Info Vis

Article

Information Visualization
1–19

DOI: 10.1177/1473871614526077
ivi.sagepub.com
SAGE

Iterative cohort analysis and exploration

Zhiyuan Zhang[1], David Gotz[2] and Adam Perer[3]

Abstract

Cohort analysis is a widely used technique for the investigation of risk factors for groups of people. It is commonly employed to gain insights about interesting subsets of a population in fields such as medicine, bioinformatics, and social science. The nature of these analyses is evolving as larger collections of data about individuals become available. Examples of emerging large-scale data sources include electronic medical record systems and social network datasets. When domain experts perform cohort analyses using such massive datasets, they typically rely on a team of technologists to help manage and process the data. This results in a slow and cumbersome analysis process in which iterative exploration is difficult. To address this challenge, we are exploring technologies designed to help domain experts work more independently and more quickly. This article describes *CAVA*, a platform for Cohort Analysis via Visual Analytics. We introduce three primary types of artifacts (cohorts, views, and analytics) and an architecture that connects these elements together to provide an interactive exploratory analysis environment designed for domain experts. In addition to the CAVA design, this article presents two use cases from the health-care domain and a domain-expert evaluation to demonstrate the power of our approach.

Keywords

Visual analytics, cohort analysis, interactive systems, information visualization, health care

Introduction

Cohort analysis is a common technique used in a variety of fields to study risk factors within population groups. In fields as diverse as health care and ecology, the cohort study is a foundational tool that helps experts uncover correlations between specific risk metrics and the underlying attributes of individuals within the study population.

Cohort studies are often performed prospectively using techniques that are statistically mature and powerful. However, the analytical process is often slow and expensive when collecting data prospectively. Retrospective analyses, which use previously collected data, are a possible alternative. Unfortunately, the use of retrospective studies has been relatively limited due to the historical difficulty in collecting and analyzing very large datasets. However, as more and more data become electronic, very large repositories suitable for retrospective cohort analysis are becoming increasingly common. For example, large medical institutions are now adopting electronic medical record (EMR) systems in increasing numbers. These data warehouses can contain comprehensive historical observations of millions of people over time spans of many years.

The increasing availability of such data helps overcome the fundamental limitations of the retrospective

[1]The State University of New York at Stony Brook, Stony Brook, NY, USA
[2]The University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
[3]IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA

Corresponding author:
David Gotz, The University of North Carolina at Chapel Hill, 206 Manning Hall, CB #3360, Chapel Hill, NC 27599, USA.
Email: gotz@unc.edu

approach. In theory, domain experts can use these data to perform interactive, exploratory cohort studies without the overheads associated with prospective techniques. In practice, however, interactive cohort studies exploring large-scale retrospective data collections produce their own set of challenges. Data management, analysis, and summarization all become more difficult and typically lead to the use of more advanced technologies. Instead of relying on a spreadsheet and some basic statistics, users must also use technologies such as databases, data mining, and visualization tools to help make sense of the large scale of data they wish to examine.

The end result is that domain-expert users are still critically constrained. When new hypotheses are developed, users no longer have to design a new prospective study. However, they now need to speak with a team of technologists to perform data transformations, run data mining routines, and visualize the results. This process can be both slow and expensive, and the domain experts are still unable to quickly perform iterative and exploratory analyses on their own.

To help address this challenge, we have designed *CAVA*—a platform for Cohort Analysis via Visual Analytics—which is designed to help domain experts work faster and more independently when performing retrospective cohort studies (Figure 1). Motivated by the needs of real-world analysts working in the health-care domain, CAVA follows a novel system design centered around three primary types of artifacts: (1) cohorts, (2) views, and (3) analytics. *Cohorts* are CAVA's fundamental data construct and represent a set of people and their associated properties. *Views* are visualization components that graphically display a cohort and allow users to directly manipulate or refine the underlying cohort. *Analytics* are computational elements that create, expand, and/or alter the contents of a cohort. In this way, CAVA treats both Views and Analytics as *functional components* which operate on an *input cohort* and produce an *output cohort*. Building on this design principle, CAVA allows users to chain together complex sequences of steps that intermix both manual and machine-driven cohort manipulations. This capability is provided through an easy-to-use, web-based user interface that supports an interactive exploratory analysis environment for retrospective cohort studies.

This article describes CAVA in more detail, beginning with a discussion of the user requirements we identified in the health-care domain that motivate our work. Then, after a brief review of related work, we introduce the CAVA design and highlight how user requirements drove several key aspects of our approach. We further demonstrate the utility of our approach by describing our prototype CAVA implementation and introducing two use cases where CAVA was used to analyze data from a population of at-risk medical patients. Finally, we conduct an evaluation to justify the usability and the applicability of our approach.

Motivating scenario in health-care domain

CAVA is designed to provide a general solution for interactive cohort analysis. However, the design decisions embodied in CAVA are motivated by a set of real-world requirements distilled from our target domain: health-care population analysis.

Our work was originally inspired by a problem faced by a group of physicians, who, together with a team of technologists, were trying to uncover new insights about a population of cardiology patients being treated at their institution by analyzing a collection of electronic medical data. Because the cardiologists work in a relatively large medical organization, the doctors have access to longitudinal records from hundreds of thousands of patients, each with tens of thousands of features. Many organizations have access to even more data, often storing millions of patient records spanning decades of historical treatments.

