequences were grouped together; all results dealing with co-occurrences of events were grouped together, etc.). With this organization scheme, the metric selected by the analysts implied a lot about the sequences in its result set and all sequences looked identical. In feedback on this design, many analysts felt that only visualizing the sequence (with no indication of what the hypothesis was), was confusing and they would often have to remember which metric was selected.

I conducted interviews with three domain experts to determine how to distinguish between various event sequence features. In the interviews, each expert was asked how they would visually differentiate the following types of sequences and their properties:

- Whole record sequence
- Concurrent events
- Consecutive vs. non-consecutive sequence

Mockups of responses are shown in Figure 5.5. Analysts suggested differentiating whole record sequences (a) by adding markers indicating the beginning and end of the sequence, to signify no events occur before or after the sequence. Square markers were chosen over the angled brackets to avoid ambiguity with the notion of a "set." Analysts showed concurrent events (b) by either overlapping them or grouping them with a circle. The overlapping method was preferred because it was more compact. Analysts had more variation in how they chose to differentiate consecutive (c) vs. non-consecutive (d) events. Two analysts chose to keep consecutive sequences the same, while differentiating non-consecutive sequences by placing a marker between events. The third analyst suggested the opposite: show no differentiation between non-consecutive events, but place a bar to join consecutively occurring events.

In a later version, the event icons were changed from triangles to slim rectangles to conserve screen real estate (and give analysts the option to toggle between the two versions). The current scheme is shown in Figure 5.6.

Guideline 2: Visualize statistical results and differences.

In designing the result displays, the design needed to convey information about the difference in value (both magnitude and direction) and the statistical significance of the result. Additionally, color was already used to encode the event categories and needed to be avoided. The three methods of visual comparison outlined by Gleicher et al. [6] were tried to encode this data: juxtaposition, superposition, and explicit encoding. Figure 5.8 shows the designs that were considered. Juxtaposition (a) showed the absolute values in each cohort, and worked well for values that had a fixed range (e.g., percentages for 0% to 100%). However, it was not adaptable for variable range values (e.g., time, where a difference can be as small as 1 minute or as large as 3 months) or for displaying time and prevalence metrics in the same view. It is also not ideal for scanning for differences easily because the difference is not explicitly encoded. Superposition (b) has the advantage of displaying the raw values and direction of difference more clearly, but had similar problems to juxtaposition in displaying time and prevalence results on the same axes; because it is axis dependent, it is not possible to display time and percentage in the same view. I found that an explicit encoding only (c) offered the best option by allowing analysts to easily see and interpret differences between the datasets, despite the absolute values in each cohort are obscured. The absolute value information was available using interactions such as hover or details-on-demand to display them.

With the explicit encoding method, it is also able to explore different methods for encoding the differences: absolute difference, relative difference, and ratio. The values in datasets can be categorized in three groups. Take for example, the occurrence of a sequence:

1. Occurs in both datasets the same way (no difference)
2. Occurs in both datasets, but more in one
3. Occurs in only one dataset

Again, providing the absolute difference is sufficient when presenting prevalence results, because percentages are bounded to 100%. However, with results dealing with time, a single scale does not accurately convey differences because 1) time is unbounded, and 2) simply scaling the axis does not always work because even within a single dataset, different time granularity may exist (e.g., a hospital stay is on the order of days whereas a prescription is on the order of months). Relative differences and ratios eliminate the problems of multiple units and granularities, however I found analysts understand ratios more clearly than relative differences. For example, it is easier to interpret "hospitals stays are two times longer in cohort A than cohort B" rather than "hospitals stays are 100% longer." Because ratios can be anywhere from 1 (in case 1) to infinity (case 3), I bound the axis to an analyst-defined maximum (default: 4x). If the ratio is above the maximum, the bar grows off the side, and if it is infinite, an infinity symbol is displayed next to the ratio bar.

Guideline 3: Allow flexible methods for organizing results.

I explored four methods for organizing the results, each with its own benefits:

1. Metric hierarchy. This was the approach taken in the initial version and it worked well in guiding the analysts. Analysts typically started based on sequence length looking at single events before looking at longer sequences, then progressing based on their specific questions. This method worked best when analysts had specific questions about the datasets (e.g., if they were only concerned with whole record sequences).

2. Flattened. In open-ended and unstructured exploration, the analysts do not seem to care about what the metric is, just how important or distinguishing the result is. A flat design displays all hypothesis results in a single list view,

regardless of metric or sequence type, and orders them by the significance and magnitude of difference.

3. Sequence. Some researchers may have questions about a specific sequence of events. For these questions, it is best to group results by event sequence.

4. Metric/flat list hybrid. In this view, the top 10 results for each metric are displayed. A hybrid view will give a good overview of the most important features of the dataset.

In the initial implementation, results were organized based on their category (Method #1). However, interviews with domain experts and analysts indicated that in their analyses, they didn't always care which metric was significant they wanted all the significant results in one place, regardless of what type of metrics they corresponded to. Method #2 is the most flexible for most uses and that analysts can use result filters if they have specific questions dealing with a particular sequence or metric.

Guideline 4: Provide flexible interactions for parsing results.

Displaying large result sets presents challenges in parsing them. Three interaction techniques are provided in Coco for parsing the results. First, with so many hypotheses, not every hypothesis will apply to the dataset or domain. Researchers might have different priorities based on questions they already have. For example, some analysts might only be concerned with whole record sequences, while others want to see patterns across shorter subsequences. Some analysts might be concerned with only metrics dealing with prevalence, whereas others are interested in both time and prevalence metrics. Filtering and sorting provides flexibility by allowing analysts to manage their data based on what is relevant to their questions.

Second, as analysts sort through the results, they might easily disregard some hypothesis given their domain knowledge (e.g., results that are spurious correlations) and would need some way to keep track of everything they care about or have hidden. For this, I suggest simple journaling options: starring, hiding, and annotating (Section 7.2.7).

Lastly, when there are thousands of tested hypotheses, it is difficult for analysts to keep track of how many hypotheses they have viewed, how many are left to view, and of those results that are unviewed, which are significant. A progress bar that indicates analysts' progress through the result set, so analysts feel comfortable that all possibly meaningful results have been reviewed.

Guideline 5: Provide context.

As analysts progress through the result set, it is difficult to understand if a result is meaningful based on a single result, especially when dealing with event sequences. For example, if patients visit the ICU after the emergency room more often in a cohort of patients who died versus lived, it may only be significant because the "ICU" event occurs more often in the cohort of patients who died. CoCo provides details on demand and provides the analysts with an overview of their progression through the result set.

Analysts are able to see the underlying data for a selected result. Depending on the type of metric, analysts will see different information. Because metrics dealing with prevalence are only a matter of percentage, all this data is shown in the result snapshot and the details on demand don't show any additional information. For metrics that show an average (e.g., all time metrics and frequency metrics), the details on demand show the exact distribution for all values (Figure 5.9). Additionally, the details on demand show high-level statistics about the distribution: sample size (n), average, minimum, maximum, and standard deviation.

Guideline 6: Provide an overviews of both cohorts

With a large dataset, an analyst may not know what his or her data looks like. EventFlow displays are embedded for each cohort to provide this overview. EventFlow was chosen because many of the analysts are familiar with it, and its aggregate display provides an overview of the most frequent patterns across the cohorts in a compact view that will scale to large datasets without using more space.

Guideline 7: Provide guidance on beginning analysis

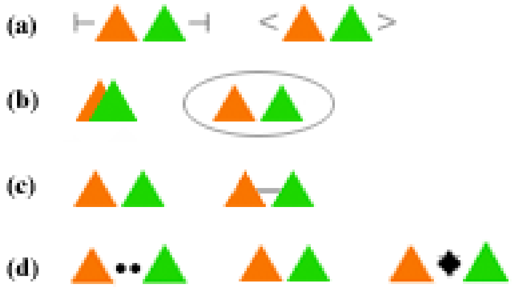
With hundreds of thousands of hypothesis results, it might be daunting for analysts to know where to start with their analysis. To simplify this process, I suggest two methods.

First, follow a recommended process model and arrange the layout to match this process. The panels on CoCo to suggest the order in which analysts should explore their dataset. CoCo first provide methods for seeing an overview of the all the data (scattergram and cohort overviews) on the top left, followed by more detailed views of the result set. Controls for filtering and sorting this list are prominently displayed on the top right.

Second, CoCo provide default values for all result filters and sorting methods. While these result filters are customizable, the default values provide the simplest starting point for the analysts. It is important that the default values are carefully chosen. For example, for sequence length, I decided to start with length 1, since analysts are often overwhelmed after looking for at the long results list. Starting with length 1 allows analysts to get a bearing on the events in their cohorts and allowing them to choose when they are ready to move onto the next result set.

5.3 Summary

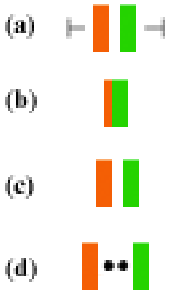
In this chapter, I present Contribution 3: A family of visualizations and guidelines for interaction techniques. High-volume hypothesis testing results in large result sets which are difficult for analysts to parse. Through an interactive user interface, analysts are able to more easily identify important results. I provide an overview of the visual analytics tool, CoCo, and discuss the design decisions that lead to its development. Through case studies with CoCo, I explore the utility of these designs and interactions and distill the lessons learned into seven design guidelines.



(a)

t

Figure 5.5: Analysts' responses.

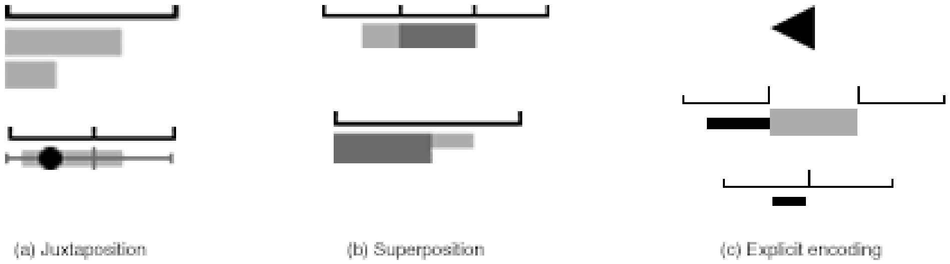


(a)

t

Figure 5.6: Current scheme.

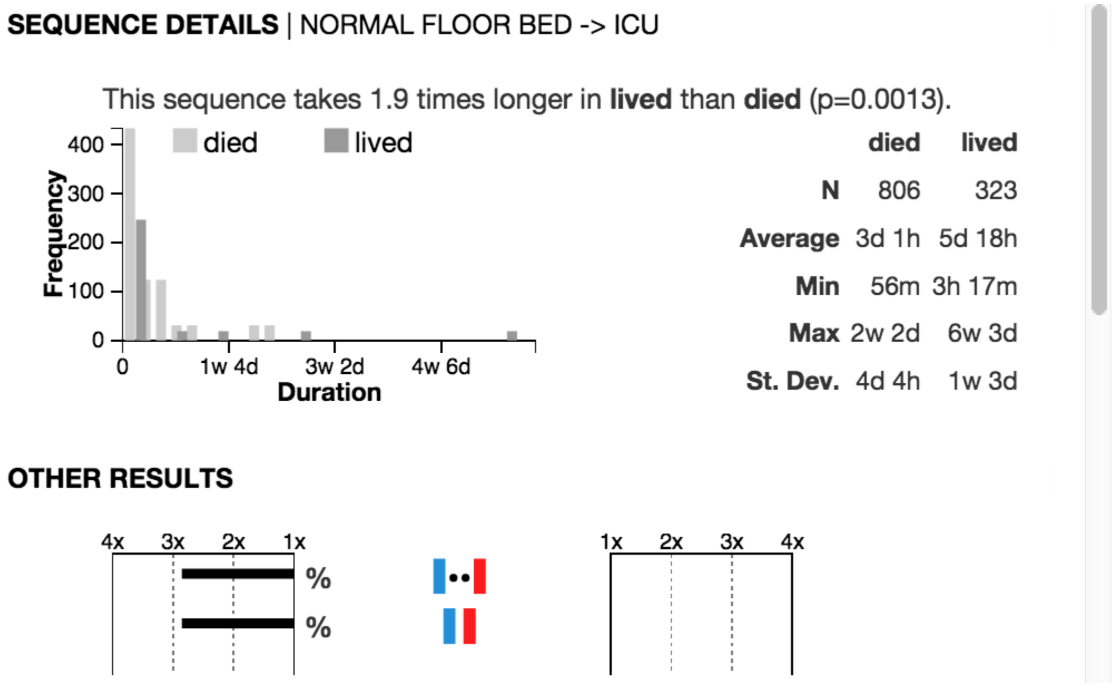
Figure 5.7: Mockups of expert analysts' responses (left) and resulting glyphs (right) for visually differentiating four properties of event sequences: (a) whole record sequences, (b) concurrent events, (c) consecutive sequences, and (d) nonconsecutive sequences.


(a) Juxtaposition

(b) Superposition

(c) Explicit encoding

Figure 5.8: Designs considered for presenting difference results between cohorts and : (a) juxtaposition (directly comparing two bars), (b) superposition (overlaying bars darkened area is the shared amount while the lightened area indicates the difference), and (c) explicit encoding only, which encodes only information about the direction and magnitude of the difference.



SEQUENCE DETAILS | NORMAL FLOOR BED -> ICU

This sequence takes 1.9 times longer in **lived** than **died** (p=0.0013).

	died	**lived**
N	806	323
Average	3d 1h	5d 18h
Min	56m	3h 17m
Max	2w 2d	6w 3d
St. Dev.	4d 4h	1w 3d

OTHER RESULTS

Figure 5.9: Analysts can view details about a result by clicking it. Results that correspond to comparing averages (such as average duration or average frequency) will show the distributions of all the values and statistics about the average, minimum, maximum, and standard deviation in both cohorts.

Chapter 6: Evaluation and Case Studies

6.1 Preliminary User Study

To refine CoCo's design and to observe actual practice of analyzing real-world dataset using a combined CoCo and EventFlow tool, we conducted an early, preliminary user study with volunteers who expressed interest in learning about data visualization and about a new form of statistical analysis.

6.1.1 Method

Our evaluation design was based on the VDAR scenario [70]. More specifically, the goals of our user study were as follows:

1. To learn about the insights users would find.

2. To gain insights into the strategies users would follow.

Before this study, we tested our materials with four participants, where they used either CoCo or EventFlow and we counted the number of insights. We observed that they tended to report everything from CoCo as insights, without considering their actual meanings and importance. In this study, we asked participants to provide further suggestions based on the their insights to guide research at the hospital. As a result, participants were more engaged in the analysis and on average provided 3.5 ($SD = 1.07$) suggestions.

Participants and Settings. We recruited 10 computer science graduate students (7 male, 3 female) through our university's mailing list. The participants' ages ranged from 23 to 29($M = 26, SD = 2.06$). All participants had normal color vision.

We ran CoCo and EventFlow on the same computer. CoCo was displayed on a 1440 × 900 screen while two side-by-side EventFlow windows were displayed on a 1920 × 1200 screen (Figure 1.2).

For simplicity, we began by implementing only a subset of all possible metrics that users may want:

- Prevalence of events, subsequences, and record sequences.
- Duration between sequentially occurring event pairs.
- Prevalence of events and subsequence by attribute.

Procedure. Each 45-minute session included training, data analysis and post-study interview. Training started with a 2-minute introduction on each interface's features. For each interface, participants performed 5 simple tasks and were encouraged to ask questions. We used a pair of synthetic datasets for the training. Questions included clarifying the difference between a "sequence" and a "subsequence" (CoCo), the difference between an event category and attribute (CoCo) and the meaning of gaps between bars (EventFlow) were frequently asked. After the training, all participants said they understood everything.

In the 30-minute data analysis session, a different pair of datasets (representative of hospital room transfer data) were used: patients discharged alive or patients who died. Participants were asked to play the role of a data scientist and analyze the datasets using both CoCo and EventFlow. Their job was to provide insights into the similarities and differences between the paths of the two groups in the hospital. We encouraged thinking aloud and an experimenter took notes of their findings. In particular, we asked them to provide a reason when they switched between CoCo and EventFlow.

During the post-study interview, participants provided comments and reflections about their experience.

6.1.2 Results

During the analysis, no participant asked any interface-related questions and instead concentrated on finding insights. Every participant used both interfaces. In particular, three said they prefer CoCo, while two said they prefer EventFlow. Five expressed no preference. On average, eleven ($SD = 3.67$) insights were reported and four ($SD = 1.12$) interface switches were made per participant. During the interview, all participants stated they wanted to use both interfaces. Below we summarize the results in the context of our user study goals.

Types of Insights. All insights can be categorized into four categories: events, whole record sequences, subsequences and time (Figure 6.1).

Seven out of the ten participants mentioned that it is easier to find subsequence patterns with CoCo while it is easier to find whole record sequence patterns with the side-by-side EventFlow display. They thought EventFlow didn't actually support detecting subsequence patterns because it only showed records as sequences, and they had to visually scan each record to compare subsequences. On the other hand, CoCo specifically provided a metric for subsequences. As a result, significantly more subsequence insights were found by using CoCo than EventFlow ($p < 0.05$) (Figure 6.1).

As for whole record sequences, they preferred EventFlow and stated that it visually encoded the number of records into the heights of color bars, which made it more comprehensive, e.g., *"I just have no feeling about the numbers in CoCo. EventFlow is more interesting."* Meanwhile, six out of the ten participants mentioned that it was hard to compare a specific sequence using the side-by-side windows of EventFlow because they had to search for that sequence on each side separately and had to visually compare them by height, which was not very accurate. In contrast, a participant noted that *"CoCo does the comparison for me."*. There are no significant differences between the two interfaces for the number of insights under the whole record sequence category ($p = 0.39$) (Figure 6.1).

Participants found most of the insights in the event category by using CoCo. This might be related to the fact that CoCo specifically provides a metric for comparing the distribution of events. *"It shows numbers and details."* One participant said when he surprisingly found in CoCo that the *Intermediate Care* event only occurred in the group of patients who died. *"I didn't notice that in EventFlow!"* he added. Also, as the event metric was listed at the top of the metric list in CoCo, all participants started using CoCo by looking at that metric.

As for the time metric, six out of the ten participants mentioned that they liked the way EventFlow combined gaps with sequences together. They also mentioned that in CoCo, they had to switch back and forth between the time metric and the sequence metric to look for insights. *"It provides a better big picture and the time is more visible,"* one participant commented when he was looking at EventFlow to help himself understand the time gaps in the dataset. More insights in the time category were found by using CoCo but it was not significant ($p = 0.25$). One potential reason might be that CoCo was able to average the gap between event pairs, while those pairs were more difficult to find in EventFlow since they are not aggregated and could appear in multiple places in the visualization.

Description of Strategies. During the study, participants were allowed to switch freely between the two interfaces, but we asked participants to describe why they switched.

Nine out of ten participants chose to start with EventFlow. Of these nine participants, three said EventFlow provided a better overview, four said EventFlow seemed simpler and more comprehensive than CoCo (e.g., *"EventFlow is more friendly to my eyes!"*), and the other two said they wanted to look at the actual data with EventFlow before doing the analysis. The only participant who chose to

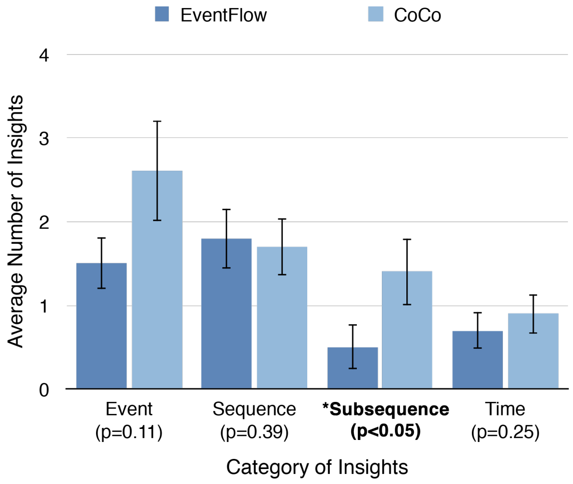


Figure 6.1: Average number of insights per participant per category using EventFlow versus CoCo. The only statistically significant difference ($p < 0.05$) is in insights about subsequences, where participants found more insights using CoCo.

start with CoCo said he liked the statistical summary provided by CoCo. *"It looks like a dashboard,"* he added.

After exploring for five to fifteen minutes, the nine participants who had started with EventFlow switched to CoCo. Three spent about fifteen minutes before making the switch. They explained that they preferred to find everything with one interface first. Four said they got stuck with EventFlow and wanted to get some inspirations from CoCo. Two said EventFlow didn't support comparing subsequences very well, so they temporarily switched to CoCo to do the comparison. The only participant who started with CoCo also switched to EventFlow. He mentioned that he had found an interesting pattern so he switched to EventFlow to have a look at the actual data, which might help to confirm the finding.