As one would expect of highly trained domain experts, the cardiologists have extensive medical knowledge and intimate familiarity with the various lab tests, diagnoses, and other pieces of information in the dataset. Unsurprisingly, however, these same clinicians have relatively limited technology skills. For this reason, a team of technologists—software engineers, database administrators, data miners, and visualization designers—work together with the clinicians to gather and transform data, build systems to perform analysis, and present the results for discussion.

The typical workflow is as follows: first the physicians propose several hypotheses; based on these hypotheses, a database expert prepares the necessary Structured Query Language (SQL) queries and data transformation scripts to gather data for the patient cohorts specified by the clinician's hypotheses. Then, data analysts work with the patient data to build models that process and extract additional information (e.g. risk scores) about the patients in the cohorts. Then, finally, the analytical results are visualized for the clinicians who re-engage to see whether the data support their hypotheses.

Of course, the clinicians' hypotheses may be wrong. Therefore, this process is typically repeated iteratively as doctors explore new alternatives: "There were only 10 patients in that group? What if we change constraints?" "What about women, do they have a similar distribution?" "What risk scores do these patients have for other conditions?" "Are there more patients like

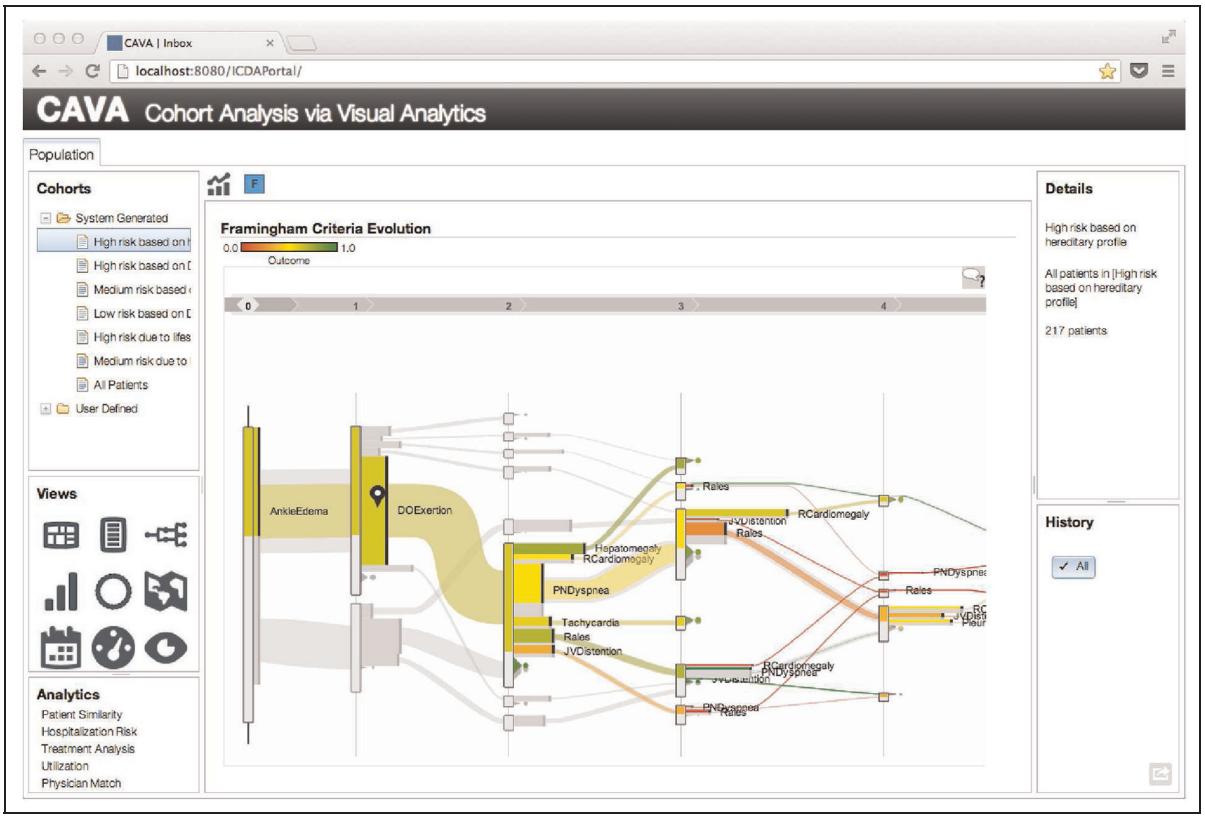

Figure 1. CAVA, shown here being applied to a population of cardiac patients, provides an interactive visual analytics environment for iterative exploratory cohort analysis. Through CAVA, users can chain together complex analysis pathways that combine interactive cohort refinements with automated analytics algorithms in an ad hoc exploratory fashion.
CAVA: Cohort Analysis via Visual Analytics.

these and do they have similar outcomes?" Moreover, even when hypotheses appear to be correct, they often lead to a large number of follow-up questions. Therefore, nearly all hypotheses eventually result in additional work for the team of technologists who diligently work to help answer the clinicians' subsequent questions.

As one can see from this iterative workflow, the reliance on technology professionals as intermediaries can result in a slow and cumbersome process. Ideally, clinicians would be able to independently conduct ad hoc exploration and analysis: visually defining and refining cohorts, and requesting interactive analytics, and looking at the results without any manual assistance. It is this ideal goal that we are striving to reach in our work.

Motivated by the health-care cohort analysis scenario outlined above, we have identified the following set of key requirements that should be satisfied in an ideal solution:

- *Easy cohort definition.* Clinical domain experts should be able to easily select cohorts of patients for investigation. This can be in the form of both computational analyses which automatically identify interesting cohorts from a large population and visual interfaces that allow clinicians to define cohorts ad hoc.
- *Flexible visualization.* Clinical domain experts should be able to flexibly visualize cohorts, pivoting between various visualization metaphors as part of an interactive exploratory process without requiring any intervention by technology experts.
- *Flexible analysis.* Clinical domain experts should be able to flexibly perform analyses of various kinds on a cohort without requiring any intervention by technology experts.
- *Cohort refinement and expansion.* Clinical domain experts should be able to easily constrain and/or expand cohorts based on discovered findings as part of their exploration.
- *Iterative analysis.* The above requirements should be supported within an iterative process that allows refinement and exploration during an open-ended investigation.

These design requirements were informed by unstructured interviews about workflow and task requirements conducted with a number of clinicians. As part of ongoing research collaborations on a range of medical informatics projects, we have captured input from physicians with different specialties, home institutions, and work experiences. The length of time spent interacting with each clinician has varied widely, ranging from a few hours to years of close collaboration. Using the insights obtained from our clinical collaborators, we developed the consolidated set of common requirements presented here.

These requirements form CAVA's core set of design guidelines as reflected in section "System design." Moreover, the use cases outlined in section "Use cases" have been included to demonstrate how our approach helps satisfy these requirements when applied to our motivating problem domain. Finally, key benefits from our approach are documented in the results of the evaluation presented in section "Domain-expert evaluation."

Related work

This section provides an overview of related work. We focus on techniques most relevant to CAVA, including general cohort analysis, data analysis algorithms, visual analytics techniques, and visual analytics systems applied to health care and beyond.

Cohort analysis

A widely used technique in fields such as ecology, bioinformatics, social science, and health care, cohort analysis is a research method for analyzing changes in group members through the use of a set of statistical techniques.[1] One of the most common ways that cohort analysis is used is to analyze medical risk factors in clinical studies.[2] Often, such studies are designed to follow a group of people without a disease. Based on longitudinal observations, a correlation analysis is performed to determine the risk of a subject contracting the disease by correlation analysis and the relative importance of various risk factors. In epidemiology,[3] similar techniques are used to find correlations or causal relations between a given disease and the exposure to certain environmental conditions or behaviors.

Most often, cohort studies are performed prospectively. Cohorts of interest are defined in advance based on an expert's initial hypotheses (e.g. defining cohorts with and without a given pharmacological exposure, with various controls over the population's characteristics). A predetermined set of data for the individuals in these cohorts is then gathered over time and eventually analyzed to uncover significant correlations. While this approach can produce very valuable insights, prospective studies of this form typically require expensive data gathering efforts and take significant time to design and execute. Moreover, any changes to the initial hypotheses most often result in additional time-consuming studies with new populations. This burden limits a domain expert's ability to easily explore and refine hypotheses based on his/her findings.

In contrast, retrospective cohort analyses offer significantly more flexibility. Because these studies utilize already-captured data records, an expert can investigate new hypotheses or find valuable correlation without the time or expense associated with designing and running an additional prospective study. However, the method of retrospective study also has issues. While valuable correlations may be discovered, such a technique is not suitable to prove causation. Moreover, this approach is often at higher risk to bias and may miss rare events. For this reason, retrospective studies are most effective when applied to larger sample sizes. Nevertheless, the large scale of retrospective data often requires a domain expert to work with a team of technology experts (i.e. database support, data mining, and visualization expertise) to explore alternative hypotheses.

Data analysis algorithms

Given the large scale that is typical of retrospective data—both in terms of population sizes and the number of data features available per person—data analysis algorithms have been widely used to support cohort analyses tasks. They automatically process large collections of population data to classify or segment a population or to compute new derived features. For example, similarity analysis can be used to identify a cohort of similar patients given a target index patient, which can then be used as the basis for decision support.[4] Similarly, Ebadollahi et al.[5] utilized similarity analysis for near-term prognostics for physiological data. Chattopadhyay et al.[6] used similarity for risk assessment. Huxley et al.[7] estimate the relative risk of fatal coronary heart disease associated with diabetes by inverse variance and meta-analysis of 37 prospective cohort studies. Seid et al.[8] adopted the method of PedsQL 4.0 Generic Core Scales[9] and other statistical analysis methods such as multiple linear regression analyses and conducted a 2-year prospective cohort analysis. Other examples of analytics include predictive models that generate new derived features (e.g. disease-specific risk scores) after being trained from large sets of population data.[10]

These algorithms allow for the efficient processing of large amounts of data. However, they typically work like a black box with users having little or no control

over the analytical pipeline. The lack of user input makes it impractical for exploratory analysis where users need to explore alternative hypotheses.