After the first switch, all participants were familiar with both interfaces and on average made three more switches. The reasons they provided mainly fell into three categories: (1) switching from CoCo to EventFlow for the overview of event sequences and gaps, (2) switching from CoCo to EventFlow to *"get a sense"* of the patterns they found in CoCo, and (3) switching from EventFlow to CoCo to see the statistical information and to confirm their findings. *"I like the fact that you give me two different tools. I can look at the data in different ways,"* one participant commented at the end of his analysis session.

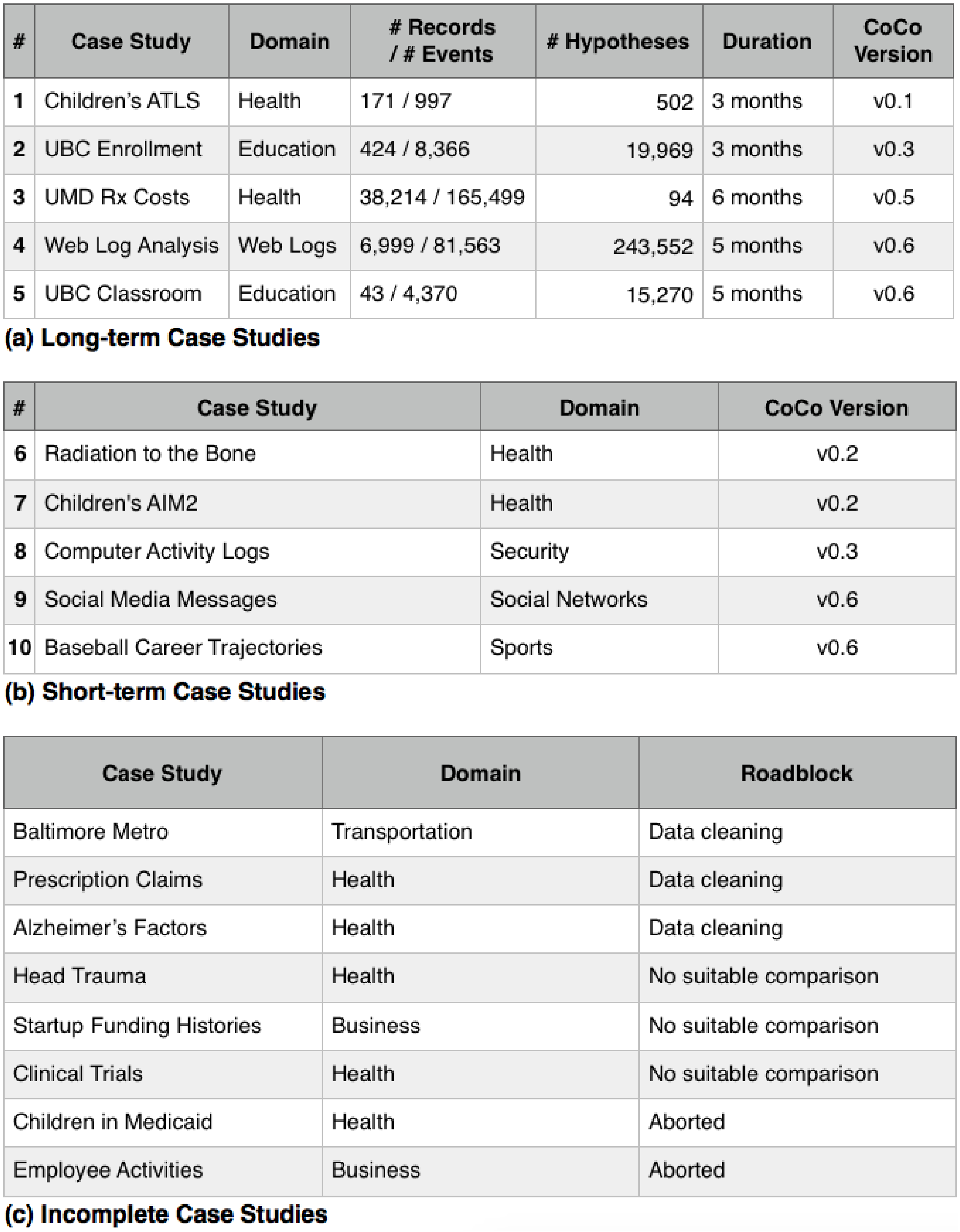
Usability of CoCo. During the post-study interview, all 10 participants stated that training was important to CoCo. In particular, 8 said they could use CoCo skillfully after the training and 2 said they needed more practice. Nine participants liked the visualization of CoCo. The other 1 said he thought traditional scatter plot was more effective. Background knowledge in statistics, especially in p-value was critical to understanding CoCo. Eight participants said the statistics was easy to understand while the other 2 said they didn't get the idea of the p-value. However, the 2 participants added they remembered the rule that black dots were more important than gray or white ones, and it helped a lot in the analysis. 8 participants thought the sequence aggregation feature was useful, because it provided an overview and could show details on demand. One participant commented he seldom expanded it but added that *"It helped me to focus though."*. The other 1 said he had difficulty locating the main sequences after doing expansions. Nine participants liked the layout and navigation of CoCo. However, 2 of them commented the interface was not impressive (e.g., *"The gray background is boring."*, *"It didn't catch my* *eyes.”*). The other 1 disliked the layout because several panels were seldom used in his analysis.

6.2 Case Studies: Introduction

To investigate the strengths and limitations of CoCo as an automated cohort comparison tool, we conducted case studies following the procedure of a Multi-Dimensional, Long-term In-depth Case Study (MILCS) [71]. This methodology was chosen for its emphasis on evaluating the use of the entire system with partners who are real-world analysts using their own data, outside a traditional laboratory study.

The MILCS process begins with understanding the partners’ data, needs, and analysis objectives through an introductory questionnaire and interview. From there, we set a schedule for regular meetings and observation, where the analyst is able to explore their data and provide feedback, while I iterate over the design of the system, fix bugs, and address the analysts’ feedback. At the end of the specified period (which may range from a few weeks to several months), I reflect on the outcomes of the study and lessons learned for both the analyst and my own work.

Fifteen groups expressed interest in using CoCo for their analysis purposes for a total of eighteen case studies. Eight were aborted for various reasons: incompatible problem, lack of time, or the data wasn’t prepared. The remaining eleven were successful case studies performed with CoCo at different stages, over the course of two years. Five of these successful case studies were long-term (CS1 – CS5) and five were short-term (CS6 – CS10). Table 6.2 summarizes all eighteen case studies. Sec-



#	Case Study	Domain	# Records / # Events	# Hypotheses	Duration	CoCo Version
1	Children's ATLS	Health	171 / 997	502	3 months	v0.1
2	UBC Enrollment	Education	424 / 8,366	19,969	3 months	v0.3
3	UMD Rx Costs	Health	38,214 / 165,499	94	6 months	v0.5
4	Web Log Analysis	Web Logs	6,999 / 81,563	243,552	5 months	v0.6
5	UBC Classroom	Education	43 / 4,370	15,270	5 months	v0.6

(a) Long-term Case Studies

#	Case Study	Domain	CoCo Version
6	Radiation to the Bone	Health	v0.2
7	Children's AIM2	Health	v0.2
8	Computer Activity Logs	Security	v0.3
9	Social Media Messages	Social Networks	v0.6
10	Baseball Career Trajectories	Sports	v0.6

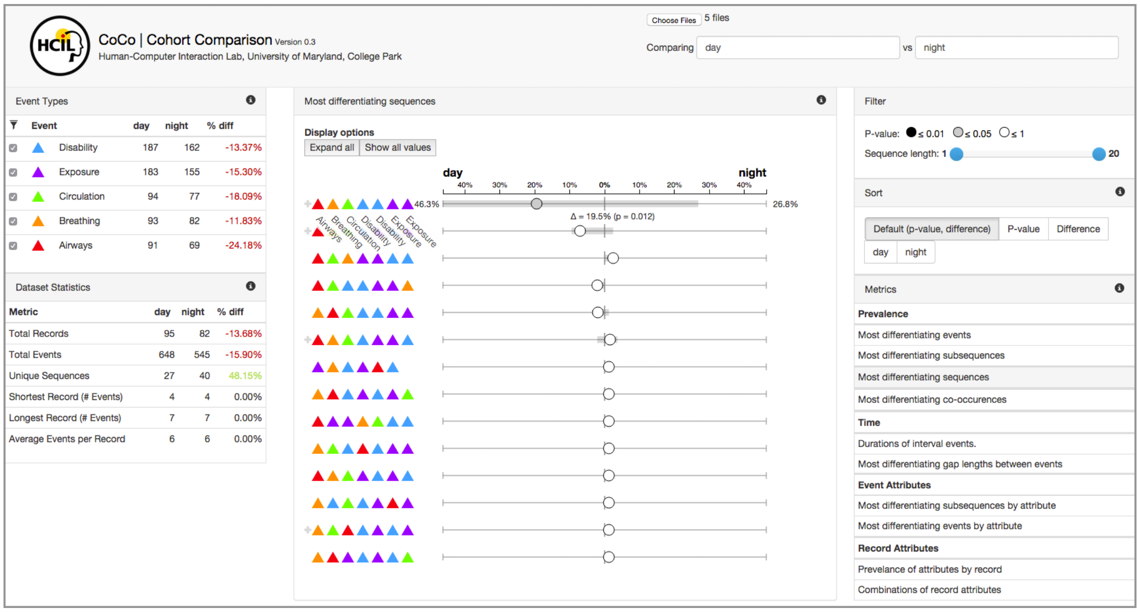
(b) Short-term Case Studies

Case Study	Domain	Roadblock
Baltimore Metro	Transportation	Data cleaning
Prescription Claims	Health	Data cleaning
Alzheimer's Factors	Health	Data cleaning
Head Trauma	Health	No suitable comparison
Startup Funding Histories	Business	No suitable comparison
Clinical Trials	Health	No suitable comparison
Children in Medicaid	Health	Aborted
Employee Activities	Business	Aborted

(c) Incomplete Case Studies

tions 8–6.7 describe the successful long-term case studies in detail, Sections 6.8–6.12 describe short-term, one-time use case studies, and Section 6.13 discusses aborted case studies.

6.3 CS1: Exploring Adherence to Advanced Trauma Life Support Protocol

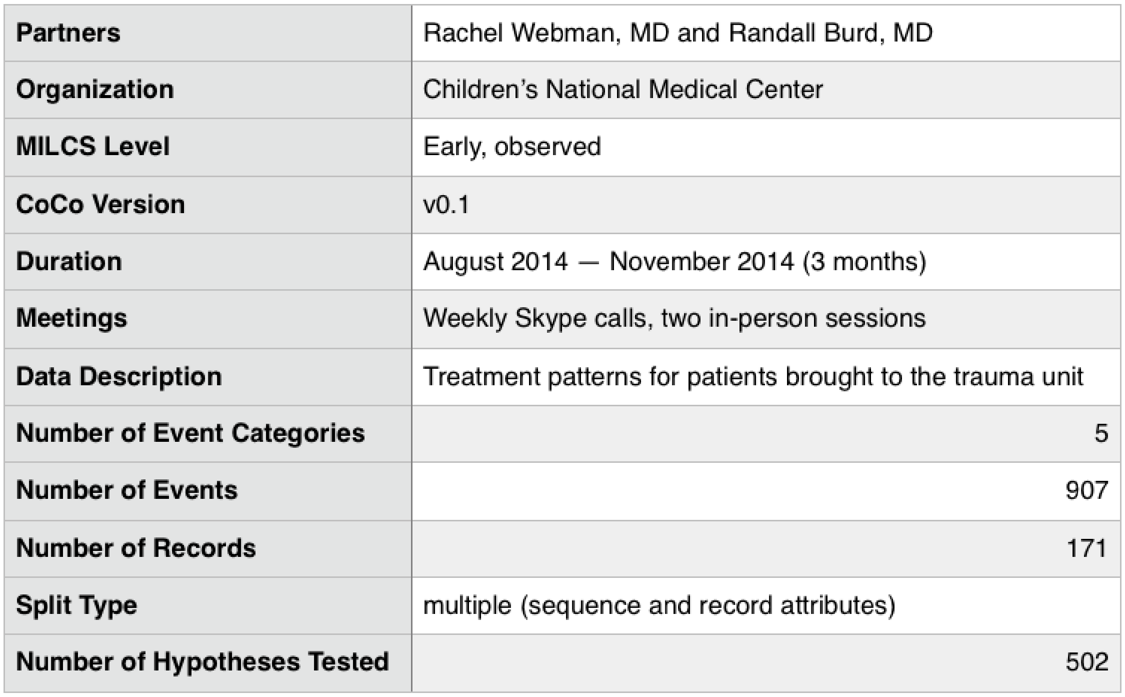

CoCo | Cohort Comparison Version 0.3
Human-Computer Interaction Lab, University of Maryland, College Park
Choose Files
5 files
Comparing
day
vs
night
Event Types
Dataset Statistics
Most differentiating sequences
Display options
Expand all
Show all values
day
night
40%
30%
20%
10%
0%
10%
20%
30%
40%
46.3%
26.8%
Δ = 19.5% (p = 0.012)
Airways
Breathing
Circulation
Disability
Disability
Exposure
Exposure
Filter
P-value: ● ≤ 0.01 ◐ ≤ 0.05 ○ ≤ 1
Sequence length: 1
20
Sort
Default (p-value, difference)
P-value
Difference
day
night
Metrics
Prevalence
Most differentiating events
Most differentiating subsequences
Most differentiating sequences
Most differentiating co-occurences
Time
Durations of interval events.
Most differentiating gap lengths between events
Event Attributes
Most differentiating subsequences by attribute
Most differentiating events by attribute
Record Attributes
Prevelance of attributes by record
Combinations of record attributes

		Event	day	night	% diff
☑	▲	Disability	187	162	-13.37%
☑	▲	Exposure	183	155	-15.30%
☑	▲	Circulation	94	77	-18.09%
☑	▲	Breathing	93	82	-11.83%
☑	▲	Airways	91	69	-24.18%

Metric	day	night	% diff
Total Records	95	82	-13.68%
Total Events	648	545	-15.90%
Unique Sequences	27	40	48.15%
Shortest Record (# Events)	4	4	0.00%
Longest Record (# Events)	7	7	0.00%
Average Events per Record	6	6	0.00%

Figure 6.2: Analysts at Children's National Medical Center used CoCo to understand potentially distinguishing attributes between patients who are treated according to the Advanced Trauma Life Support (ATLS) protocol versus those who are not.

Participants. I worked with Dr. Rachel Webman at Children's National Medical Center, a pediatric care provider in the Washington, D.C. area.



Partners	Rachel Webman, MD and Randall Burd, MD
Organization	Children’s National Medical Center
MILCS Level	Early, observed
CoCo Version	v0.1
Duration	August 2014 — November 2014 (3 months)
Meetings	Weekly Skype calls, two in-person sessions
Data Description	Treatment patterns for patients brought to the trauma unit
Number of Event Categories	5
Number of Events	907
Number of Records	171
Split Type	multiple (sequence and record attributes)
Number of Hypotheses Tested	502

Procedure. After an initial phone conversation to discuss potential analyses, I provided the partners with a demonstration and tutorial of CoCo to Dr. Webman. After converting the data to the CoCo format and installing CoCo on her machine, we meet weekly thereafter to discuss the analysis process, which included data cleaning, visualization (in both EventFlow and CoCo), and data analysis (using Stata). I met with Dr. Webman for two sessions to observe her use of CoCo. The case study ran from August 2014 to December 2014.

Analysis goals. In a previous study [3], they found that about 50% of resuscitations did not follow the ATLS protocol. As a follow-up, the researchers’ were interested in:

1. What percent of patients are treated in adherence to protocol?

2. Are there distinguishing attributes (e.g., time of day, patient gender, team lead) between protocol adherence and non-adherence?

3. What are the most common deviations from the protocol?

Dataset. We began by cleaning the data. Based on the question at hand, the original dataset included many extra event categories and attributes that weren't relevant to the analysis. This initial filtering was done in excel and reduced the number of event categories from 22 to 6. All twenty patient attributes were retained, because they did not add any complexity to the visual display and were important in determining associated attributes for protocol deviation. Patient attributes included injury severity score (ISS), the day of week, length of hospital stay, time between notification and arrival at the hospital, and if the patient was admitted to the hospital, among others.

Next, the data was displayed in EventFlow. Further data manipulation was performed here. Because Coco only accepts point events, we converted the secondary scan interval event into a single point event representing the start of the interval. Additionally, there were two types of pulse events which were merged into a single event. Lastly, 7 records that had inconsistencies in the dataset, such as the secondary scan ending before it begins, or the patient arriving after other events had happened were removed after verifying the data in the original data sheets.

The resulting dataset consisted of 171 patient records, with event categories for the five steps in the ATLS protocol: airway evaluation, listening for breath sounds, assessment of circulation, evaluation of neurological status disability, and temperature control.

6.3.1 System Use

The first observational session was three hours. Over the course of the 3 hours, we split the dataset in six ways to load six different pairs of cohorts in CoCo as they explored different hypotheses:

1. Patients treated in adherence to the ATLS protocol versus those that showed any deviation.

2. Patients admitted to the floor versus ICU (with discharged patients removed).

3. Resuscitations where the trauma team received at least 5 minutes advanced notice and teams where there was fewer than 5 minutes notice ("now" resuscitations).

4. Patients with a high (> 15) versus low ISS.

5. Patients treated on the weekend versus on a weekday.

6. Patients treated during the day versus at night.

In every comparison group, the analysts began by looking at the prevalence of single events, to determine how often they occurred. The analysts then looked at the most differentiating *entire record* sequences, because the subsequences were less informative about how the protocol was followed. They would then make their way down the provided metrics list, in the order that they appeared: most differentiating time gaps and then prevalence of record attributes. They did not look at the prevalence of record attribute combinations for any of the datasets.

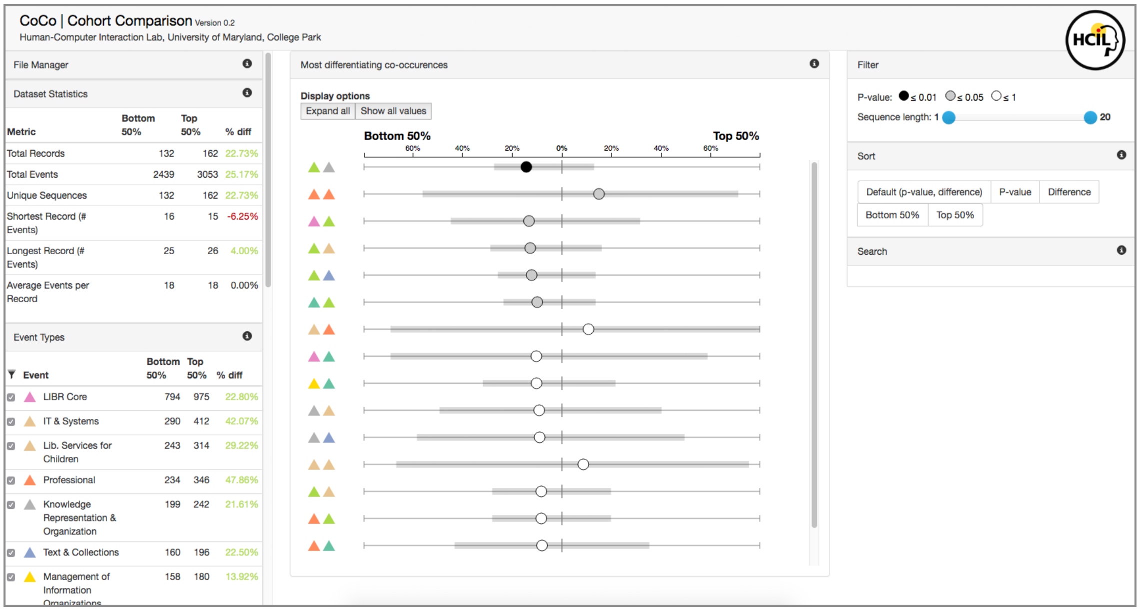
6.3.2 Outcomes

For analyst. For this dataset, the analysts expected to see that each patient record contained every event category. However in two of the comparisons, (1) correctly treated patients versus those with deviations and (2) day versus night patients, the latter of both groups received the airway check significantly less than former. In the day versus night group, the analyst also found that the "most differentiating sequence" was the *correct* order, meaning that the nighttime patients were treated in the correct order significantly less often than daytime patients. Additionally, patients treated at night had more variance in the procedure, with 26 unique sequences in the 83 patients versus 20 unique sequences in the 101 daytime patients. A possible reason for this finding is that during the day, nurse practitioners perform these procedures, but at night, junior residents, who may have less experience with this type of task at this particular institution, are on-call instead. From these results, the analysts presented these findings at an internal symposium on pediatric care and a city-wide conference on trauma care.