Visual analytics techniques

To help capture user input and make the analysis process more interactive, visual analytics techniques combine visualization, data mining, and statistics with interactive interfaces that allow the users to apply their domain knowledge.[11–13] Visual analytics can help overcome limitations with the classical one-way data analysis process by letting the user become directly involved influencing the way data are processed.[14] A similar concept was also proposed in the field computational steering,[15] which lets the user lead the exploration of simulations, with prompts to direct the simulation. All of these techniques allow the users to apply their domain knowledge in the traditional automatic analytical/computational process, and visualizations are used to provide immediate feedback to guide the following steps. CAVA is designed similarly, allowing the user to directly influence the inputs, control parameters, and timing of execution for supported data analysis computations.

A key element of any visual analytics system is visualization. Interactive visualization techniques can provide domain-expert users with intuitive data representations that can be quickly understood, explored, and manipulated. Complementing more traditional charting methods (such as tables, bar charts, and histograms), an enormous variety of sophisticated techniques have emerged from the information visualization community over the years. Many of these designs can be effectively employed in a cohort analysis context and our CAVA prototype includes components that implement several well-known visual metaphors. Given the nature of many cohort analyses, two types of visualizations are often highly critical: (1) hierarchical or multi-dimensional data visualizations to segment based on complex sets of population attributes[16–21] and (2) temporal visualizations to navigate events over time in populations' longitudinal data.[22–29] For populations that include geographic information, maps are also a powerful visual metaphor.

The prototype implementation of CAVA described in this article supports several different visualization types. For example, treemaps are used as a display for hierarchical data[30] and are coded to support selection which can coordinate with other linked views. Similarly, CAVA employs a temporal visualization view that is a generalized extension of the Outflow technique.[27,31,32] Recognizing that different types of cohort analyses can require different sets of visualization capabilities, CAVA's design allows for the easy integration of additional views without changes to the underlying cohort analysis platform.

Another aspect of visual analytics research related to CAVA is analytics provenance. Many projects have explored techniques for capturing and modeling a user's analytical process history. For example, Jankun-Kelly et al.[33] introduced the P-Set model of visualization exploration and a framework to encapsulate, share, and analyze visual explorations. Perer and Shneiderman[34] designed a systematic yet flexible framework that allows analysts to take exploratory excursions while keeping track of overall progress. In related work, Shrinivasan and Van Wijk[35] presented an information visualization framework that captures the analytical reasoning process via interaction with multiple views. The views are used for data visualization, recording analysis artifacts, or representing the analysis states. Gotz and Zhou[36] characterized users' visual analytic activity at multiple levels of granularity and then identified a critical level of abstraction, Actions, that can be used to represent visual analytic activity with a set of general but semantically meaningful behavior types. CAVA also captures a user's history and exposes it through the user interface for inspection and manipulation. CAVA's history is modeled in terms of its key artifacts: cohorts, views, and analytics.

Systems and applications

Recently, Rind et al.[37] gave a survey of tools and systems in the health-care field and showed that effective information visualization can facilitate analysis of electronic health records (EHRs) for patient treatment and clinical research. A number of cohort-focused visual analytics systems have been developed and applied to specific types of applications. For example, in the domain of health-care analytics, Steenwijk et al.[38] proposed a visual analysis framework for cohort studies of heterogeneous data using mappers to connect features across domains. Cao et al.[39] designed DICON, a tool that allows users to interactively view and refine patient clusters. Gotz et al.[40] extended DICON and integrated it into a similarity-based clinical decision intelligence system. Lins et al.[41] developed VisCareTrails, a system that captures patients' medical events and summarizes event paths. Chui et al.[42] introduced a system for disease monitoring and biosurveillance that uses a multi-panel graph to integrate temporality and demographics. Perer and Sun[43] introduced MatrixFlow that analyzes temporal patterns of co-occurring clinical events and supports comparison across cohorts. In all of these systems, however, the analysis follows a pre-defined flow. Our work, in contrast, allows flexible, user-composed workflows that combine analytics and visualization. CAVA builds

upon the early work of Zhang et al.[44] who described a preliminary prototype for interactive cohort analysis. In this article, we extend this approach and propose a standardized model and platform for exploratory and iterative cohort analysis workflows.

Finally, while we have focused on the health-care domain as our motivating scenario, similar problems have been explored in other domains. For instance, Ferreira et al.[45] built a system that supports the analysis of spatiotemporal bird distribution models. Dou et al.[46] introduced ParallelTopics that dealt with large document collections. Xu et al.[47] presented a system that can analyze large-scale digital collections for archival purposes. Each of these addresses an application domain featuring the analysis and exploration of collections of entities. Therefore, our CAVA framework could be extended to support these topics by developing an appropriate set of views and analytic components. For example, to explore document collections, an analytic component for Latent Dirichlet allocation (LDA) topic modeling (as used in Dou et al.[46]) could be developed.

System design

CAVA is designed to meet the key requirements outlined in our motivating application scenario. It allows users to select, visualize, analyze, and refine cohorts obtained from large population-oriented datasets. Moreover, these steps can be performed iteratively as part of complex ad hoc analytical workflows. This section provides a detailed discussion of this design, beginning with a discussion of CAVA's three key artifacts: cohorts, analytics, and views. We then describe how CAVA binds these artifacts together into an integrated architecture. Finally, we discuss CAVA's user interface and the prototypical workflow that the system supports. As concepts are presented, we adopt the notations defined in Table 1 which we then use throughout the remainder of this article.