For CoCo. In the closing interview, one analyst said, *"We don't need to solve everything with EventFlow and CoCo. These tools let us explore the data and narrow our hypothesis."* This early case study was indication that CoCo can be effective for exploratory analysis and hypothesis generation. Additionally, through observation of the analyst, we were able to develop the basis for a process model that was implemented and tested in later versions of CoCo.

Overall, the analysts noted that CoCo was useful for their needs, and without it, the analysis would have been possible, but much more difficult to perform. One area that CoCo could be improved is to include multivariate analysis, as this is a central need for their analyses.

6.4 CS2: Student Course Enrollments



Metric	Bottom 50%	Top 50%	% diff
Total Records	132	162	22.73%
Total Events	2439	3053	25.17%
Unique Sequences	132	162	22.73%
Shortest Record (# Events)	16	15	-6.25%
Longest Record (# Events)	25	26	4.00%
Average Events per Record	18	18	0.00%

	Event	Bottom 50%	Top 50%	% diff
☑	▲ LIBR Core	794	975	22.80%
☑	▲ IT & Systems	290	412	42.07%
☑	▲ Lib. Services for Children	243	314	29.22%
☑	▲ Professional	234	346	47.86%
☑	▲ Knowledge Representation & Organization	199	242	21.61%
☑	▲ Text & Collections	160	196	22.50%
☑	▲ Management of Information Organizations	158	180	13.92%

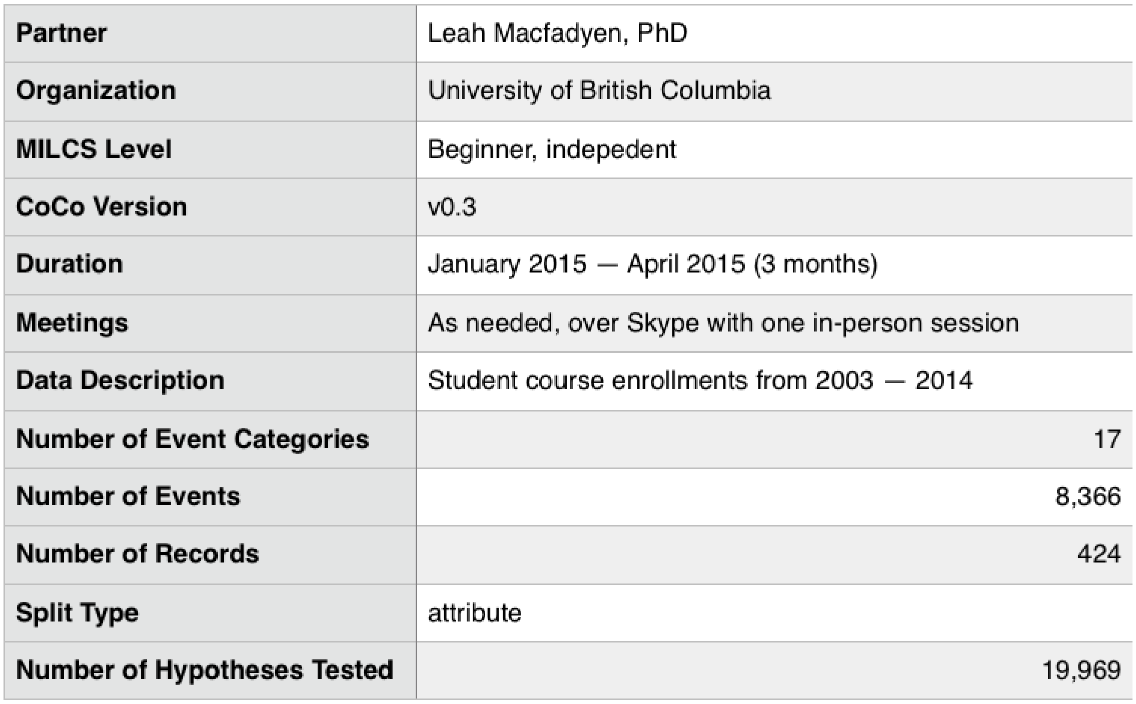
P-value: ● ≤ 0.01 ○ ≤ 0.05 ○ ≤ 1

Sequence length: 1 20

Figure 6.3: An analyst at the University of British Columbia (UBC) was interested in using CoCo to better understand the pathways UBC's students typically pursue towards degree completion

Participants. I worked with Dr. Leah MacFadyen who is the Program Director of Evaluation and Learning Analytics in the Faculty of Arts at the University of British Columbia (UBC).

Procedure. All sessions, except one, were conducted remotely, through Skype and screensharing. These sessions were at irregular intervals that were scheduled as needed. The sessions would consist of troubleshooting, guidance on how to use CoCo, feedback from Dr. MacFadyen on her experience, as well as observation of how CoCo was used. In between these sessions, Dr. Macfadyen used CoCo inde-



Partner	Leah Macfadyen, PhD
Organization	University of British Columbia
MILCS Level	Beginner, indepedent
CoCo Version	v0.3
Duration	January 2015 — April 2015 (3 months)
Meetings	As needed, over Skype with one in-person session
Data Description	Student course enrollments from 2003 — 2014
Number of Event Categories	17
Number of Events	8,366
Number of Records	424
Split Type	attribute
Number of Hypotheses Tested	19,969

pendently and would regularly email her thoughts, experiences, results, questions, and requests related to CoCo.

Analysis goals. The University of British Columbia (UBC) permits students to register in and complete courses towards their degree without a required or prespecified order. Some temporal ordering in course enrollments is imposed by requiring "core" courses to be completed first or by requiring pre-requisites. Beyond these constraints, however, flexibility of enrollment results in a complex and highly heterogeneous record of student enrollment patterns.

Dr. Macfadyen was interested in using both EventFlow and CoCo to better understand the pathways UBC's students typically pursue towards degree completion. Major research questions included:

- Are some enrollment pathways more common than others?
- Are some course sequences more frequently associated with success in a given degree program or specialization?
- Do certain course combinations or sequences seem to channel students towards or away from Majors or Honours programs?
- Which course sequences have the highest rates of attrition?

In understanding student enrollment (and dropout) patterns over time could, the results could inform curriculum, course planning, and student advising.

Dataset. The dataset consisted of the course enrollment records and selected demographic and graduation data of 796 students in the enrolled in three departments in the School of Library and Information Science (iSchool):

- Masters of Library and Information Studies (MLIS),
- Masters in Archival Studies (MAS), and
- a joint Library and Information Studies/Archival Studies Degree Program (MASLIS).

The course enrollments were all in the period 2004-2013. Each event category was a course the student had enrolled in, aggregated by department (e.g., Archival Studies, Information & Society, etc). Record attributes imported from the student records included:

- Degree program: MLIS, MAS, MASLIS
- "Grad Group:" 1 = bottom 50% by graduation average grade, 2 = top 50% by graduation average grade
- International/Domestic status
- Student citizenship
- Student gender

The iSchool was interested in enrollment pathway differences between male and female students, domestic and international students, and any relationship between course enrollment patterns and student performance, as represented by weighted average on graduation.

6.4.1 System Use

Three paired sets of data for exploration were then exported from these data sets:

1. MLIS students distinguished by gender
2. MLIS students differentiated by achievement (graduation group 1 or 2)
3. iSchool students differentiated by degree program (MLIS vs MASLIS)

6.4.2 Outcomes

For analyst. Dr. Macfadyen was able to make numerous insights about the students' enrollment behavior.

1. Gender differences in MLIS student course enrollment choices. Female students are over-represented in the MLIS cohort by a ratio of 2:1 (as in the Faculty of Arts as a whole). CoCo analysis suggested that female students are significantly more likely to complete courses in Library Services for Children ($p = .002$) and Professional courses ($p = .036$).

Analysis of most differentiating subsequences suggested that male students are more likely to complete multiple IT & Systems courses alongside their LIBR core courses, while female students are more likely to combine LIBR core courses with Library Services for Children courses.

In line with these observations, analysis of "most differentiating co-occurrences" showed that female students are significantly more likely to combine Library Services for Children courses (yellow) with selected other courses.

2. Top 50% versus Bottom 50% of MLIS students by graduation GPA. The only moderately significant ($p < 0.05$) differentiating event (course enrollment) between these two groups is that the lower-achieving group are more likely to have enrolled in Information & Society courses. There were no significantly different sequences or subsequences of courses were observed between the two groups, and co-occurrence of two Professional courses is significantly more likely ($p = .012$) in the higher-achieving group.

3. Comparing course enrollment choices of MLIS and MASLIS students. For these comparisons, Archival Studies (ARST) courses were excluded, since MLIS students do not complete ARST courses. Analysis of most differentiating events indicates that MASLIS students are significantly more likely to complete Library Services and Texts & Collections courses than MLIS students ($p < .01$).

Meanwhile, a range of co-occurring event combinations involving Text & Collections and Professional courses are observed significantly more frequently for MASLIS students.

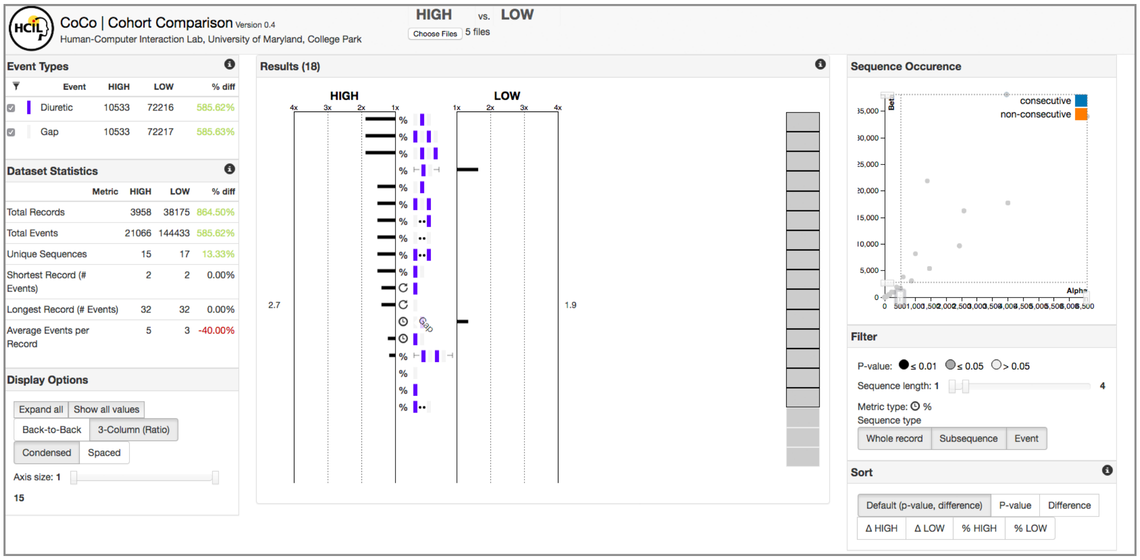
For CoCo. Because students take multiple courses on a semesterly basis, the dataset naturally contained many concurrent events. This was CoCo's first major case study that involved concurrent events and because of this, this case study was the single most helpful in determining the requirements for dealing with concurrent events. On the backend, this introduced the need for an altered datastructure and new metrics.The data structure only provided a minor change: instead of including all events in a flat list, the list was converted to nested lists, where events are bucketed by timestamp.

The addition of new metrics provided more challenges in dealing with how concurrent events should be treated in the context of subsequences. First, when two events occur at the same timestamp, it is unclear whether to count it as a sequence of length 1 or length 2, because as there is only one timestamp, this may not necessarily be considered a “sequence.” We chose to count this as a sequence of length 1, because it provides additional insight into how often two events occur concurrently, versus on its own. Second, it is not immediately clear how to handle concurrent events in the context of subsequences. For example, suppose we have the sequence (AB)D, where A and B occur concurrently and are followed by D. Depending on the analysis, it may or may not be necessary to count the sequence as an instance of “AD” or “BD” on its own. We added the new metrics to the taxonomy, but leave their implementation for future work.

On the frontend, CoCo needed to be adjusted to show sequences that have overlap. Section 5.2 shows our method for determining how to represent concurrent events.

Along the way, Dr. Macfadyen also provided valuable feedback for usability and bug fixes, such as allowing customizable colors and adjusting the display for multiple screen sizes.

6.5 CS3: Medication Adherence Patterns of Hypertension Patients



Event Types

	Event	HIGH	LOW	% diff
☑	Diuretic	10533	72216	585.62%
☑	Gap	10533	72217	585.63%

Dataset Statistics

Metric	HIGH	LOW	% diff
Total Records	3958	38175	864.50%
Total Events	21066	144433	585.62%
Unique Sequences	15	17	13.33%
Shortest Record (# Events)	2	2	0.00%
Longest Record (# Events)	32	32	0.00%
Average Events per Record	5	3	-40.00%

Axis size: 1

15

Filter

P-value: ● ≤ 0.01 ● ≤ 0.05 ○ > 0.05

Sequence length: 1 4

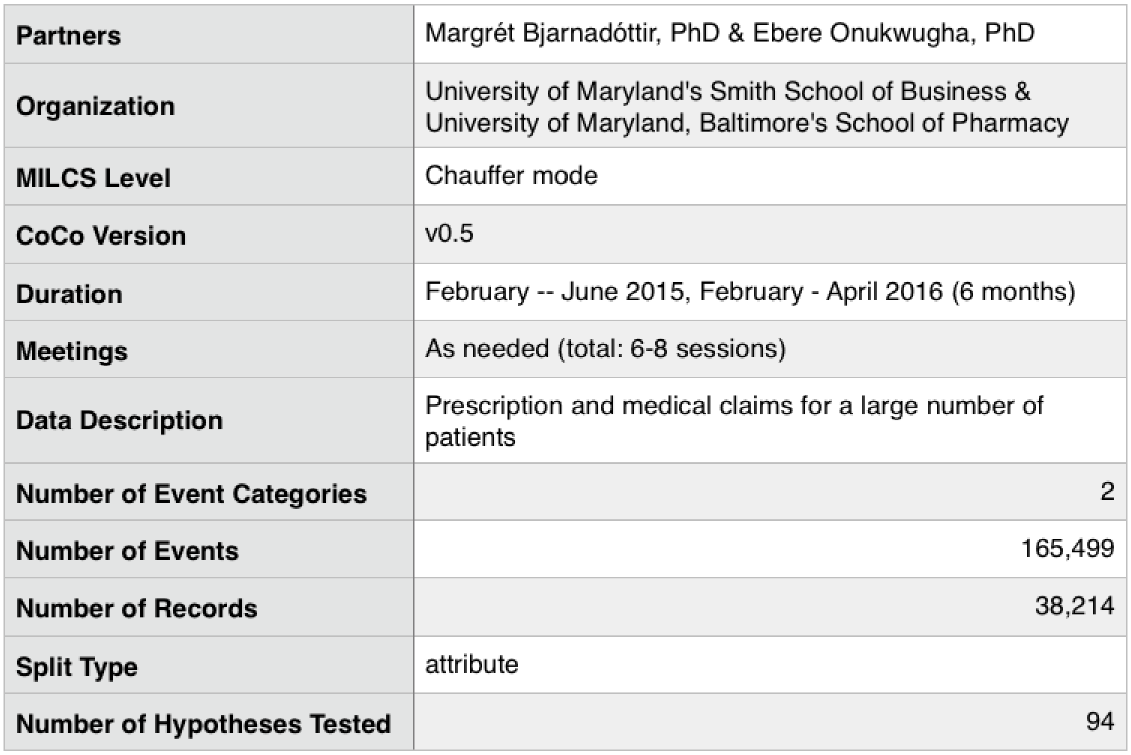
Metric type: ⏲ %

Sequence type

Whole record | Subsequence | Event

Figure 6.4: Researchers at the University of Maryland used CoCo to compare whether drug adherence affected the cost that patients incurred over a year. In other words: Could taking medication as prescribed result in lower overall medical costs?

Participants. I worked with Dr. Margrét Bjarnadóttir and Dr. Eberechukwu Onukwugha for this case study. Dr. Bjarnadóttir is a Professor at the Smith School of Business specializing in operations research methods using large scale data. Dr. Onukwugha is a Professor at the Department of Pharmaceutical Health Services Research and specializes in cost-effectiveness analysis, health disparities, and medical decision-making by individuals.



Partners	Margrét Bjarnadóttir, PhD & Ebere Onukwugha, PhD
Organization	University of Maryland's Smith School of Business & University of Maryland, Baltimore's School of Pharmacy
MILCS Level	Chauffer mode
CoCo Version	v0.5
Duration	February -- June 2015, February - April 2016 (6 months)
Meetings	As needed (total: 6-8 sessions)
Data Description	Prescription and medical claims for a large number of patients
Number of Event Categories	2
Number of Events	165,499
Number of Records	38,214
Split Type	attribute
Number of Hypotheses Tested	94

Procedure. The analysis was done in chauffer-mode, with me using Coco and being advised by Dr. Bjarnadóttir and Dr. Onukwugha based on the results.

Analysis goals. The researchers were analyzing the medication adherence patterns of patients on diuretics (i.e. are patients taking their drugs as prescribed, in which combinations, what characterizes the gaps between prescriptions, etc.). In particular they are interested in the differences between high-cost versus low-cost diuretics patients and want to know what patterns are representative of each group.

The researchers wanted to compare whether drug adherence affected the cost that patients incurred over a year. In other words: Could taking medication as prescribed result in lower overall medical costs?

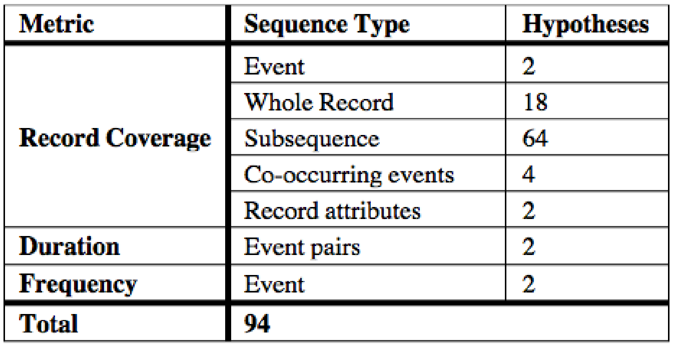
Current methods for adherence analysis consist merely in calculating a Medication Possession Ratio (MPR) [72] or similar aggregated measures that do not represent the diversity of patterns found in the data. The MPR for a pre-defined period is calculated as:

$$MPR = \frac{\text{number of days prescribed}}{\text{days elapsed over period}}.$$

For example, if patients only refilled one 30-day prescription over a period of 90 days, their MPR is 30/90, or 1/3. This method oversimplifies a patient's prescription history into a single number, which may not provide an accurate representation. A patient might refill prescriptions early when planning to leave on vacation, thus leaving a larger-than-usual gap in their refill history, when in fact they were taking their medication regularly. Conversely, a patient who switches to another medication after a recent prescription refill may have a history that incorrectly indicates that the patient regularly took their prescription.

Dataset. The data these researchers gathered consisted of prescription refill histories of five drugs commonly used to treat hypertension. The data spanned one year and contained over 1 million patients. The data also included the total cost of all prescription costs over the year.

We report here only on the analysis of the adherence patterns of patients who took medications from only one drug class: diuretics, which consisted of a total of 113,401 patients. The dataset consisted of two event categories: diuretic and gap, where diuretic indicated the start time of a prescription and gap indicated the start time of no medication usage. The patients were categorized into "HIGH"



Metric	Sequence Type	Hypotheses
Record Coverage	Event	2
	Whole Record	18
	Subsequence	64
	Co-occurring events	4
	Record attributes	2
Duration	Event pairs	2
Frequency	Event	2
Total	**94**	

Table 6.1: Number of hypotheses generated by metric and sequence type.

versus “LOW” cost patients based on the distribution of prescription costs for the patients. Patient costs ranged from $0 to $9,528 (USD). Most patients (55%) had no prescriptions costs and the average cost was $25.39. We excluded patients with $0 costs and patients with more than $380 (top %1), to exclude outliers due to multiple prescriptions or medical costs that were not associated with hypertension (e.g., automobile accident).

The final dataset consisted of 3,958 patients categorized as HIGH cost and 38,175 patients categorized as LOW cost.

6.5.1 System Use

The third version of CoCo was used to compare prescription patterns of high- versus low-cost patients. In total, CoCo generated results for a total of 94 hypotheses. The hypotheses are broken down by metric and sequence type in Table 6.1.

The analysts first used the Sequence Occurrence panel to review only results with a sufficient sample size. The threshold was set at 10% of each cohort, or 395 in the HIGH cost group and 3,817 in the LOW cost group, which reduced the number of hypotheses to review to 24. Next, the remaining insignificant results ($p > 0.05$) were removed using the Filter by Significance feature, leaving a more simplified display of 21 results.

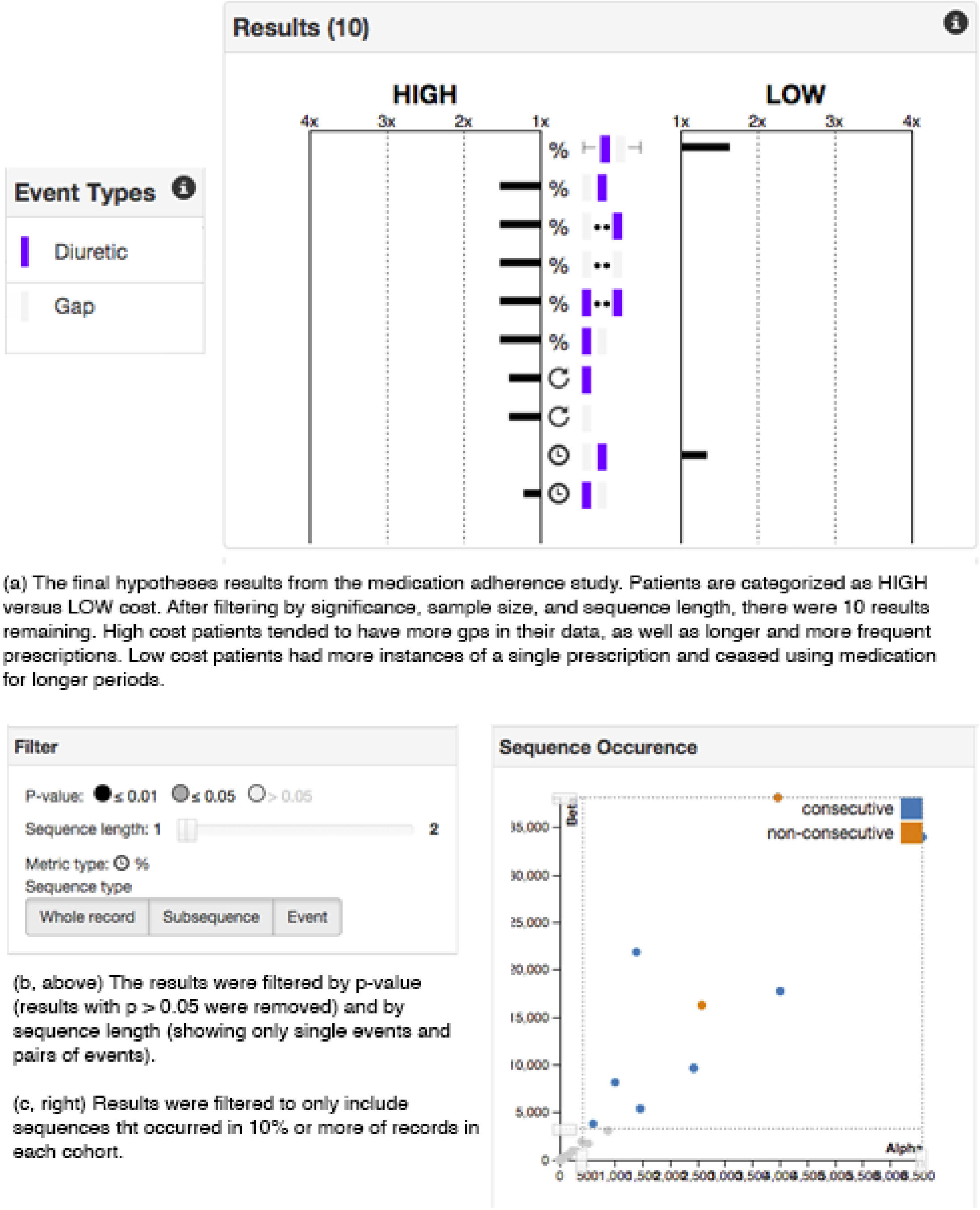
Finally, the results were Filtered by Sequence Length, to view only sequences of length 1 (single events) or 2 (event pairs). Because there are only two event categories in the dataset, longer sequences were just repetitions of length 2 or less, so this was all that was necessary to view all unique patterns. Thus, there were 10 remaining hypotheses to review in detail. The final result display and settings are shown in Figure 6.5.

The analysts then evaluated the remaining hypotheses one by one, using context information provided in the details on demand panel.

6.5.2 Outcomes

For analyst. This made it easy to conclude that high-cost patients tended to have longer sequences, with more gaps and prescription refills, whereas low-cost patients had shorted sequences, most commonly filling only a single prescription. Low-cost patients also took significantly longer gaps between prescription refills. As a follow-up, analysts will incorporate medical claims data to understand the more serious medical implications of medication adherence, such as heart attacks or stroke.

For CoCo. Better understanding on having all metrics in a single view. First testing of prescribed analysis process. This case study provides an illustrative example of the challenges that researchers and analysts encounter, and describes how



(a) The final hypotheses results from the medication adherence study. Patients are categorized as HIGH versus LOW cost. After filtering by significance, sample size, and sequence length, there were 10 results remaining. High cost patients tended to have more gps in their data, as well as longer and more frequent prescriptions. Low cost patients had more instances of a single prescription and ceased using medication for longer periods.

Filter

P-value: ● ≤ 0.01 ● ≤ 0.05 ○ > 0.05

Sequence length: 1 2

Metric type: %

Sequence type

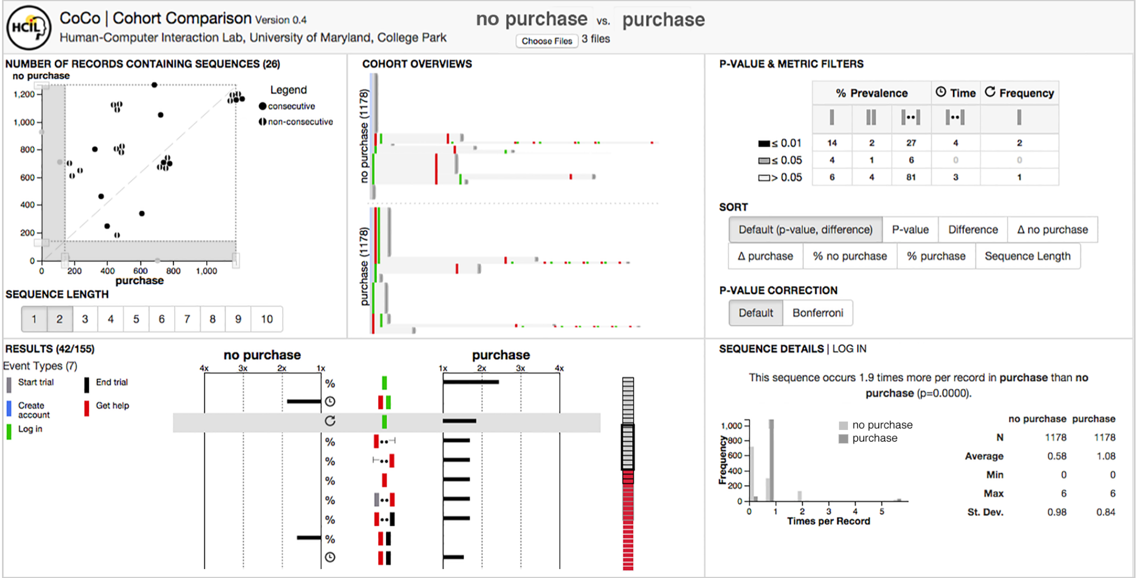
(b, above) The results were filtered by p-value (results with p > 0.05 were removed) and by sequence length (showing only single events and pairs of events).

(c, right) Results were filtered to only include sequences tht occurred in 10% or more of records in each cohort.

Figure 6.5: Final results and usage of drug pattern case study. Analysts used the Sequence Occurrence panel (c) to control sample size, and the Filter panel (b) to control significance and sequence length. This resulted in only 10 hypotheses (a) for the researchers to manually review.

the implementation of new visualization interaction techniques for event sequence hypotheses in CoCo enables the automatic analysis of two groups of records.

6.6 CS4: Customer Web Logs



	% Prevalence			Time	Frequency
■ ≤ 0.01	14	2	27	4	2
▨ ≤ 0.05	4	1	6	0	0
□ > 0.05	6	4	81	3	1

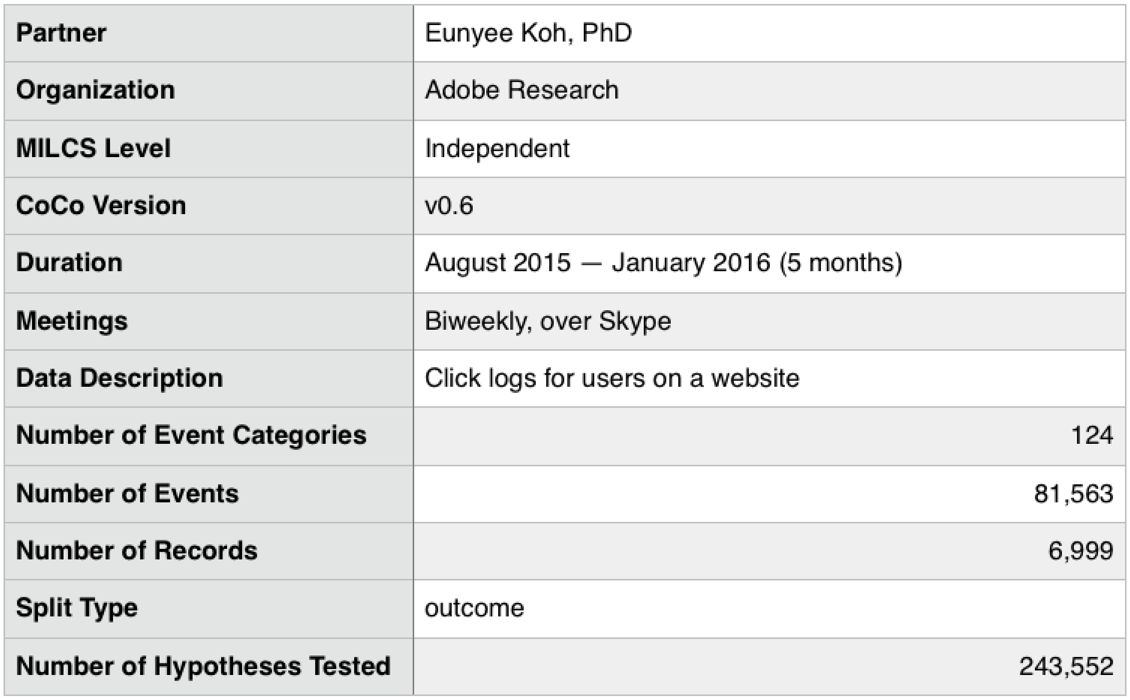
This sequence occurs 1.9 times more per record in **purchase** than **no purchase** (p=0.0000).

	no purchase	purchase
N	1178	1178
Average	0.58	1.08
Min	0	0
Max	6	6
St. Dev.	0.98	0.84

Figure 6.6: Analysts at Adobe were interested in comparing user click logs using CoCo to understand which events lead to a product purchase versus don't.

Participants. I worked with Dr. Eunyee Koh, a Senior Research Scientist at Adobe Research. Her research focuses on semantic analysis and metadata extraction from media, and how to visualize those extracted metadata interactively for people.

Procedure. I worked with Dr. Koh to train her on the use of CoCo and form the objective on the analysis. We met biweekly over Skype, where she provided feedback dealing with the scalability of CoCo. After becoming an advanced user of CoCo, Dr. Koh then performed independent evaluations of CoCo with two analysts at Adobe.



Partner	Eunyee Koh, PhD
Organization	Adobe Research
MILCS Level	Independent
CoCo Version	v0.6
Duration	August 2015 — January 2016 (5 months)
Meetings	Biweekly, over Skype
Data Description	Click logs for users on a website
Number of Event Categories	124
Number of Events	81,563
Number of Records	6,999
Split Type	outcome
Number of Hypotheses Tested	243,552

Analysis goals. The analysts were interested in understanding user behaviors and exploring the data in a free-form way.

Dataset. The dataset contained users' events on a product website, such as viewing the display ads, signing up for promotions or free trials, and purchasing products.

6.6.1 System Use

All three analysts used the same dataset to compare the group of users who purchased the products without using trials versus with using product trials. In particular, the analysts explored the occurrence of the display ads and retargeting events (e.g., an ad for a product the user has already viewed) between the two cohorts. By exploring events that are statistically significant in the result panel, analysts found one group viewed display ads more than the other group, and that group also contained more retargeting events than the other group. By investigating more on other events such as product trial and adoption using CoCo, analysts hypothesized that the first group, who viewed the display ads more, seemed fairly new to the websites' product offerings ("explorers") while the other group, who were exposed to fewer display ads and retargeting, seemed to have good knowledge about the websites' products and offerings ("experience users").

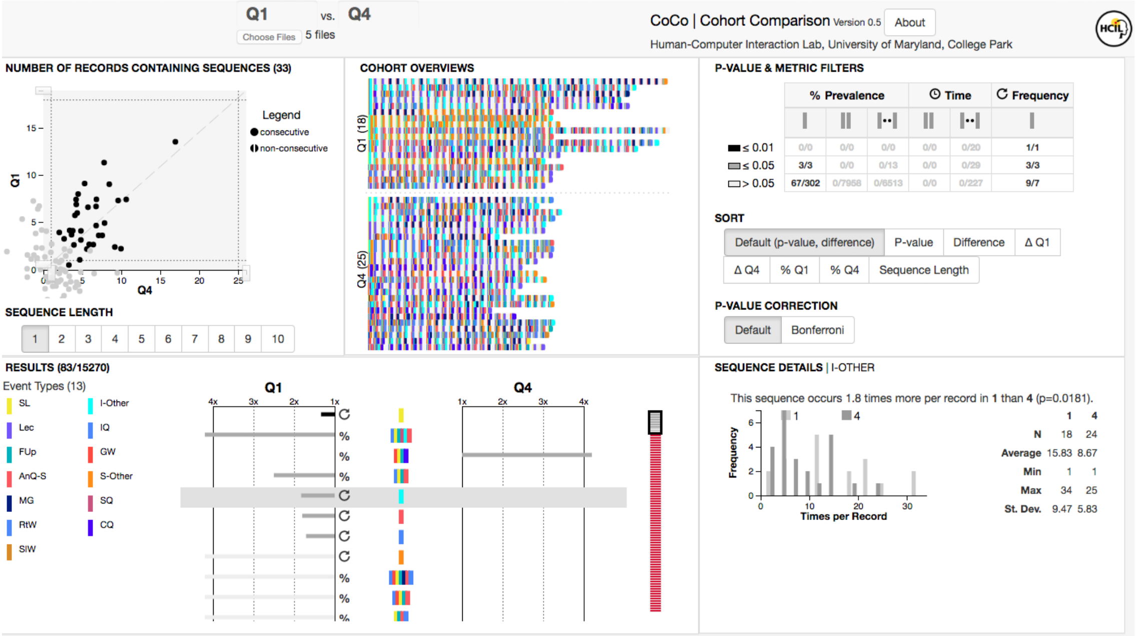
Since the datasets contained a lot of events (over 120), the analysts found the event filtering panel most helpful and they were able to focus the analysis on specific events. In addition, the reduced metric calculation time provided a much better user experience for data analysis, as the analysts did not need to wait for CoCo to load data and finish hypothesis testing before they could begin their explorations. Analysts all mentioned that they would like to explore the individual event sequences in the dataset more freely. They said that the results were a bit linear, and they would prefer to have freeform exploration and interactions.

6.6.2 Outcomes

For user. The work was useful for analysts to discover attributes about user behaviors. In the exit interview, analysts stated that the use of CoCo made finding these insights much easier than with the use of other tools and they would use CoCo again in the future.

For CoCo. Previous versions of CoCo had only been used on relatively small datasets of up to 2,000 records per cohort and up to 50 event categories. Web log datasets, on the other hand, record millions of users who access the website per day and hundreds of clickstream events per user. The increased volume of records and variety of event categories presented new challenges for CoCo on both the front- and back-ends. Through an iterative process over six weeks, we proposed solutions, implemented them into CoCo, and received feedback from analysts. The scalability techniques used were formalized into guidelines that were presented in a joint paper [73].

6.7 CS5: In-Classroom Student Behaviors



	% Prevalence			Time		Frequency
	▮	▮▮	▮··▮	▮▮	▮··▮	▮
■ ≤ 0.01	0/0	0/0	0/0	0/0	0/20	1/1
▨ ≤ 0.05	3/3	0/0	0/13	0/0	0/29	3/3
□ > 0.05	67/302	0/7958	0/6513	0/0	0/227	9/7

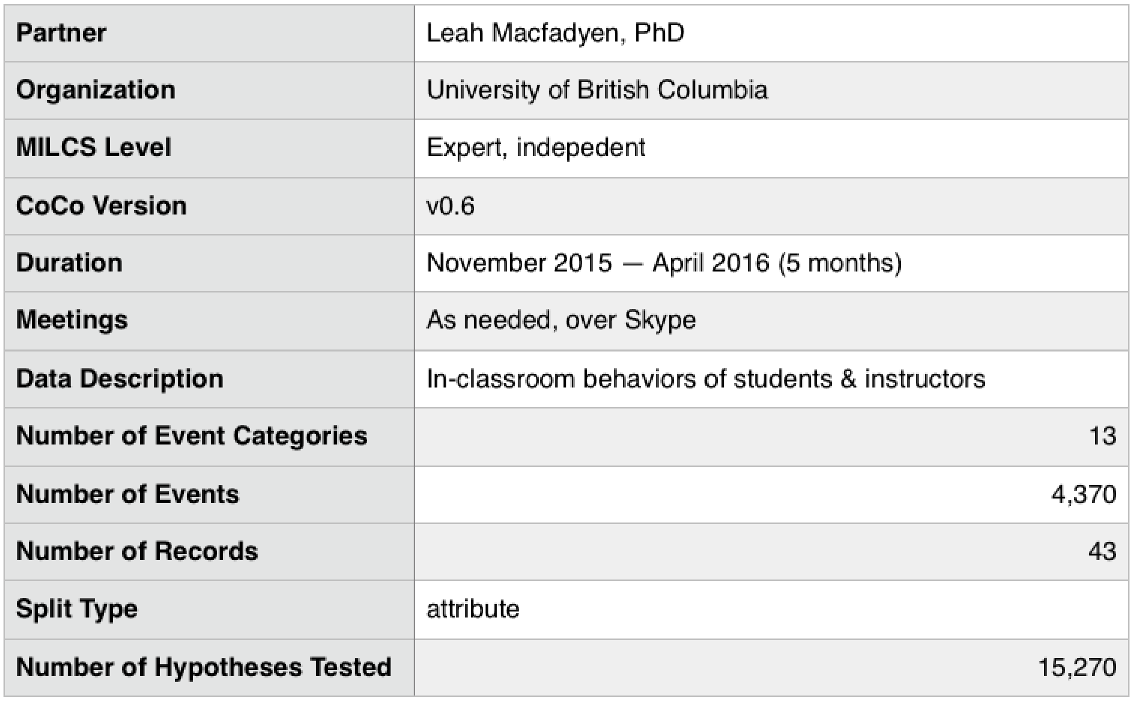
This sequence occurs 1.8 times more per record in 1 than 4 (p=0.0181).

	1	4
N	18	24
Average	15.83	8.67
Min	1	1
Max	34	25
St. Dev.	9.47	5.83

Figure 6.7: An analyst at the University of British Columbia (UBC) used CoCo to compare the in-classroom behaviors of students in the top quartile versus bottom quartile.

Participants. I worked with Dr. Macfadyen, Program Director of Evaluation and Learning Analytics in the Faculty of Arts at the University of British Columbia.

Procedure. All sessions were conducted remotely, through Skype and screensharing. These sessions were at irregular intervals that were scheduled as needed. The sessions would consist of troubleshooting, guidance on how to use CoCo, feedback from Dr. Macfadyen on her experience, as well as observation of how CoCo was used. In between these sessions, Dr. Macfadyen used CoCo independently and



Partner	Leah Macfadyen, PhD
Organization	University of British Columbia
MILCS Level	Expert, indepedent
CoCo Version	v0.6
Duration	November 2015 — April 2016 (5 months)
Meetings	As needed, over Skype
Data Description	In-classroom behaviors of students & instructors
Number of Event Categories	13
Number of Events	4,370
Number of Records	43
Split Type	attribute
Number of Hypotheses Tested	15,270

would regularly email her thoughts, experiences, results, questions, and requests related to CoCo.

Analysis Goals. In 2013, Smith et al. [74] outlined their development and use of a new tool called COPUS, which stands for the Classroom Observation Protocol for Undergraduate STEM. As part of a focus on improving student learning, they developed COPUS to facilitate the collection of information on the range and frequency of in-class teaching practices at department-wide and institution-wide scales. They and others have subsequently reported results generated through use of the tool, but almost exclusively present this data in pie chart form indicating student and instructor activity as percent of total time or activity intervals.

To date, analyses appear to have ignored the sequential element of the data.

Dr. Macfadyen wanted to explore and compare the actual sequencing of in-class activity in relation to student learning.

Specifically, her question in comparing two classes was to undercover correlating in-classroom behaviors with student performance across different classes.

Dataset. The COPUS data comprised manually collected observational data for student and instructor in-class activity in 14 different biology courses.