Key artifacts

There are three key artifacts at the core of CAVA's design. First, CAVA's primary *data artifact* is the cohort which represents a collection of individuals selected from an overall population. Cohorts are then manipulated by two different types of *operational artifacts*: analytics and views.

Cohorts. A *cohort* is CAVA's most important data construct. We define a cohort C_i as a set of individual members m_z such that $C_i = \{m_z\}$. In addition to its membership, a cohort has global properties, such as a label (i.e. a human-consumable name for display through a user interface) and aggregate statistical summaries of the underlying membership. Each member m_z has associated with it a feature vector, noted as $\vec{f}_{m_z}$. This feature vector contains the set of all information known about the corresponding member.

Table 1. A summary of the notation used in this article.

Notation	Description
C_i	A cohort
m_z	An individual member of a cohort
$\vec{f}_{m_z}$	The feature vector for member m_z
$F_{C_i}^{in}$	The inner feature set for C_i
$F_{C_i}^{out}$	The outer feature set for C_i
A_j	An analytic (a type of operational artifact)
V_k	A view (a type of operational artifact)
$\vec{\alpha}_j$	A vector of input parameters for operational artifact j
$\vec{\alpha}_k$	A vector of input parameters for operational artifact k
$F_{A_j}^{pre}$	The prerequisite feature set for A_j
$F_{V_k}^{pre}$	The prerequisite feature set for V_k

For example, in the health-care domain described in our motivating scenario, a cohort would represent a set of patients. A potentially large number of features may be associated with each member in the cohort, such as a patient's demographics (age, gender, etc.), diagnoses, treatments, and lab results.

An important aspect of this definition is the potential for large variations between different cohort members' feature vectors. Unlike data that have been carefully curated for prospective cohort studies, retrospective data can be sparse, irregular, and suffer from many missing values. For example, consider the health-care scenario. The many member patients in a cohort are likely to have different sets of co-morbidities and to have undergone a wide range of lab tests and treatments. In addition, derived data for a given patient—such as computed risk scores—may only be available for portions of the population. In practice, individual patients are not identical.

For these reasons, the data contained in $\vec{f}_{m_z}$ can vary widely between $m_z \in C_i$. Based on this observation, we define two additional global properties for a cohort: (1) the inner feature set and (2) the outer feature set. Adopting the semantics from SQL's inner and outer joins, the inner feature set, $F_{C_i}^{in}$, is the set of features that are present in all members of the cohort

$$F_{C_i}^{in} = \bigcap_{m_z \in C_i} \vec{f}_{m_z} \quad (1)$$

This is in contrast to the outer feature set, $F_{C_i}^{out}$, which represents the union of all features found at least once in the cohort's membership

$$F_{C_i}^{out} = \bigcup_{m_z \in C_i} \vec{f}_{m_z} \quad (2)$$

The sparsity typical of retrospective data corpora means that $F_{C_i}^{in}$ is generally much smaller than $F_{C_i}^{out}$ for a given cohort. This fact becomes important during the cohort binding process described in section "Architecture overview."

During typical operation, CAVA maintains a collection of multiple cohorts. First, a special pre-defined cohort, P, is used to represent the set of all members in an overall population. For instance, P might contain all patients in a cardiology department's medical record system. A number of other cohorts—such as a group of elderly men or a group of patients at risk of hospital readmission—are then defined as subsets of P using the operational artifacts described later in this section: analytics and views. In this way, P is the superset for all CAVA cohorts.

Analytics. Analytics are the first of two distinct types of operational artifacts in CAVA. Unlike cohorts, which represent data, operational artifacts represent components that manipulate data in some way, converting an input cohort C_i into a newly modified output cohort C_i'.

An *analytic*, noted A_j, is a specific type of operational artifact which applies a computational algorithm to C_i in order to produce the output result. A CAVA system contains a collection of one or more analytic components, each responsible for performing a distinct analytical function. Along with an input cohort, many analytic algorithms expose additional setting or control parameters. Therefore, CAVA allows individual analytic components A_j to require a custom vector of additional input parameters, noted as $\vec{\alpha}_j$. Each A_j can have its own specification for what values are required as part of $\vec{\alpha}_j$, which typically reflects an algorithm's control parameters, such as thresholds or settings. Given this formulation, we model analytic A_j as the function defined in equation (3)

$$A_j(C_i, \vec{\alpha}_j) = C_i' \quad (3)$$

Analytic components can modify cohorts in two distinct ways. First, analytics can *refine the membership* of a cohort by adding and/or removing members. For example, an analytic might perform a similarity analysis to grow a cohort by finding "more people like these." Analytics that refine membership may also, indirectly, impact global properties of a cohort. Second, analytics can *refine the feature space* for a cohort by adding, removing, or updating features from the cohort members' feature vectors. Most typically, such analytics compute new types of feature, expanding $F_{C_i}^{in}$. For example, an analytic in the health-care domain could be developed to derive a body mass index (BMI) score (BMI is calculated from a patient's height h (in meters) and weight w (in kilograms)[48] as follows: $BMI = w/h^2$) for all patients in a cohort using their corresponding height and weight measurements.

In the example above, computing BMI requires access to height and weight measurements. This demonstrates that some analytics may require that certain features be present in $F_{C_i}^{in}$ to function properly. For example, the BMI analytic described above would require that each member of the input cohort have both height and weight. We refer to these required features as the *prerequisite feature set*, which we note as $F_{A_j}^{pre}$.