Each class has an outcome variable, "performance," which has been computed for each class as an average "% learning gain" based on pre- and post-tests. Performance is computed as normalized changes in test performance per class per student.

The class histories were then grouped based on their performance quartile and Dr. Macfadyen compared the top quartile against the bottom quartile.

The dataset contained 8 event types, grouped into "passive" and "active" actions for both students and instructors.

6.7.1 System Use

After intitially exploring the dataset in EventFlow, Dr. Macfadyen used CoCo to conduct two comparisons of top and bottom quartile students. The first comparison was of all courses, whereas the second was filtered by first year courses only.

6.7.2 Outcomes

For analyst. Overall, Dr. Macfadyen found CoCo to be useful, though early iterations of CoCo struggled to analyst the dataset. Dr. Macfadyen found that the frequency of use of clicker questions (CQ) and moments of independent student work (SIW) are significantly higher in top quartile-achieving courses. As a result of this exploration, Dr. Macfadyen presented her work with EventFlow and CoCo at a workshop on Learning Analytics and Knowledge (LAK) [75].

For CoCo. This case study deepened CoCo's ability to handle concurrent events. Because the COPUS data is bucketed into 2-minute timeslots, all events are concurrent. Significant changes to CoCo, as a result of this case study, include extending "sequences of length 1" to include concurrent events. That is, the occurrence of overlapping events is shown as a sequence of length 1 because they occur at a single timestamp. In doing so, analysts can more easily see which events commonly occur with other events and which do not.

This case study also served as an example of CoCo's usefulness in datasets with a low-volume of records but a high-volume of events. Though there were only 43 classroom histories, the volume of the events allowed for sufficient sample sizes and showed that CoCo is still suitable for relatively low-volume datasets.

6.8 CS6: Distinguishing Types of Radiation to the Bone

We worked with partners at the Department of Pharmaceutical Health Services Research at the University of Maryland School of Pharmacy in Baltimore. In previous work, the researchers were interested in developing an algorithm using claims data to differentiate between radiation delivered to the bone versus radiation delivered to the prostate gland, because billing codes available in claims data do not distinguish the site of radiation. Reliable measures for identifying the receipt of radiation to the bone are important in order to avoid bias in estimating the prevalence and/or mortality impact of skeletal-related events, including radiation to the bone.

Studies using healthcare claims employ various claims-based algorithms to identify radiation to the bone and mostly condition on prior claims with a bone metastasis diagnosis (billing) code [76–78]. They developed three classification algorithms that were compared using CoCo and EventFlow to investigate the timing of possible radiation to the bone among patients diagnosed with incident metastatic and nonmetastatic prostate cancer. One algorithm was based on prior literature while the other two were based on insights gained from data visualization software. Based on clinical input regarding the duration of palliative [79, 80] versus curative radiation, the researchers investigated the length of radiation episodes and found differences between cohorts in terms of the length of radiation. As expected, patients diagnosed with metastatic disease received shorter course radiation than patients diagnosed with nonmetastatic disease.

The feedback on CoCo was positive and the team valued the opportunity to visually compare cohorts of patients using summary statistics that pertained to the timing and frequency of events. The graphical results were shared with clinicians on the research team in order to determine whether the patterns were consistent with their expectations. The researchers felt the meaning of metrics could be explained more clearly; it was sometimes unclear what the x-axis represented and what statistical tests were used. They also suggested always showing the event labels, particularly for single-event metrics, to make understanding the icons a bit easier. The researchers expressed a need to be able to sort the rows of results with different factors, including by raw percentage of values in each cohort. We implemented this feature before the formal case study.

6.9 CS7: Children's AIM2

After the initial case study with Children's National Medical Center (Section 8), we began working on another dataset. Similar to how the ATLS protocol is standardized for the trauma bay, researchers were interested in seeing if similar patterns emerge during resuscitations dealing with head injuries which might guide in the development of guidelines or protocol.

The case study lasted about two months, while the analysts and I worked together to clean the data. Because this was a relatively new dataset however, there weren't enough records to form statistically significant conclusions ($n < 20$). However, CoCo was helpful in finding errors in the datasets and determining what remained to be cleaned. Although ultimately the case study didn't provide significant insights, it was helpful to understand CoCo's limitations in terms of number of records and number of event categories that can be supported and it helped the Children Hospital team understand how CoCo may be helpful in the future when the number of records increases.

6.10 CS8: Computer Activity Logs

Fan Du, a PhD student in Computer Science at the University of Maryland, used CoCo to identify patterns to detect insider threats using computer activity logs. The dataset contained approximately 180 million events from monitoring computer usage of employees, consisting of 6 event categories (e.g., login, email, web browsing, etc). The users were divided into "suspicious" versus "normal" users. After much data cleaning, the data was reduced to only several thousand events. The analyst was interested in identifying event categories which indicate suspicious user activity, in order to further simplify the datasets.

For each subset, CoCo identified event categories that occurred significantly more or less prevalently in high scored days than low scored days. Thus, analysts inspected a display that used only these differentiating event categories. For the above medium size subset, this strategy further reduced the number of events by 92% (from 462 to 24), and the number of unique complete sequences by 74% (from 27 to 7). Comparisons in temporal patterns between days with high and low scores were made based on the simplified visualization.

While differences were found we believed that the data itself was not complete or detailed enough to make inferences about might constitute suspicious event sequences. This case study resulted in the analyst using this dataset as an example for methods for cleaning and simplifying temporal event sequence data [81].

6.11 CS9: Social Media Messages

Cody Buntain, a PhD candidate in Computer Science at the University of Maryland, used CoCo to identify differences in structures for credible versus non-credible Twitter messages. The dataset used was CREDBANK, a large-scale corpus of social media messages collected between mid October 2014 and end of February 2015. It is a collection of streaming tweets tracked over this period, topics in this tweet stream, topics classified as events or non-events, and events annotated with credibility ratings [82]. Each record is an "event" that happened (e.g. the Boston marathon bomping) and events (in the context of CoCo) are individual tweets or messages.

Using the credibility ratings, the data was divided into credible (i.e., true) versus non-credible tweets, and CoCo was used to determine whether there are any structural differences between these two datasets to help identify features that may be used in developing automated credibility detection for Twitter messages.

CoCo revealed several differences in the structure of credible versus non-credible events [83]:

- First, credible events had a statistically significant higher frequency ($p < 0.01$)

of tweets than non-credible events.

- Breaking down the tweet type, credible events also exhibited a twice as many of retweets and tweets with media and four times as many web links ($p < 0.01$), while non-credible events had a higher frequency of hashtags and mentions of other users.
- Credible events showed a significantly higher proportion of media posts than non-credible events ($p < 0.01$).

6.12 CS10: Baseball Career Trajectories

Sean Barnes, an Assistant Professor of Operations Management in the Robert H. Smith School of Business at the University of Maryland, College Park. Dr. Barnes was interested in understanding how to determine characteristics that indicate promising players. Using a baseball-reference.com [84] dataset, which calculates a yearly Wins Above Replacement (WAR) average per player per year. The WARs are categorized into five groups, which show the player's demonstrated ability. The player's WAR is calculated once per year.

The initial analysis compared pitchers versus batters. Because of the very long histories of the players (in some instances, over 15 years), Dr. Barnes found the most useful metric to be the non-consecutive and consecutive subsequence results (e.g., long-term or multiyear patterns). The explicit way that CoCo breaks down each unique sequence was also helpful in quantifying the variety of player's career trajectories. Though variety is expected across all players, one key insight was that pitchers had more unique patterns than batters, possibly due to a higher potential for injuries.

6.13 8 Incomplete Case Studies

include figure/list again here?

There were eight other groups that expressed interest in using CoCo for analysis and received a demo of CoCo. Five in the healthcare domain, two in business, and one in transportation. However, these were not completed for a variety of reasons (Table 6.2c):

- Data quality deemed unsatisfactory - In three cases, the data required too much cleaning to continue with the case study. In the transportation case study, there were hundreds of event categories because they had been typed by operators instead of selected among a list of possible event names (e.g., they included the names of individuals being contacted instead of the job title). After the categories had been aggregated, the analyst realized that the procedure and timing of the recording of events events was different for different agencies so no valid comparisons could be made. The analyst effort was then redirected to attempting to change the way data is recorded.

- No suitable comparison - Three case studies were not completed because though data existed and was cleaned for event sequence analysis, there was no suitable comparison. In all three cases, the analysts had used EventFlow or CoCo for a previous trial and were successful in finding results, and were invited to try CoCo. However, we found that although cohort comparison seemed like the next logical step, there was no driving hypothesis which supported pre-defined subpopulations in the dataset to be compared. For example, in the case of the head trauma dataset, analysts were interested in understanding if there was a pattern that emerges consistently when treating head trauma patients and thus wanted to compare "similarly treated" patients versus "deviations." However, a central issue was that defining these subpopulations was part of the task and CoCo is not suited for clustering tasks. Although cohort comparison is relate to clustering and classification problems, CoCo is designed for exploratory analysis and open-ended questions, and there still must be a driving hypothesis that allows analysts to split the dataset into groups. Thus, CoCo is best suited for retrospective cohort analysis where the splitting method involves comparing outcomes, treatments (existence of an event), or record attributes.

- Aborted - In the remaining three cases, the case study partners became busy or unavailable after expressing interest and receiving a demo.

6.14 Summary

This chapter covers Contribution 4: Evaluations to demonstrate the utility and impact of these methods. Through a user study and a series of five long-term and five short-term case studies. The early, preliminary user study refined CoCo's design and allowed me to observe anaylysts' actual practice of analyzing real-world dataset using a combined CoCo and EventFlow tool. Following the procedure of a Multi-Dimensional, Long-term In-depth Case Study (MILCS) [71]. All case studies with CoCo illustrated the strengths of the system and highlighted limitations, which allowed me to iterate on its design. Though at each step, many improvements were necessary, each case study partner was able to understand their data and answer questions about cohort comparison better using CoCo than they had previously been able to.

Chapter 7: Discussion and Future Work

Event sequence data is being collected more and more, in a wide range of domains. With this increased volume of data, developing new, efficient methods for analyzing it is paramount. Despite the commonality of the data type, existing analysis tools for cohort comparison fail to address the unique challenges that come with comparing event sequences. My work aims to bridge this gap by providing an understanding of the complex task of event sequence comparison and provides a visual analytics tool that combines statistics with an interactive visualization to enable more rapid data exploration, hypothesis generation, and insight discovery.

The direct contributions of this dissertation are:

A taxonomy of metrics for comparing cohorts of temporal event sequences. Through a systematic literature review of EventFlow and other case studies, I identified common questions that users ask when comparing two or more groups of event sequences and organized these questions in a taxonomy of metrics.

A statistical framework for exploratory data analysis. I implemented a subset of the metrics introduced in the taxonomy and identify and solve the major practical challenges of applying thousands of statistical tests, a method I refer to as high-volume hypothesis testing (HVHT),

A family of visualizations and guidelines for interaction techniques. Through an iterative design process with case study partners, I develop and implement visualizations and interaction techniques that are useful for understanding and parsing large sets of hypothesis results.

Evaluations to demonstrate the utility and impact of these methods. I preform three types of evaluation through the development of CoCo:

- a preliminary user study comparing CoCo to EventFlow for the task of cohort comparison,
- six long-term case studies with case study partners: three in the medical domain, two in education, and one in web log analysis, and
- five short-term case studies: two in the medical domain and one each in sports analytics, social networks, and security.

7.1 Limitations

Though CoCo has shown to be a powerful analysis tool for analysts in a wide array of domains, there are some limitations to its application. Section 7.2 discusses avenues for future research,

7.1.1 Difference Metrics

CoCo focuses on differences between the cohorts, but metrics to show similarity between cohorts can also be useful. Additionally, CoCo focuses exclusively on events and sequences which *do* occur in a dataset.

7.1.2 Statistical False Positives

When running thousands of statistical tests on a single dataset, the chance of false positives and erroneous correlations increases. We attempt to mitigate these risks by providing options for statistical corrections, making the distribution of p-values more transparent, and allowing users to see hypothesis results in the context of other related sequences. Despite these considerations, however, advanced domain expertise of statistics is required to truly appreciate the associated pitfalls of this type of analysis. It is important to note that CoCo is intended for exploratory data analysis and that any significant results should be followed with a formal, controlled study to confirm or deny any hypotheses.

7.2 Future Work

7.2.1 Supporting Comparison of Three or More Groups

Extending CoCo to support comparison of three of more groups would require changes to the statistical methods and to its visualizations. On the statistics side, using a method such as one-way Analysis of Variance (ANOVA) [85] or linear regression would allow comparing three or more groups. Currently CoCo uses t-tests, which are adequate for but limited to comparing only two groups.

Extending CoCo to three or more groups would also require changes to its display. The current method of using left versus right columns for each of the cohorts works well for two groups, because each hypothesis result can be ranked and listed. However, extending to three or more groups would require losing the ability to rank and list the results, or require new displays entirely. One option for the display would be to use a lower triangle matrix to show multiple pairwise comparisons for each group pair, similar to the Simplified Overviews method [86].

7.2.2 Integrated Cohort Selection

The first step in every case study partner's analysis was to determine the two cohorts that were being compared. Providing methods for cohort selection integrated directly into CoCo would not only be more convenient, but would allow for more complex analysis. Allowing users to switch between split features will also enable them to determine causal relationships. For example, in the Children's Hospital case study, the analysts first looked at patients who were treated correctly versus those who were not. They found that the "now" attribute was a discriminative feature. To confirm this difference, they then split by "now" versus "not now" patients, and found that there was indeed a significant difference in protocol adherence.

Analysis can be further aided by integrating tools for cohort selection within CoCo by provided simple interaction techniques to select the split feature. This problem is interesting in the number of ways a cohort can be selected depending on the analysis to be performed:

- Record attribute
    - Binary values - each value corresponds directly to a cohort
    - Categorical values - the user must select which values go into which cohort (e.g., if attribute is what browser a user was using, we can divide into mobile vs. desktop)
    - Continuous values - choosing binary ranges or ranges that may not be contiguous (e.g., normal blood sugar level between 70 and 99, abnormal otherwise)
- Absolute date (e.g., this year vs last year, beginning by..., ending by..., occurring during...)
- Relative time (e.g., lasting longer or shorter than...)
- Outcome
- Event/sequence ("contains")

Further, by integrating the split type into the tool itself, the algorithm can make use of this information for optimizing and reducing the subspace for metrics to calculate, explained more in the Section 7.2.3.

Record Attribute Visualization. The next step, after choosing cohorts, is to visualize the record attributes. There are two primary use cases for how cohort visualizing cohort attributes might be useful: (1) In the case of where cohorts are already selected (for example, in A/B testing of web sites, where the user is placed in a group at random or not based on any user-attributes), we might use a visualization to easily determine whether the groups are balanced in all possible attributes. The interest in this is not necessarily for any actionable outcome, just to be aware of any biases or imbalances between the cohorts. (2) In the case of a retrospective study where cohorts must carefully be chosen for analysis, the visualization can help a user select the patients for each cohort by providing a real-time, responsive visualization that shows the attribute balances as the user moves records between the two groups. Visualizing these attributes is becomes more complex when we take the combination of attributes.

Secondly, the same problem of visualization the different types of values an attribute can have occurs. For example, a simple pie chart might be sufficient for binary values, whereas continuous or categorical values might require some sense of the minimum, maximum, and average values, as well as an overall distribution.

Automatically Generating Balanced Cohorts Based on Record Attributes. A natural extension might be automatically suggesting which records to move in order to balance the cohorts best (best could mean fewest number of moves, most balanced number of total records, etc).

7.2.3 Optimization

Currently, CoCo will run a metric on the datasets when the user selects the metric. Because the set of metrics is bounded, users can benefit from automated computation of every metric in advance. Automatic computation will provide users guidance in exploring the problem-space and save time during exploratory data analysis.

When comparing patterns across two or more cohorts, statistical tests are important for comparing means and proportions. However, calculating the significance (e.g., p-value) for a statistical test is a computationally time consuming approximation problem. Because the number of unique subsequences in a cohort grows exponentially with the number of event types and records, even small datasets with 250 records, 10 event types, and under 5,000 unique subsequences can result in significant wait times for the user. Preliminary timing tests using CoCo indicate such a dataset would take as long as 1.5 seconds to calculate the significance tests for prevalence of all 5,000 subsequences in both cohorts.

One visual analytics approach to reducing wait times between user operations is given in Progressive Visual Analytics. Stolper et al. [7] give design guidelines which include allowing the user to direct the algorithm via prioritizing subspaces and designing an algorithm to give meaningful, partial feedback. However, with statistical data, there are unique methods for the algorithm to prioritize subspaces automatically. I propose that progressive visual analytics can be improved by self-directed algorithms.

Implementing the design guidelines of Stolper et al. provides users feedback during the computation process and allows them prioritize or ignore subspaces of interest. Besides user-directed methods, progressive analytics algorithms can "self-direct" in order to maximize efficiency in computing many metrics.

Prioritizing by P-Value Estimation. Because calculating the exact p-value is time consuming, we can significantly reduce calculation time by prioritizing sequences that are likely to be significant. For example, the χ^2 statistic can be calculated in constant time and from this, we can determine the range that the p-value will be in using a look-up table of pre-computed values. If the p-value is above 0.1, it is likely that the user will not care about the exact p-value, and we can skip this subspace. Similarly, for p-values in the range below 0.1, the algorithm should prioritize these results and determine the exact p-value before moving on to potentially less significant results.

Ignoring Subspaces by Split Feature. All metrics may not be applicable on all datasets or sequences. For example, the factor on which the cohorts are formed may call for different types of questions to be asked about the data. Consider a set of medical patient records split by date (e.g., last month's trials versus this month's). A researcher might see how outcomes for the patients differ between the cohorts, whereas in a dataset split by the outcome of the record (e.g., patients who die versus those who live) would ignore such a metric.

CoCo enables users to split cohorts by factors such as:

- time (patients this month versus last month)
- outcome
- patient attribute (age, gender, location, team, position)
- event occurrence (treatment a versus treatment b)

If the algorithm knows what the cohorts are split by, we can eliminate some metrics completely. Users can specify the split factor or the algorithm can automatically detect the split factor. For example, if the cohorts are split by a binary patient attribute, we can expect to see 100% of patients on cohort α to have one value, and 100% of β to have the other.

7.2.4 Database Backend

CoCo currently, stores all data in memory. As a result, the size of datasets and results is limited by the size of the user's machine memory and by browser data transfer limits. Using a database would allow for more scalable data storage and more seamless integration with users' existing tools and databases.

Tables. Such a dataset would require tables in three major areas: (1) raw data storage for each cohort, (2) intermediary sequence counts and information, and (3) hypothesis results table.

Cohort Data. The current file input format lends itself nicely to a relational database. The raw cohort data can be stored in a single Events table, with headings that match the current input file schema:

Event ID	Cohort	Record ID	Event	Time

The Event ID would be the key that is automatically assigned. If interval events were introduced, a fourth column could be added for "end time" with an imposed constraint that it must occur after the "start time."

Event attributes could be stored in a third table with information of the Event ID (corresponding to an entry in the Events table), in another three column table:

Event ID	Attribute	Value

A Record Attributes table would similarly include a Record ID, Attribute, and Value:

Record ID	Attribute	Value

Intermediate Sequence Value Tables. In calculating the metrics for the event sequences in CoCo, the intermediate results should be stored in tables because they are accessed by many parts of the system. First, a Sequence table should store all the sequences found in the dataset and the number of times it occurs in each cohort.

Sequence ID	Sequence	Consecutive	Occurrences in A	Occurrences in B

The Sequence ID would be automatically assigned by the database. Sequence would be the sequence of events and consecutive would be a True or False boolean value. Tables for other metrics would include number of records containing a sequence and duration of sequence (from first to last event).

Results. Hypothesis results would be stored in a table as they are calculated, based on the count in the intermediary tables.

Sequence ID	Metric	Value A	Value B	P-Value

Queries. Queries could then be made from the interface based on users' selected sorting and filtering mechanisms. Most importantly, results should be filterable by p-value, sample size, and values. Sequences should be selectable by sequence length.

7.2.5 Interval Events

Extending to interval data requires considerations on both the backend and the frontend. Monroe [63] covers these challenges in great detail, from input processing to data structures and storage to display methods, and many of these issues and their solutions can be applied to CoCo. For example, the inclusion of interval events would require additional information about the granularity of timestamps and consistency checks during the file processing.

Regarding visualizing hypotheses and sequences, adaptations would need to be made to account for the 13 temporal relationships between two intervals and 5 relationships between intervals and points [63]. A method similar to EventFlow could be employed, where the start and end events are represented as points and the interval between the events are shown as a shaded region.

Aside from practical issues for storing and representing intervals, the task of cohort comparison specifically would require the addition of new metrics. While point events have the notion of "before" and "after," interval events introduce the concept of “during.” More work would need to be done extend the taxonomy of metrics for new hypotheses involving intervals, but potential metrics could include:

- Duration of an interval events. Does one interval tend to last longer in one cohort than the other?
- Duration of interval events. Aside from how often certain intervals overlap, how long do they overlap for?
- Prevalence of events and sequences that occur during an interval.

7.2.6 Extending to Other Data Types

Though my dissertation focuses on comparing cohorts of event sequences, this work can be extended to use with other data types, such as network graphs, time series, or multivariate data. In each case, the metrics, visualizations, and interactions would have to be adjusted for the appropriate data type.

Take for example extending to network graph data. Metrics can be similarly divided into “summary metrics” which summarize the networks as a whole (e.g., node and edge counts, degree of connectedness, and reciprocity). Additionally, there are metrics dealing with the node-level, such as degree (including in- and out-degrees for directed networks), node centrality, and node closeness. The metrics could also include metrics about specific subgraphs within the graph – for example, the distance between pairs of nodes, the number of unique paths between the nodes, and nodes that are on the path between two other nodes.

Metrics would be applied similarly, by mining for all nodes and subgraphs and applying the metrics to each in order to find significant difference in the structure and values of the metrics, and the results could be distilled similarly into two values (one for each cohort) and a p-value.

Regarding visualization, the same guidelines would stand: it would be necessary to have an overview of both cohorts and a dedicated results view to review statistical test results in detail, which could remain very similar to the existing work. However, new representations for displaying the networks and hypotheses would be necessary. For example, because each node is unique, it would not be possible to encode the nodes with color. Color could potentially be used to represent node attributes or additional metric values.

7.2.7 Journaling

Some users expressed needing a method to keep track of progress so far when exploring a result set. For example, it is important to be able to mark certain results as "reviewed" versus not and to mark whether reviewed results should be kept. Users also requested a way to annotate results that they found interesting, to indicate possible factors for the result. Lastly, users requested a way to export annotated and results marked as important.

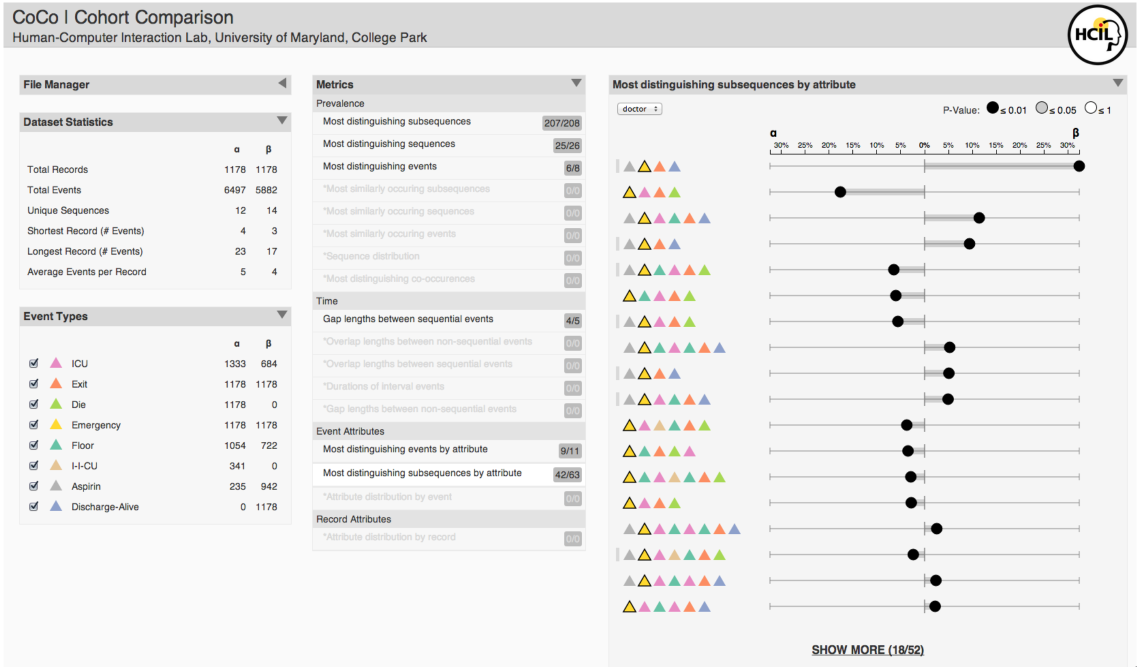
7.3 Conclusion

This dissertation aims to bridge the gap between the unique needs of event sequence cohort comparison and the limitations of existing tools by providing an understanding of the complex task of event sequence comparison and providing a visual analytics tool that combines statistics with an interactive visualization to enable more rapid data exploration, hypothesis generation, and insight discovery. Through implementation of the scalability, design, and interaction principles into a visual analytics tool, CoCo, I present a ready-to-use tool to support this type of comparison. Work with real-world analysts using CoCo shows the utility of this tool. This chapter summarizes the work of my dissertation and discusses the opportunities for future work that my dissertation opens.

Appendix 8: Evolution of CoCo

This appendix includes detailed descriptions of the previous versions of CoCo and highlights the changes made during its evolution.

8.1 Version 1



CoCo I Cohort Comparison

Human-Computer Interaction Lab, University of Maryland, College Park

File Manager

Dataset Statistics

	α	β
Total Records	1178	1178
Total Events	6497	5882
Unique Sequences	12	14
Shortest Record (# Events)	4	3
Longest Record (# Events)	23	17
Average Events per Record	5	4

Event Types

			α	β
☑	▲	ICU	1333	684
☑	▲	Exit	1178	1178
☑	▲	Die	1178	0
☑	▲	Emergency	1178	1178
☑	▲	Floor	1054	722
☑	▲	I-I-CU	341	0
☑	▲	Aspirin	235	942
☑	▲	Discharge-Alive	0	1178

Metrics

Prevalence	
Most distinguishing subsequences	207/208
Most distinguishing sequences	25/26
Most distinguishing events	6/8
*Most similarly occuring subsequences	0/0
*Most similarly occuring sequences	0/0
*Most similarly occuring events	0/0
*Sequence distribution	0/0
*Most distinguishing co-occurences	0/0
Time	
Gap lengths between sequential events	4/5
*Overlap lengths between non-sequential events	0/0
*Overlap lengths between sequential events	0/0
*Durations of interval events	0/0
*Gap lengths between non-sequential events	0/0
Event Attributes	
Most distinguishing events by attribute	9/11
Most distinguishing subsequences by attribute	42/63
*Attribute distribution by event	0/0
Record Attributes	
*Attribute distribution by record	0/0

Most distinguishing subsequences by attribute

SHOW MORE (18/52)

Figure 8.1: The first version of CoCo was largely textual, with results grouped by metric type. Analysis could select the results they wished to view using the metric list in the middle panel.

The initial version of CoCo (Figure 8.1) focused on how to implement statistics and organize and display the result set. The initial version was largely textural, with five panels. Summary statistics were shown on their own, with side-by-side values for each cohort α and β. The event legend displayed counts for each event and allowed used to filter by a checkbox.

The middle panel was the main form of navigation for results: a four-tiered list displayed all possible metrics and the number of hypotheses. The list was static and included all metrics, even if they were not implemented yet. Users were able to click on a metric in a list, to view the corresponding results.

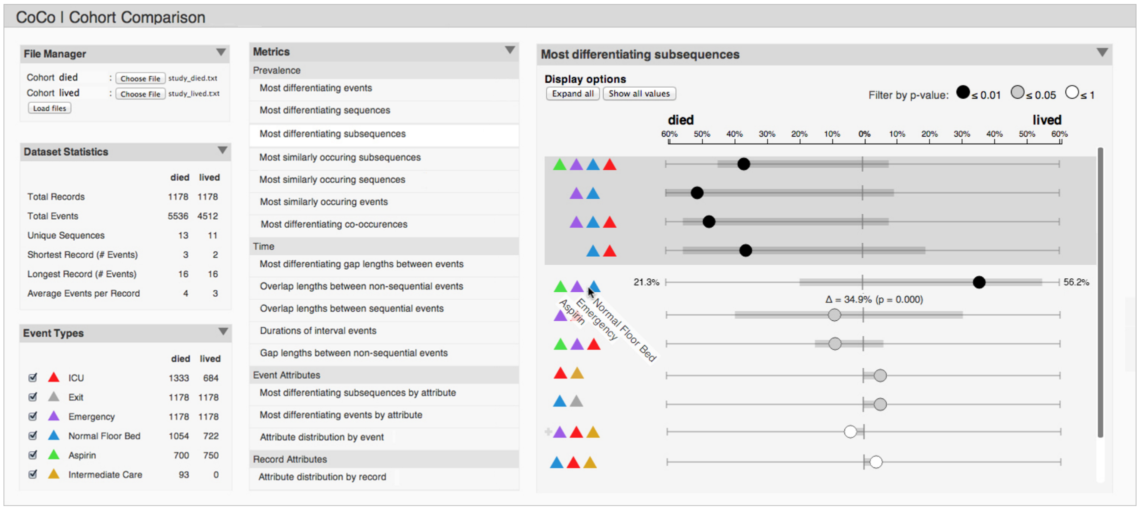
There were no methods of filtering, though results that were subsets of another were grouped together and expanded if a user clicked it. These aggregated results were denoted by a shaded bar to the left of the sequence.

The result display was based on the type of metric and the axes changed depending on which metric was selected. For prevalence metric, the axes were scaled to the largest percentage and grey bars grew from the middle to either side to indicate the value in each cohort. A circle was placed to indicate the difference in value for each cohort, placed on the side where the value was greater.

8.2 Version 2

The changes in the second version of CoCo were primarily usability based:

Organized metric list to suggest anaysis order. The metrics list was reorganized based on our observations in the Children's ATLS case study (Section 8).


CoCo | Cohort Comparison
File Manager
Choose File
Load files
Metrics
Prevalence
Most differentiating events
Most differentiating sequences
Most differentiating subsequences
Most similarly occuring subsequences
Most similarly occuring sequences
Most similarly occuring events
Most differentiating co-occurences
Time
Most differentiating gap lengths between events
Overlap lengths between non-sequential events
Overlap lengths between sequential events
Durations of interval events
Gap lengths between non-sequential events
Event Attributes
Most differentiating subsequences by attribute
Most differentiating events by attribute
Attribute distribution by event
Record Attributes
Attribute distribution by record
Most differentiating subsequences
Display options
Expand all
Show all values
Filter by p-value: ● ≤ 0.01 ○ ≤ 0.05 ○ ≤ 1
died
lived
60% 50% 40% 30% 20% 10% 0% 10% 20% 30% 40% 50% 60%
21.3%
56.2%
Δ = 34.9% (p = 0.000)
Aspirin
Emergency
Normal Floor Bed

Cohort died : study_died.txt

Cohort lived : study_lived.txt

Dataset Statistics

	died	lived
Total Records	1178	1178
Total Events	5536	4512
Unique Sequences	13	11
Shortest Record (# Events)	3	2
Longest Record (# Events)	16	16
Average Events per Record	4	3

Event Types

			died	lived
☑	▲	ICU	1333	684
☑	▲	Exit	1178	1178
☑	▲	Emergency	1178	1178
☑	▲	Normal Floor Bed	1054	722
☑	▲	Aspirin	700	750
☑	▲	Intermediate Care	93	0

Figure 8.2: CoCo version two brought a variety of usability fixes.

Within each category, analysts needed to first look at prevalence of single events, whole sequences, then subsequences.

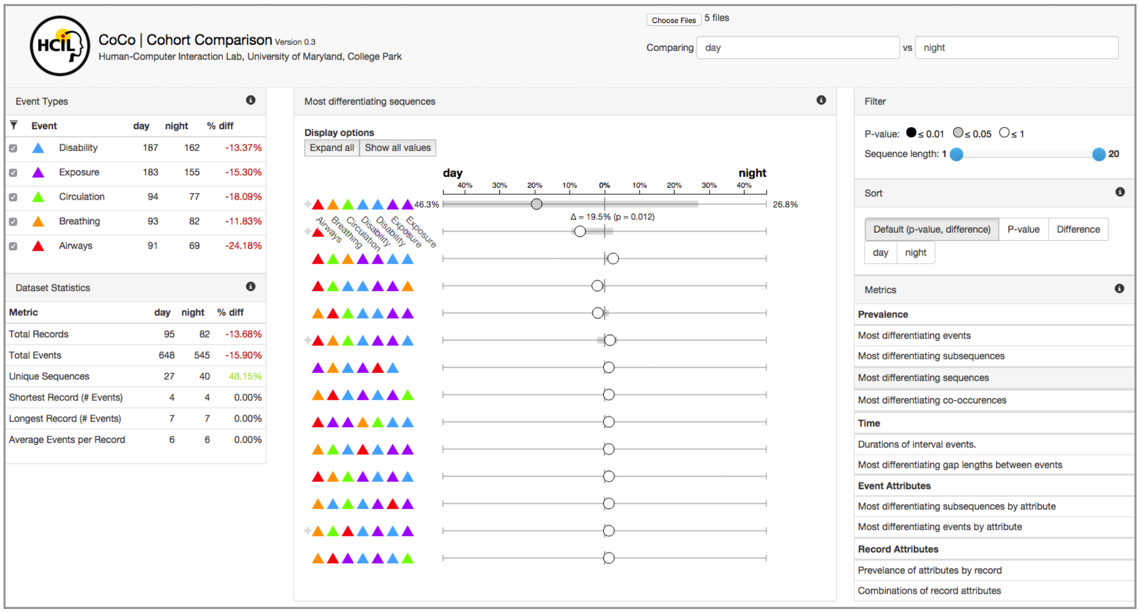
Custom cohort names. Analysts are able to rename the cohorts.

Hover tooltips for contextual information. It was noted that it was hard to remember which colors corresponded to which events, so hover tooltip information was introduced to ease this. Exact values for the hypothesis result are also shown. Additional display options allowed all tooltips to be shown or hidden (regardless of hovering).

Filtering by p-value. Filtering by p-value was introduced.

8.3 Version 3

Version 3 added more utility to parsing the result set.



CoCo | Cohort Comparison Version 0.3
Human-Computer Interaction Lab, University of Maryland, College Park

Choose Files 5 files
Comparing day vs night

Event Types

	Event	day	night	% diff
☑	Disability	187	162	-13.37%
☑	Exposure	183	155	-15.30%
☑	Circulation	94	77	-18.09%
☑	Breathing	93	82	-11.83%
☑	Airways	91	69	-24.18%

Dataset Statistics

Metric	day	night	% diff
Total Records	95	82	-13.68%
Total Events	648	545	-15.90%
Unique Sequences	27	40	48.15%
Shortest Record (# Events)	4	4	0.00%
Longest Record (# Events)	7	7	0.00%
Average Events per Record	6	6	0.00%

Most differentiating sequences

Display options
Expand all | Show all values

Filter
P-value: ● ≤ 0.01 ○ ≤ 0.05 ○ ≤ 1
Sequence length: 1 — 20

Sort
Default (p-value, difference) | P-value | Difference
day | night

Metrics
Prevalence
Most differentiating events
Most differentiating subsequences
Most differentiating sequences
Most differentiating co-occurences
Time
Durations of interval events.
Most differentiating gap lengths between events
Event Attributes
Most differentiating subsequences by attribute
Most differentiating events by attribute
Record Attributes
Prevelance of attributes by record
Combinations of record attributes

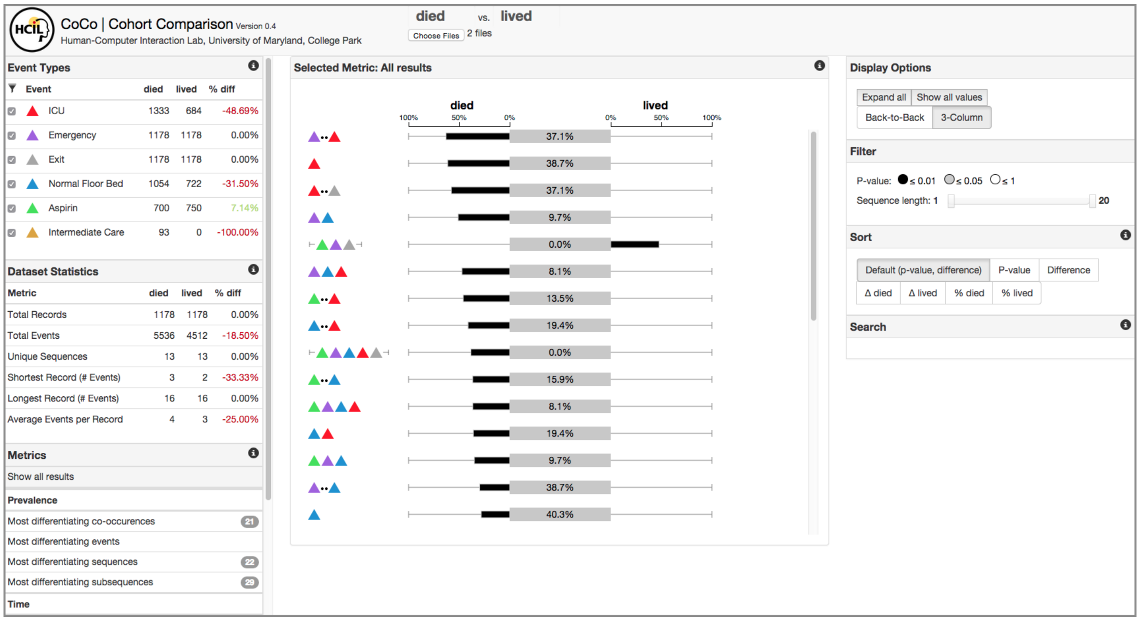
Figure 8.3: Version 3 added more utility to parsing the result set through methods for filtering and sorting, layout changes, and explicit difference encodings.

Methods for filtering and sorting. Users were able to sort by sequence length, in addition to p-value. Methods for sorting the result display were introduced. Previous versions defaulted to a method based on p-value and difference size, which remained the default sorting method. However, more controls were provided so users could sort by p-value only, difference size only, or the raw value of the result of either cohort A or B.

Usability changes to layout. The layout was rearranged and lightened, to allow provide more detail to the result view.

Explicit differences in overview statistics. A third column was added to the summary statistics and event legend to show the percent difference between the two cohorts, colored by which cohort was bigger (green = Cohort A, red = Cohort B).

8.4 Version 4



CoCo | Cohort Comparison Version 0.4
Human-Computer Interaction Lab, University of Maryland, College Park

died vs. lived

Event Types

Event	died	lived	% diff
☑ ICU	1333	684	-48.69%
☑ Emergency	1178	1178	0.00%
☑ Exit	1178	1178	0.00%
☑ Normal Floor Bed	1054	722	-31.50%
☑ Aspirin	700	750	7.14%
☑ Intermediate Care	93	0	-100.00%

Dataset Statistics

Metric	died	lived	% diff
Total Records	1178	1178	0.00%
Total Events	5536	4512	-18.50%
Unique Sequences	13	13	0.00%
Shortest Record (# Events)	3	2	-33.33%
Longest Record (# Events)	16	16	0.00%
Average Events per Record	4	3	-25.00%

Selected Metric: All results

P-value: ● ≤ 0.01 ○ ≤ 0.05 ○ ≤ 1
Sequence length: 1 20

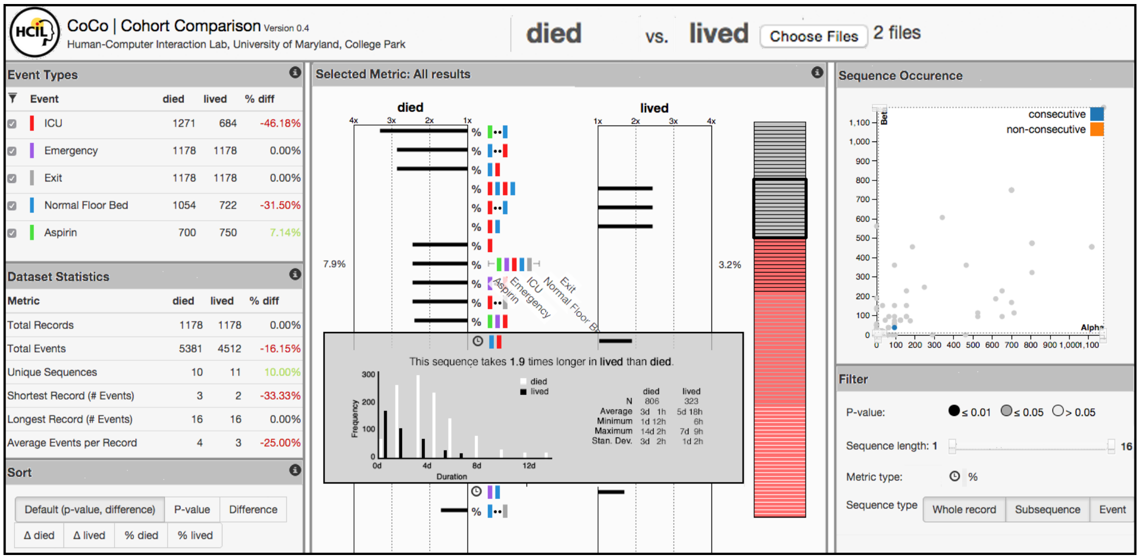
Figure 8.4: CoCo v4 introduced important changes in the way sequences and hypothesis results were displayed.