CAVA supports two different types of analytic components: (1) interactive analytics and (2) batch analytics. Both interactive and batch analytics adhere to the definition presented in equation (3). However, as described in more detail in section "Architecture overview," they are treated quite differently by the CAVA platform.

An *interactive analytic* operates on an input cohort synchronously. It executes immediately upon request and blocks any further user interaction until it returns C_i' upon completion. In the user interface, this is reflected with a progress bar indicator. Interactive analytics are used for relatively fast computations where the time required for execution is sufficiently minimal that the delay is acceptable within an interactive user interface. For example, the BMI calculator described above would be encapsulated as an interactive analytic because the arithmetic calculations required to produce a BMI score (given the raw height and weight values for all patients in a cohort) can be performed very quickly.

In contrast, *batch analytics* operate asynchronously. Designed for long-running calculations, batch analytics work in the background and save resulting C_i' cohorts for subsequent analysis when users return to see the results. This is in contrast to interactive analytics which return their results immediately to a user. This model allows users to launch long-running analytics in the background while still continuing their interactive analysis process.

For example, a batch analytic in the health-care scenario might perform risk stratification for large populations using complex high-dimensional calculations that cannot be performed at interactive rates for large input cohorts. A user could initiate such a risk stratification batch analysis on a group of interesting patients, then immediately return to visually explore other aspects of the cohort without waiting for the computation to terminate.

The example above describes how users can initiate batch analytic processes. However, batch analytics can

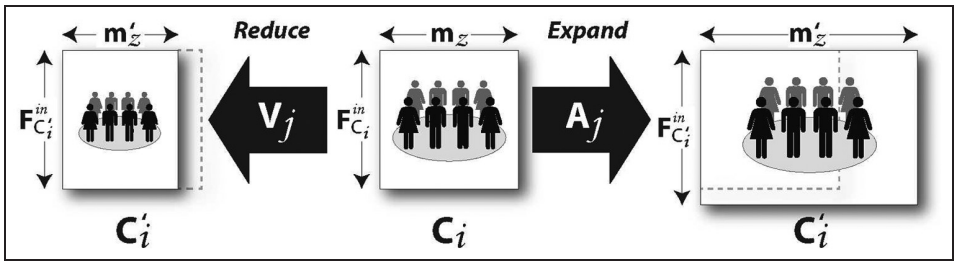

Figure 2. Both analytics and views are operational artifacts which manipulate a given cohort. Analytics (A_j) can be used to computationally change a cohort's membership and/or expand its inner feature set. In contrast, views (V_j) allow manual interaction (e.g. visual filtering) to drive the cohort manipulation process. Most typically, views are used to reduce a cohort while analytics expand a cohort.

also be executed automatically by the CAVA system itself. As described in section "Typical CAVA workflow," CAVA can run a set of batch analytics to bootstrap the system with an initial set of system-generated cohorts to serve as starting points for users' interactive analyses.

Views. The second type of operational artifact in CAVA is the *view*, noted as V_k. Like analytics, views are components that manipulate an input cohort C_i and produce a new output cohort C'_i for further analysis. However, views do not rely primarily on computational algorithms to compute C'_i. Instead, views are visualization-based display components that rely on a user's interactions to modify C_i. In general, a CAVA system contains a collection of views designed for various purposes. Some views may be general in scope, while others may be designed for a more narrow and specific task.

As with analytic components, a view V_k can be initialized with a vector of values $\vec{\alpha}_k$ which contains view-specific input parameters. For example, $\vec{\alpha}_k$ could include settings for color scales, layout options, data transformations, or mappings to initialize configurable axes. Given these input parameters and the input cohort C_i, we define a view V_k as follows

$$V_k(C_i, \vec{\alpha}_k) = C'_i \tag{4}$$

Using the inputs C_i and $\vec{\alpha}_k$, views produce a graphical depiction of C_i and allow users to interactively explore the data. In particular, views provide users with visual mechanisms to select subsets of the population to apply filters. This allows users to interactively refine a cohort, most commonly by removing members m_z from C_i that are no longer of interest with a combination of selections and filters. This use case is illustrated in Figure 2.

For example, a demographic view might provide visualizations of age, gender, and ethnicity distributions for a given cohort. In addition, the view could allow a user to interactively select subgroups for filtering (e.g. selecting only females over the age of 50 years).

However, views are not limited to simple filtering. Views can be designed to support more sophisticated cohort manipulations such as annotation, which would allow users to label selected subsets of patients with additional features (e.g. as in Gotz et al.[49]). In this way, views can be designed not only to narrow the focus of an analysis to members of interest but also to help expand $F^{out}_{C_i}$ based on discoveries made using a visualization.

As with analytics, individual views can specify a prerequisite set of features in $F^{in}_{C_i}$ for the input cohort C_i for the view to function properly. For example, the demographic view described above might require that each member of the input cohort have an age, gender, and ethnicity. The prerequisite feature set for a view V_k is noted $F^{pre}_{V_k}$.