In previous versions of CoCo, analysts comments on the similarity of CoCo's result display to statistical error visualizations. Because of the case study partners' familiarity with statistical error bars, I explored alternatives for displaying hypothesis results.

The fourth version of CoCo displayed the common percentage along the middle of the result row. Then, a bar (colored by p-value group) grew to the left or right to indicate the size and significance of the difference. In this version, analysts are more clearly able to see where the major differences in their datasets are.

This version also introduced iconography for differentiating the different types of sequences because users expressed wanting to see all results of a metric, regardless of sequence type. Thus, icons were developed to indicate which sequences were consecutive versus non-consecutive and whole records versus partial.

8.5 Version 5



CoCo | Cohort Comparison Version 0.4
Human-Computer Interaction Lab, University of Maryland, College Park

died vs. lived [Choose Files] 2 files

Event Types

Event	died	lived	% diff
ICU	1271	684	-46.18%
Emergency	1178	1178	0.00%
Exit	1178	1178	0.00%
Normal Floor Bed	1054	722	-31.50%
Aspirin	700	750	7.14%

Dataset Statistics

Metric	died	lived	% diff
Total Records	1178	1178	0.00%
Total Events	5381	4512	-16.15%
Unique Sequences	10	11	10.00%
Shortest Record (# Events)	3	2	-33.33%
Longest Record (# Events)	16	16	0.00%
Average Events per Record	4	3	-25.00%

Sort

Default (p-value, difference) | P-value | Difference
Δ died | Δ lived | % died | % lived

Selected Metric: All results

This sequence takes **1.9** times longer in **lived** than **died**.

	died	lived
N	806	323
Average	3d 1h	5d 18h
Minimum	1d 12h	6h
Maximum	14d 2h	7d 9h
Stan. Dev.	3d 2h	1d 2h

Sequence Occurence

Filter

P-value: ● ≤ 0.01 ◉ ≤ 0.05 ○ > 0.05
Sequence length: 1 — 16
Metric type: ⏲ %
Sequence type: Whole record | Subsequence | Event

Figure 8.5: The fifth version of Coco introduced the most major changes: removing the metrics list, redesigning hypothesis results, sequence scatterplot, and details on demand.

The fifth version of Coco introduced the most major changes.

Removal of metrics list. Through case studies with analysts, it became clear that analysts did not necessarily care which metrics a result was from. That is, instead of checking each metric result group individually, they wished to answer the question ”what are the top 10 differences,” regardless of metric type. In lieu of the metrics list, filters were added so users could still filter by metric or sequence type.

Redesign of result hypothesis visualization. The most major change as a result of listing all hypothesis results was the challenge of displaying results that require different units (percents versus frequency versus elapsed times). As a result, the axes were changed to ratios and the absolute values of the results were removed from the visual encoding. The center (where the absolute values were displayed) were replaced with a visual representation of the hypothesis.

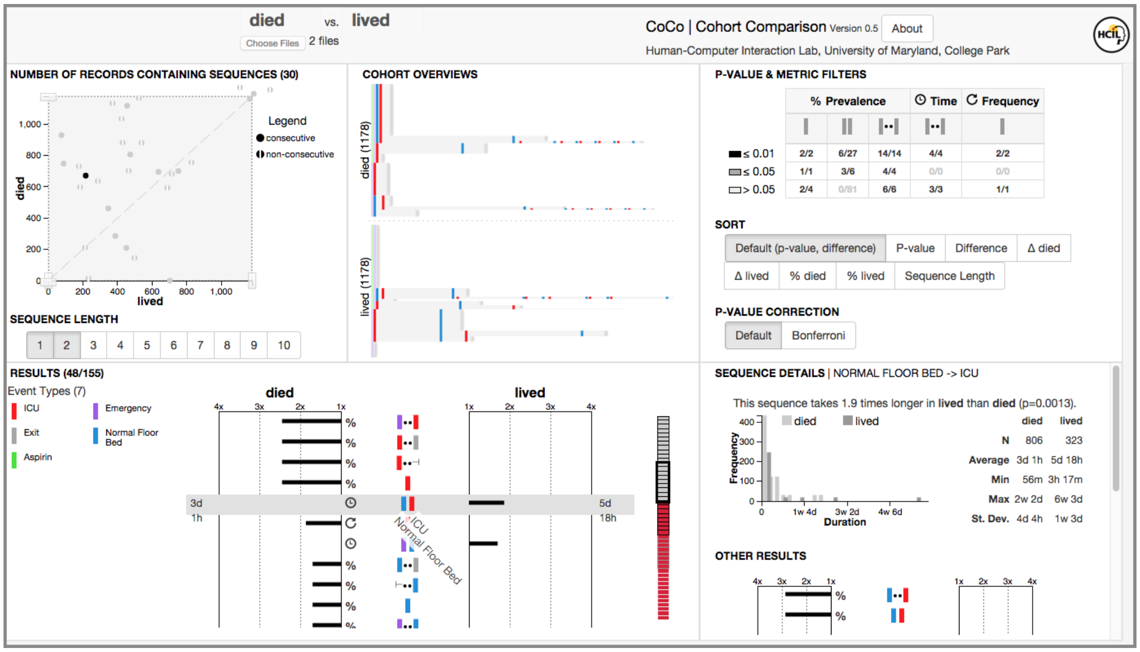
Sequence scatterplot was added. A sequence scatterplot was added to provide an overview of how individual sequences occur throughout the dataset.

Details on demand. Clicking a result provided details on demand, which include a histogram for results with a distribution of values and statistics about sample size, minimum, maximum, and standard deviation.

Event icons converted to rectangles. The event icons were converted to rectangles in order to save space.

8.6 Version 6

The final version of CoCo streamlined the process model observed through the case studies – the layout was rearranged to provide an overview first and details about specific results last. A stacked EventFlow chart was added to provide high-



	% Prevalence			◷ Time	↻ Frequency
■ ≤ 0.01	2/2	6/27	14/14	4/4	2/2
▨ ≤ 0.05	1/1	3/6	4/4	0/0	0/0
□ > 0.05	2/4	0/81	6/6	3/3	1/1

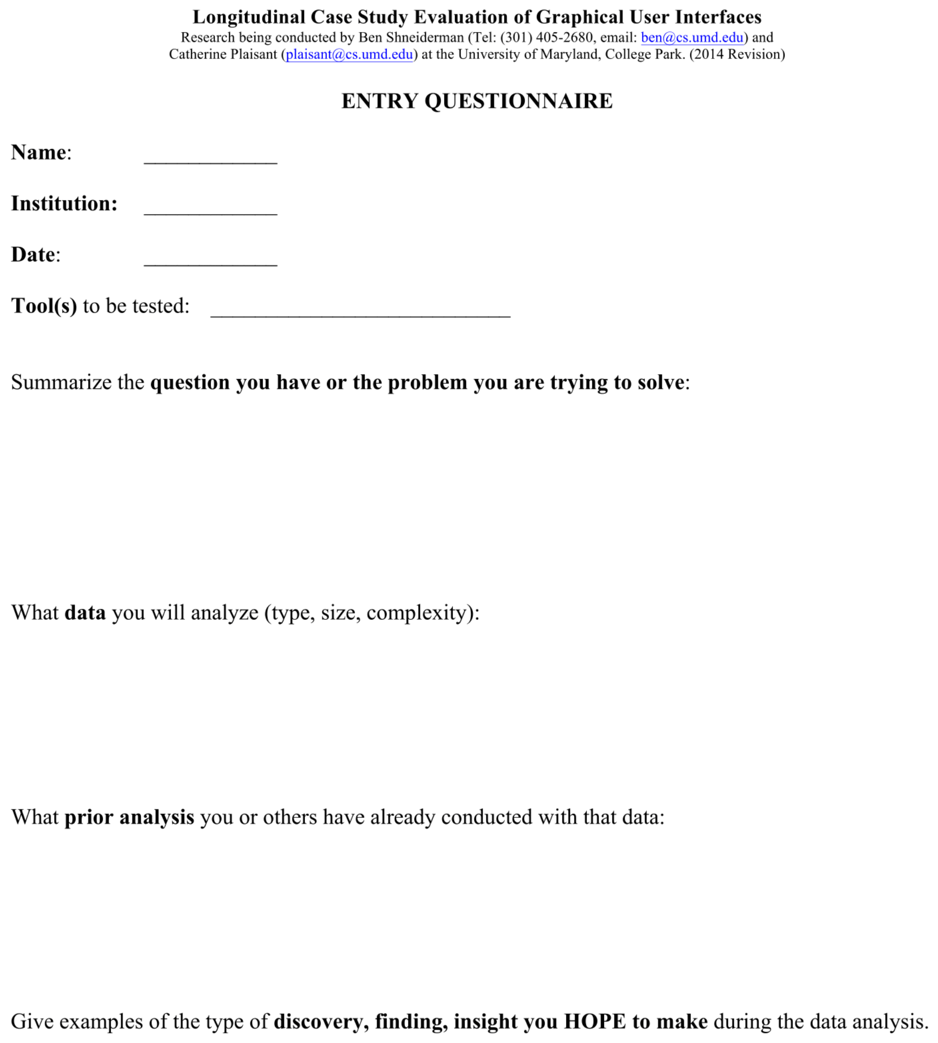
This sequence takes 1.9 times longer in **lived** than **died** (p=0.0013).

	died	lived
N	806	323
Average	3d 1h	5d 18h
Min	56m	3h 17m
Max	2w 2d	6w 3d
St. Dev.	4d 4h	1w 3d

Figure 8.6: The final version of CoCo (v6) streamlined the process model observed through the case studies.

level overviews of each dataset. Summary and event statistics were removed to focus the analysis on event metric results, since those statistics were only looked at once at the beginning of the analysis and could easily be duplicated by other tools.

Appendix 9: Case Study Questionnaires



Longitudinal Case Study Evaluation of Graphical User Interfaces

Research being conducted by Ben Shneiderman (Tel: (301) 405-2680, email: ben@cs.umd.edu) and
Catherine Plaisant (plaisant@cs.umd.edu) at the University of Maryland, College Park. (2014 Revision)

ENTRY QUESTIONNAIRE

Name: ____________

Institution: ____________

Date: ____________

Tool(s) to be tested: ____________________________

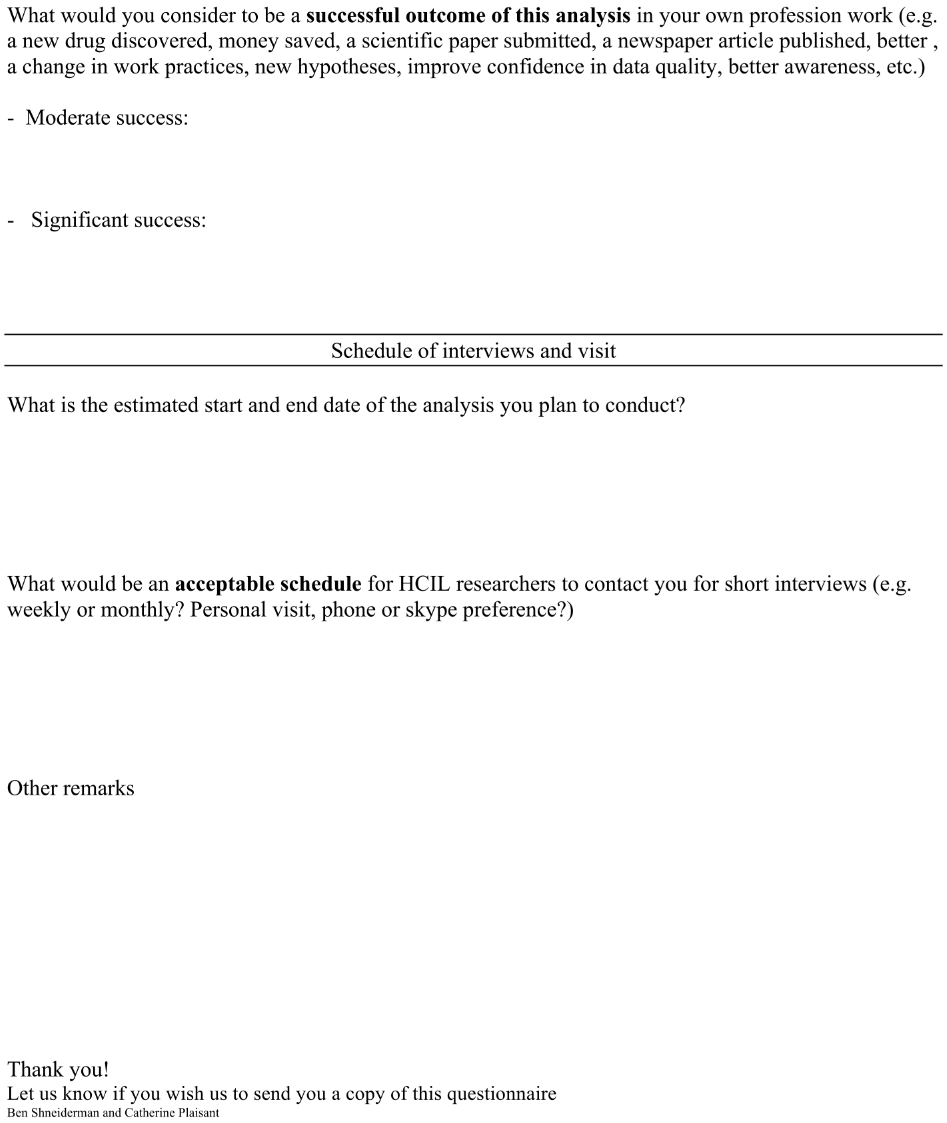
Summarize the **question you have or the problem you are trying to solve**:

What **data** you will analyze (type, size, complexity):

What **prior analysis** you or others have already conducted with that data:

Give examples of the type of **discovery, finding, insight you HOPE to make** during the data analysis.

Figure 9.1: Entry questionnaire, page 1.



What would you consider to be a **successful outcome of this analysis** in your own profession work (e.g. a new drug discovered, money saved, a scientific paper submitted, a newspaper article published, better , a change in work practices, new hypotheses, improve confidence in data quality, better awareness, etc.)

- Moderate success:

- Significant success:

Schedule of interviews and visit

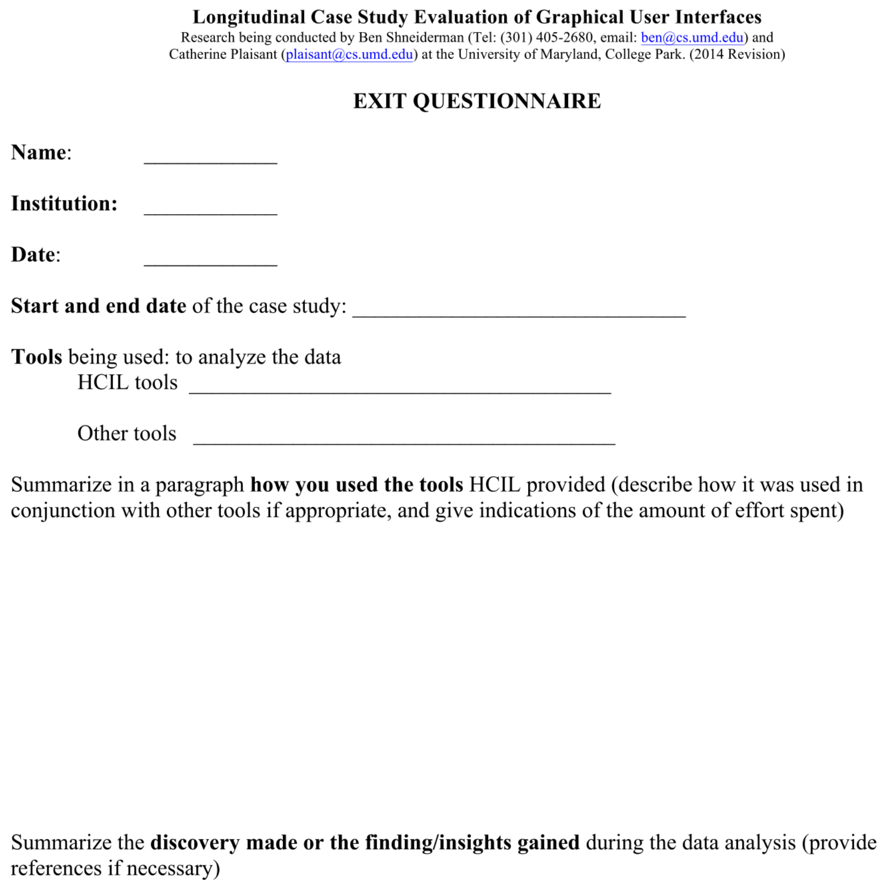
What is the estimated start and end date of the analysis you plan to conduct?

What would be an **acceptable schedule** for HCIL researchers to contact you for short interviews (e.g. weekly or monthly? Personal visit, phone or skype preference?)

Other remarks

Thank you!
Let us know if you wish us to send you a copy of this questionnaire
Ben Shneiderman and Catherine Plaisant

Figure 9.2: Entry questionnaire, page 2.



Longitudinal Case Study Evaluation of Graphical User Interfaces
Research being conducted by Ben Shneiderman (Tel: (301) 405-2680, email: ben@cs.umd.edu) and
Catherine Plaisant (plaisant@cs.umd.edu) at the University of Maryland, College Park. (2014 Revision)

EXIT QUESTIONNAIRE

Name: ____________

Institution: ____________

Date: ____________

Start and end date of the case study: _______________________________

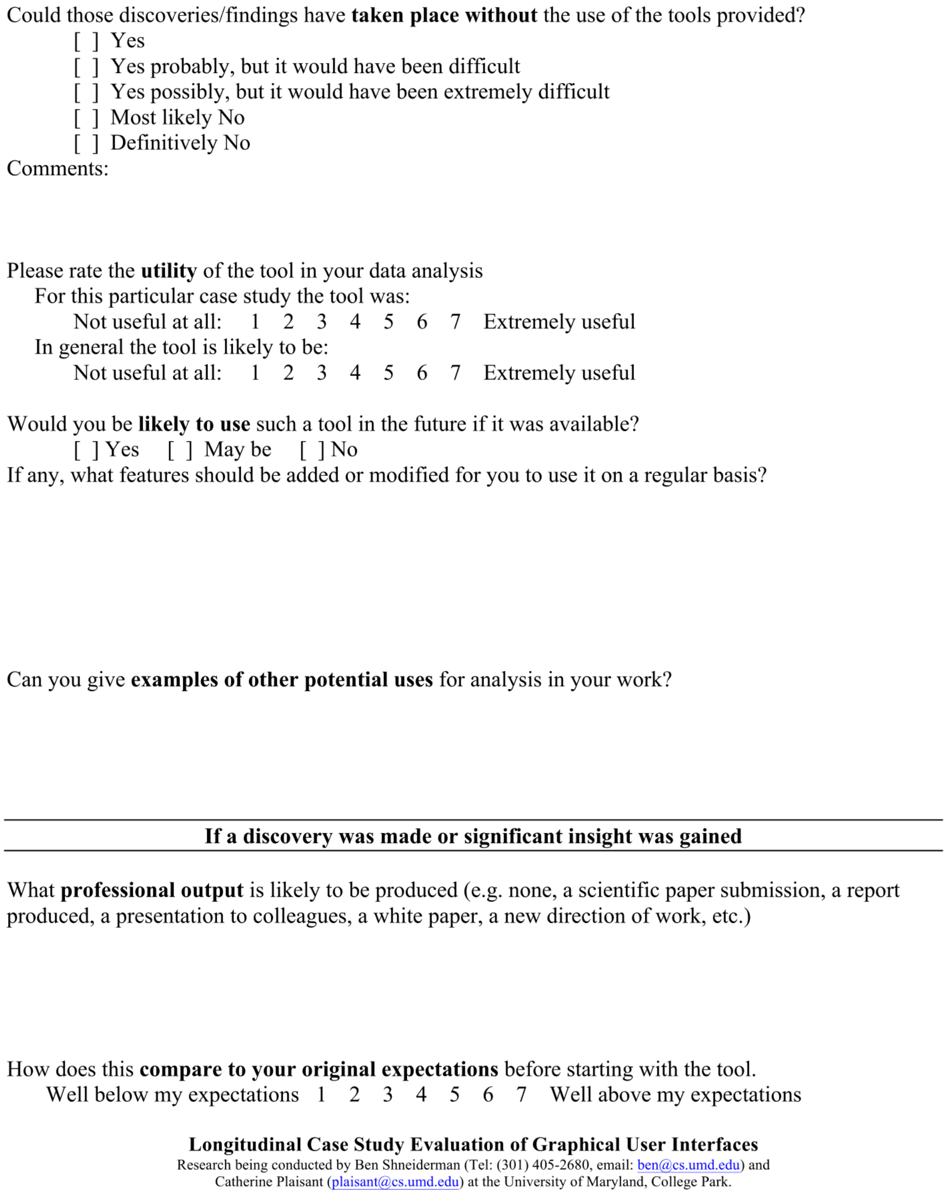
Tools being used: to analyze the data
HCIL tools _______________________________________

Other tools _______________________________________

Summarize in a paragraph **how you used the tools** HCIL provided (describe how it was used in conjunction with other tools if appropriate, and give indications of the amount of effort spent)

Summarize the **discovery made or the finding/insights gained** during the data analysis (provide references if necessary)

Figure 9.3: Exit questionnaire, page 1.