Finally, views produce output cohorts C'_i that reflect the manipulations performed within a view. Views satisfy this requirement by providing an export capability which is used to retrieve the current cohort from a view at any given point in time. To enforce this requirement, the export function is specified as part of CAVA's required application programming interface (API) for view components. Therefore, from a functional perspective, view artifacts are identical to analytic artifacts in that they both take a cohort as input (along with an optional set of input parameters) and produce a new cohort as output. This commonality is central to the CAVA design.

Architecture overview

The CAVA architecture builds directly upon the three design artifacts defined above. As illustrated in Figure 3, two logical databases are used to store the system's data. The population database contains all

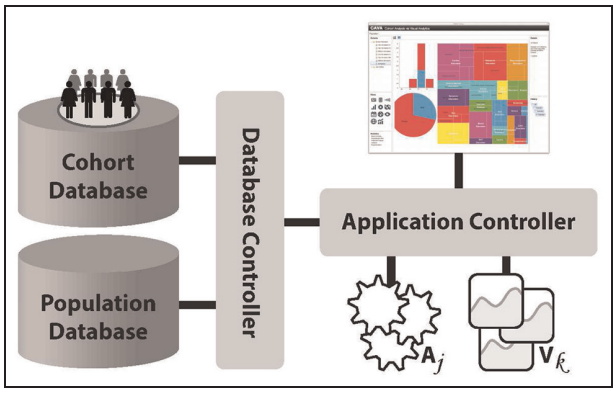

Figure 3. The high-level architecture adopted by CAVA. Two logical databases contain population and cohort data, while a database controller manages access to these resources. The application controller provides the runtime logic to (a) connect the libraries of analytic and view components with data sources and (b) manage user interaction.
CAVA: Cohort Analysis via Visual Analytics.

known information about the individual members of a population. As new data about individuals are derived (via either analytics or views), they are appended to this database.

The second database stores cohort information. Updates are made to the cohort database as new cohorts get defined or as existing cohorts get refined (i.e. changes in membership and/or global properties). For each cohort, CAVA maintains a list of member ID values (used as keys within the database schemas) which allows for the joining of data across the two databases.

Access to these data sources is provided by a database controller. This component manages database connections and provides a standard API to the underlying data. This API is then used by an application controller for all data access and manipulation. This approach allows the vast majority of the system to remain database agnostic, making it simpler to connect to alternative data sources for new applications or deployment environments.

The application controller serves as the central management component within the CAVA architecture. It connects and coordinates the various components of the platform, and it ties these elements to the user interfaces to form a single integrated system. We describe the coordination process in more detail in section "Typical CAVA workflow."

Included within the resource pool managed by the application controller are two libraries of plug-ins, one for analytics and one for views. CAVA defines generic analytic APIs and view APIs that must be implemented by each instance of these components. The APIs include callbacks to manage lifecycle events (e.g. the completion of an analytic process or the rendering of a view through the user interface) and data requirements (e.g. gathering α_j for a given A_j). They also ensure that all operational artifacts adhere to the contracts defined in equations (3) and (4), including the ability to export an output cohort. Because the same APIs are shared across all plug-ins, the central application controller can be defined to work generically, agnostic to the specifics of the underlying visualization or data analysis algorithms deployed within the system.

Configuration files allow the controller to discover which analytic and view components are deployed within the system. Then, at runtime, the controller orchestrates interactions between the deployed components using the generic APIs. Importantly, this approach allows new analysis or visualization components to be deployed dynamically without making any changes to the rest of the CAVA system. In practice, this is an important architectural detail because it allows for use-case specific CAVA plug-ins (e.g. cardiology-centric vs oncology-centric risk assessment analytics) to be deployed in different installations without having multiple builds of the overall platform.

One critical responsibility of the application controller is *cohort binding*. This is the process by which the controller initiates execution for either an analytic or a view by pairing it with an input cohort. During the binding process, the controller checks to ensure that $F^{pre}_{A_j}$ or $F^{pre}_{V_k}$ (for analytics and views, respectively) are subsets of $F^{in}_{C_i}$ for the given input cohort C_i. The controller aborts the binding and return an error if the prerequisite test fails. If the prerequisite test succeeds, then the application controller prompts the user interface to gather any required input parameters α_j (or α_k for views). Once these binding activities have been successfully completed, control is passed to the appropriate analytic or view for execution.

User interface

A common theme to the CAVA requirements outlined in section "Motivating scenario in health-care domain" is ease of use. Users in our motivating scenario must be able to interactively perform both analytics and visualization tasks flexibly, iteratively, and without requiring intervention from outside technology experts. The user interface for CAVA is designed to satisfy these requirements.

The interface consists of six panels as shown in Figure 4. The left sidebar contains three panels, one for each of the CAVA artifact types: cohorts, views, and analytics. The cohort panel displays a list of cohorts available for further study. The list contains both system-generated cohorts and cohorts manually