Could those discoveries/findings have **taken place without** the use of the tools provided?

[ ] Yes
[ ] Yes probably, but it would have been difficult
[ ] Yes possibly, but it would have been extremely difficult
[ ] Most likely No
[ ] Definitively No

Comments:

Please rate the **utility** of the tool in your data analysis

For this particular case study the tool was:

Not useful at all: 1 2 3 4 5 6 7 Extremely useful

In general the tool is likely to be:

Not useful at all: 1 2 3 4 5 6 7 Extremely useful

Would you be **likely to use** such a tool in the future if it was available?

[ ] Yes [ ] May be [ ] No

If any, what features should be added or modified for you to use it on a regular basis?

Can you give **examples of other potential uses** for analysis in your work?

If a discovery was made or significant insight was gained

What **professional output** is likely to be produced (e.g. none, a scientific paper submission, a report produced, a presentation to colleagues, a white paper, a new direction of work, etc.)

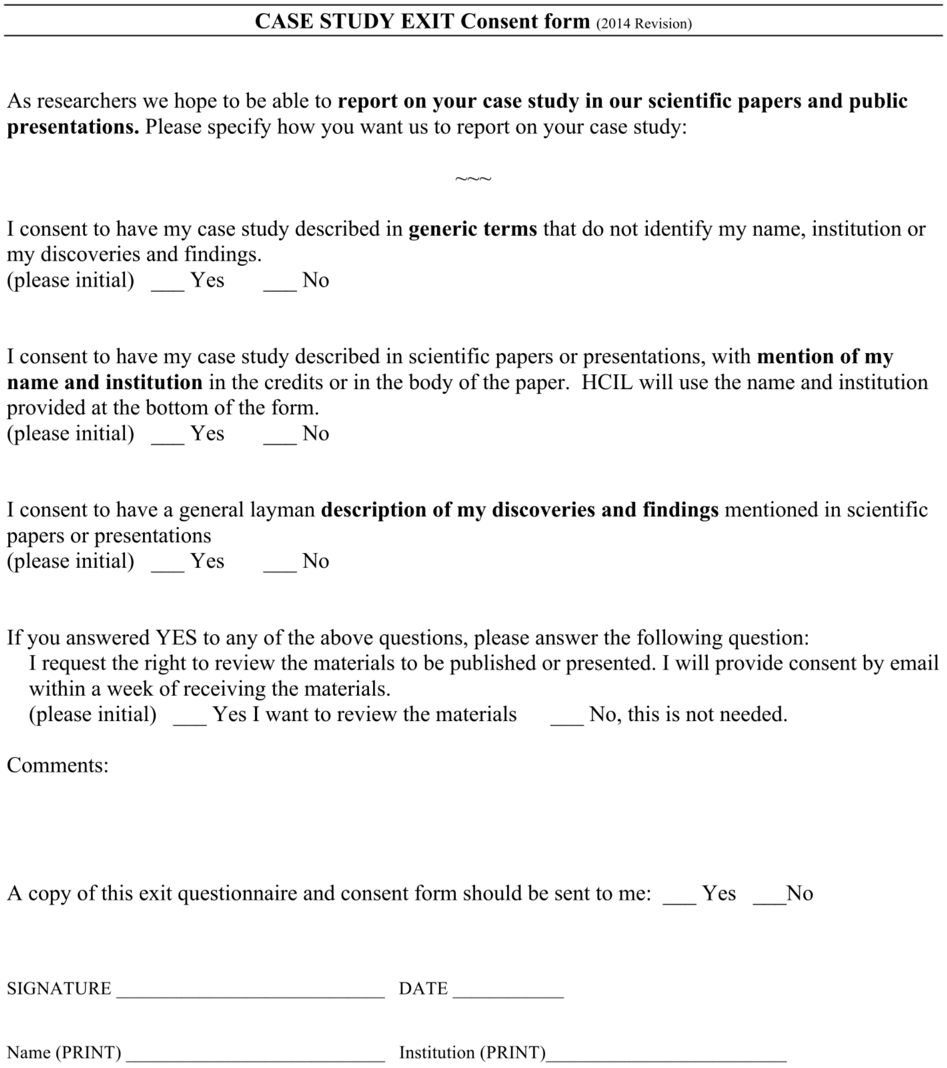
How does this **compare to your original expectations** before starting with the tool.

Well below my expectations 1 2 3 4 5 6 7 Well above my expectations

Longitudinal Case Study Evaluation of Graphical User Interfaces

Research being conducted by Ben Shneiderman (Tel: (301) 405-2680, email: ben@cs.umd.edu) and Catherine Plaisant (plaisant@cs.umd.edu) at the University of Maryland, College Park.

Figure 9.4: Exit questionnaire, page 2.



CASE STUDY EXIT Consent form (2014 Revision)

As researchers we hope to be able to **report on your case study in our scientific papers and public presentations.** Please specify how you want us to report on your case study:

~~~

I consent to have my case study described in **generic terms** that do not identify my name, institution or my discoveries and findings.
(please initial) ___ Yes ___ No

I consent to have my case study described in scientific papers or presentations, with **mention of my name and institution** in the credits or in the body of the paper. HCIL will use the name and institution provided at the bottom of the form.
(please initial) ___ Yes ___ No

I consent to have a general layman **description of my discoveries and findings** mentioned in scientific papers or presentations
(please initial) ___ Yes ___ No

If you answered YES to any of the above questions, please answer the following question:
I request the right to review the materials to be published or presented. I will provide consent by email within a week of receiving the materials.
(please initial) ___ Yes I want to review the materials ___ No, this is not needed.

Comments:

A copy of this exit questionnaire and consent form should be sent to me: ___ Yes ___No

SIGNATURE ____________________________ DATE ____________

Name (PRINT) ___________________________ Institution (PRINT)_________________________

Figure 9.5: Exit questionnaire, page 3.
~~~

Bibliography

[1] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, Dec 2013.

[2] Megan Monroe, Tamra E. Meyer, Catherine Plaisant, Rongjian Lan, Krist Wongsuphasawat, Trinka S. Coster, Sigfried Gold, Jeff Millstein, and Ben Shneiderman. Visualizing patterns of drug prescriptions with eventflow: A pilot study of asthma medications in the military health system. 2013.

[3] Elizabeth Carter, Randall Burd, Megan Monroe, Catherine Plaisant, and Ben Shneiderman. Using eventflow to analyze task performance during trauma resuscitation. *Proceedings of the Workshop on Interactive Systems in Healthcare (WISH 2013)*, 2013.

[4] John Alexis Guerra-Gómez, Krist Wongsuphasawat, Taowei David Wang, Michael L. Pack, and Catherine Plaisant. Analyzing incident management event sequences with interactive visualization. 2011.

[5] Krist Wongsuphasawat and David Gotz. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2659–2668, 2012.

[6] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, September 2011.

[7] Charles D. Stolper, Adam Perer, and David Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. volume 20, pages 1653–1662, 2014.

[8] Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. Matrixwave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 259–268, New York, NY, USA, 2015. ACM.

[9] Katerina Vrotsou, Anders Ynnerman, and Matthew Cooper. Are we what we do? exploring group behaviour through user-defined event-sequence similarity. *Information Visualization*, 13(3):232–247, 2014.

[10] Paul D Allison. Discrete-time methods for the analysis of event histories. *Sociological Methodology*, 13(1):61–98, 1982.

[11] Srivatsan Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, April 2006.

[12] Yuan Chen, Fiona Cunningham, Daniel Rios, William M McLaren, James Smith, Bethan Pritchard, Giulietta M Spudich, Simon Brent, Eugene Kulesha, Pablo Marin-Garcia, Damian Smedley, Ewan Birney, and Paul Flicek. Ensembl variation resources. *BMC genomics*, 11(1):293, January 2010.

[13] Marc Fiume, Vanessa Williams, Andrew Brook, and Michael Brudno. Savant: genome browser for high-throughput sequencing data. *Bioinformatics (Oxford, England)*, 26(16):1938–44, August 2010.

[14] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and a. D. Haussler. The Human Genome Browser at UCSC. *Genome Research*, 12(6):996–1006, May 2002.

[15] Helga Thorvaldsdóttir, James T Robinson, and Jill P Mesirov. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, 14(2):178–92, March 2013.

[16] Jun Wang, Lei Kong, Ge Gao, and Jingchu Luo. A brief introduction to web-based genome browsers. *Briefings in Bioinformatics*, 14(2):131–43, March 2013.

[17] Florin Chelaru, Llewellyn Smith, Naomi Goldstein, and Hector Corrada Bravo. Epiviz: interactive visual analytics for functional genomics data. *Nat Meth*, 11(9):938–940, September 2014.

[18] Miriah Meyer, Tamara Munzner, and Hanspeter Pfister. MizBee: a multiscale synteny browser. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):897–904, January 2009.

[19] Joel A Ferstay, Cydney B Nielsen, and Tamara Munzner. Variant view: visualizing sequence variants in their gene context. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2546–55, December 2013.

[20] Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J Byrne, Michael L Heuer, Erik Larsson, Yevgeniy Antipin, Boris Reva, Arthur P Goldberg, Chris Sander, and Nikolaus Schultz. The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer Discovery*, 2(5):401–4, May 2012.

[21] Nathan D Dees, Qunyuan Zhang, Cyriac Kandoth, Michael C Wendl, William Schierding, Daniel C Koboldt, Thomas B Mooney, Matthew B Callaway, David Dooling, Elaine R Mardis, Richard K Wilson, and Li Ding. MuSiC: identifying mutational significance in cancer genomes. *Genome Research*, 22(8):1589–98, August 2012.

[22] Peter F. Brown, Peter V. DeSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December 1992.

[23] Anthony Don, Elena Zheleva, Machon Gregory, Sureyya Tarkan, Loretta Auvil, Tanya Clement, Ben Shneiderman, and Catherine Plaisant. Discovering interesting usage patterns in text collections. In *Proc. 16th ACM Conference on Conference on Information and Knowledge Management - CIKM '07*, page 213, New York, USA, November 2007. ACM Press.

[24] Magdalena Jankowska, Vlado Keselj, and Evangelos Milios. Relative N-gram signatures: Document visualization at the level of character N-grams. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 103–112. IEEE, October 2012.

[25] Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proc. 2004 Conference on Human Factors in Computing Systems - CHI '04*, pages 575–582, New York, USA, April 2004. ACM Press.

[26] Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong Zhou. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. In *ACM SIGGRAPH 2003*, number 1, page 453, New York, USA, 2003. ACM Press.

[27] S Bremm, T von Landesberger, M Hess, T Schreck, P Weil, and K Hamacherk. Interactive visual comparison of multiple trees. In *Proc. 2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 31–40, 2011.

[28] Danny Holten and Jarke J van Wijk. Visual Comparison of Hierarchically Organized Data. *Computer Graphics Forum*, 27(3):759–766, May 2008.

[29] John Alexis Guerra-gómez, Michael L Pack, Catherine Plaisant, and Ben Shneiderman. Visualizing changes over time in datasets using dynamic hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2566–2575, 2013.

[30] Viv Bewick, Liz Cheek, and Jonathan Ball. Statistics review 12: survival analysis. *Critical Care*, 8(5):389–94, 2004.

[31] David Collett. *Modelling survival data in medical research (2nd. ed.)*. Chapman and Hall/CRC Press, 2003.

[32] Mathieu Dupont, Arnaud Gacouin, Hervé Lena, Sylvain Lavoué, Graziella Brinchault, Philippe Delaval, and Rémi Thomas. Survival of patients with bronchiectasis after the first ICU stay for respiratory failure. *Chest*, 125(5):1815–20, May 2004.

[33] Manish K. Goel, Pardeep Khanna, and Jugal Kishore. Understanding survival analysis: Kaplan-Meier estimate. *International Journal of Ayurveda Research*, 1(4):274–278, October 2010.

[34] Zhiyuan Zhang, David Gotz, and Adam Perer. Iterative cohort analysis and exploration. *Information Visualization*, Mar. 2014.

[35] Oracle. Oracle Health Sciences Cohort Explorer user's guide. Technical report, Oracle, 2011.

[36] John W. Tukey. *Exploratory Data Analysis*. Pearson, 1st edition, 1977.

[37] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4):1165–1188, August 2001.

[38] Juliet P. Shaffer. Multiple hypothesis testing. *Annual Review of Psychology*, 46(1):561–584, 1995.

[39] Guimei Liu, Mengling Feng, Yue Wang, Limsoon Wong, See-Kiong Ng, Tzia Liang Mah, and Edmund Jon Deoon Lee. Towards exploratory hypothesis testing and analysis. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 745–756, Washington, DC, USA, 2011. IEEE Computer Society.

[40] M. Gupta, Jing Gao, C.C. Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, Sept 2014.

[41] Nizar R. Mabroukeh and Christie I. Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys*, 43(1):3:1–3:41, Nov. 2010.

[42] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[43] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pages 207–216, New York, NY, USA, Jun. 1993. ACM.

[44] Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.

[45] Miguel R Álvarez, Paulo Félix, and Purificación Cariñena. Discovering metric temporal constraint networks on temporal databases. *Artificial Intelligence in Medicine*, 58(3):139–54, July 2013.

[46] Riccardo Bellazzi, Lucia Sacchi, and Stefano Concaro. Methods and tools for mining multivariate temporal data in clinical and biomedical applications. *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, 2009:5629–32, January 2009.

[47] S Concaro, L Sacchi, C Cerra, P Fratino, and R Bellazzi. Mining health care administrative data with temporal association rules on hybrid events. *Methods of Information in Medicine*, 50(2):166–79, January 2011.

[48] Yong Joon Lee, Jun Wook Lee, Duck Jin Chai, Bu Hyun Hwang, and Keun Ho Ryu. Mining temporal interval relational rules from temporal data. *Journal of Systems and Software*, 82(1):155–167, 2009.

[49] Philippe Fournier-Viger, Usef Faghihi, Roger Nkambou, and Engelbert Mephu Nguifo. CMRules: Mining sequential rules common to several sequences. *Knowledge-Based Systems*, 25(1):63–76, February 2012.

[50] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 11th International Conference on Data Engineering*, pages 3–14. IEEE Comput. Soc. Press, 1995.

[51] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, September 1997.

[52] Adam Perer and Fei Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 153–162, New York, USA, 2014. ACM.

[53] G. Niklas Norén, Johan Hopstadius, Andrew Bate, Kristina Star, and I. Ralph Edwards. Temporal pattern discovery in longitudinal electronic patient records. *Data Mining and Knowledge Discovery*, 20(3):361–387, November 2009.

[54] Robert Moskovitch and Yuval Shahar. Medical temporal-knowledge discovery via temporal abstraction. *Proc. AMIA Annual Symposium*, 2009:452–6, January 2009.

[55] Denis Klimov, Yuval Shahar, and Meirav Taieb-Maimon. Intelligent visualization and exploration of time-oriented data of multiple patients. *Artificial Intelligence in Medicine*, 49(1):11–31, May 2010.

[56] Iyad Batal, Lucia Sacchi, Riccardo Bellazzi, and Milos Hauskrecht. A temporal abstraction framework for classifying clinical temporal data. *Proc. AMIA Annual Symposium*, 2009:29–33, January 2009.

[57] Katerina Vrotsou and Aida Nordman. Interactive visual sequence mining based on pattern-growth. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST '14)*, pages 285–286, Oct 2014.

[58] Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Silvia Miksch, and Alexander Rind. Special section on visual analytics: Mind the time: Unleashing temporal aspects in pattern discovery. *Computer Graphics*, 38:38–50, February 2014.

[59] Paolo Federico, Jürgen Unger, Albert Amor-Amors, Lucia Sacchi, Denis Klimov, and Silvia Miksch. Gnaeus: Utilizing clinical guidelines for knowledge-assisted visualisation of EHR cohorts. In Enrico Bertini and Jonathan C. Roberts, editors, *Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA '15)*. The Eurographics Association, 2015.

[60] Danyel Fisher, Igor Popov, Steven Drucker, and m.c. schraefel. Trust me, i'm partially right: Incremental visualization lets analysts explore large datasets faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1673–1682, New York, NY, USA, 2012. ACM.

[61] B. Preim, P. Rheingans, H. Theisel, Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data, 2013.

[62] Timos K. Sellis. Multiple-query optimization. *ACM Trans. Database Syst.*, 13(1):23–52, March 1988.

[63] Megan Monroe. *Interactive Event Sequence Query and Transformation*. PhD thesis, University of Maryland, College Park, MD, USA, 2014.

[64] TIBCO. Spotfire. http://spotfire.tibco.com/, Mar 2014.

[65] Tableau Software. Tableau. http://www.tableausoftware.com/, Mar 2014.

[66] Armin Ronacher. Flask (a python microframework), apr 2016.

[67] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2016-04-07].

[68] Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[69] Taowei David Wang, Krist Wongsuphasawat, Catherine Plaisant, and Ben Shneiderman. Visual information seeking in multiple electronic health records: Design recommendations and a process model. In *Proceedings of the 1st ACM International Health Informatics Symposium*, IHI '10, pages 46–55, New York, NY, USA, 2010. ACM.

[70] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, November 2011.

[71] Ben Shneiderman and Catherine Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–7, New York, NY, USA, 2006. ACM.

[72] Susan E Andrade, Kristijan H Kahler, Feride Frech, and K Arnold Chan. Methods for evaluation of medication adherence and persistence using automated databases. *Pharmacoepidemiology and drug safety*, 15(8):565–574, 2006.

[73] Sana Malik and Eunyee Koh. High-volume hypothesis testing for large-scale web log analysis. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 1583–1590, New York, NY, USA, 2016. ACM.

[74] Michelle K. Smith, Francis H. M. Jones, Sarah L. Gilbert, and Carl E. Wieman. The classroom observation protocol for undergraduate stem (copus): A new instrument to characterize university stem classroom practices. *CBE-Life Sciences Education*, 12(4):618–627, 2013.

[75] *Using EventFlow and CoCo to explore classroom activity patterns and learner performance*, 2016.

[76] Nalini Sathiakumar, Elizabeth Delzell, Michael Morrisey, Carla Falkson, Mellissa Yong, Victoria Chia, Justin Blackburn, Tarun Arora, and Meredith Kilgore. Mortality following bone metastasis and skeletal-related events among patients 65 years and above with lung cancer: A population-based analysis of U.S. Medicare beneficiaries, 1999-2006. *Lung India*, 30(1):20–26, 2013.

[77] Mette Nørgaard, Annette Østergaard Jensen, Jacob Bonde Jacobsen, Kara Cetin, Jon P. Fryzek, and Henrik Toft Sørensen. Skeletal related events, bone metastasis and survival of prostate cancer: A population based cohort study in denmark (1999 to 2007). *The Journal of Urology*, 184(1):162–167, 2015/10/06.

[78] MJ Lage, BL Barber, DJ Harrison, and S Jun. The cost of treating skeletal-related events in patients with prostate cancer. *Am J Manag Care*, 14(5):317–22, may 2008.

[79] William F. Hartsell, Charles B. Scott, Deborah Watkins Bruner, Charles W. Scarantino, Robert A. Ivker, Mack Roach, John H. Suh, William F. Demas, Benjamin Movsas, Ivy A. Petersen, Andre A. Konski, Charles S. Cleeland,

Nora A. Janjan, and Michelle DeSilvio. Randomized trial of short- versus long-course radiotherapy for palliation of painful bone metastases. *Journal of the National Cancer Institute*, 97(11):798–804, 2005.

[80] Stephen T. Lutz, Joshua Jones, and Edward Chow. Role of radiation therapy in palliative care of the patient with cancer. *Journal of Clinical Oncology*, 2014.

[81] 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*, PP(99):1–1, 2016.

[82] Tanushree Mitra and Eric Gilbert. Credbank: A large-scale social media corpus with associated credibility annotations, 2015.

[83] Cody Buntain, Jennifer Golbeck, Brooke Liu, and Gary LaFree. Evaluating public response to the boston marathon bombing and other acts of terrorism through twitter, 2016.

[84] LLC. Sports Reference. War explained.

[85] Ronald Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.

[86] Matthew Louis Mauriello, Ben Shneiderman, Fan Du, Sana Malik, and Catherine Plaisant. Simplifying overviews of temporal event sequences. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 2217–2224, New York, NY, USA, 2016. ACM.