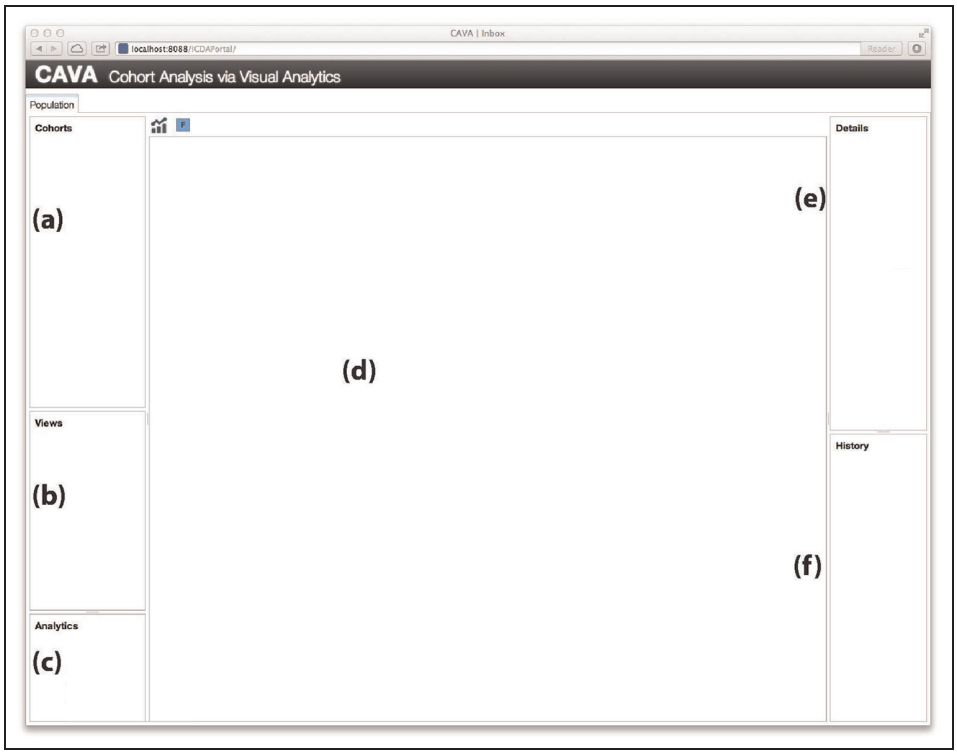

Figure 4. The CAVA user interfaces consist of six panels. The left sidebar contains lists of the three key artifacts in the system: (a) cohorts, (b) views, and (c) analytics. When a user binds a cohort to a specific view via drag-and-drop interaction, the result is displayed on (d) the visualization canvas. The (e) details panel shows more information about the visualized cohort, while (f) the history panel allows users to revisit previous steps.
CAVA: Cohort Analysis via Visual Analytics.

defined and saved by a user. These cohorts serve as starting points for user analysis tasks. The view panel displays several icons, one for each of the views available to visualize a cohort. Similarly, the analytics panel lists the analytic components that can be used to process a cohort. To apply any operational artifact on a specific cohort, a user can simply select the cohort of interest from the cohort panel, drag it to the view or analytic panel, and drop it onto the requested artifact. This simple interaction triggers the binding process described in section "Architecture overview."

To the right of the artifact panel, the visualization panel is the largest and most prominent section of the user interface. Located in the center of the screen, this area is used to display the currently active view and allows users to interact directly with the visual representation of the cohort data. As a minimum, users can select subsets of data and apply filters through this panel. The availability of additional interactive features, such as annotation, depends on the view.

The remainder of the user interface falls to the right of the visualization canvas. It includes a details panel to show additional information about the cohort currently being visualized and a history panel that provides an interactive representation of a user's visual analysis history. The history is organized as a tree with buttons that allow revisitation of prior analysis steps. Mousing over buttons provides additional details about the corresponding step such as applied filter parameters. The granularity of the history maps to the sequence of operational artifacts applied to the initial cohort (i.e. the chain of analytics and views used to manipulate the source cohort).

Typical CAVA workflow

To better illustrate how users typically navigate the CAVA interface, we describe a typical workflow as illustrated in Figure 5. The very first step in a CAVA analysis happens automatically prior to any user interaction. A set of batch analytics—pre-configured as part of the CAVA system's deployment settings—process the entirety of the population database. The result of this process is a set of system-generated cohorts which, along with the default cohort P, can serve as starting points for users' interactive analyses.

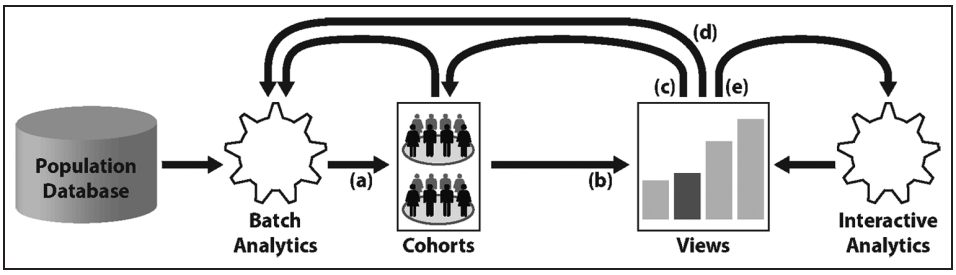

Figure 5. CAVA allows users to intermix analytics and visualization-based cohort manipulations as part of an ad hoc exploratory analysis process. A set of batch analytics process the population data to create (a) an initial set of cohorts. Those cohorts can be (b) bound to views via drag-and-drop interaction. From a view, users can either (c) save modified cohorts, (d) request additional batch analytics that run in the background, or (e) trigger interactive analytics for on-demand processing which automatically update the active view.
CAVA: Cohort Analysis via Visual Analytics.

These cohorts are stored in the cohort database. For instance, the risk stratification analytic mentioned in section "Analytics" could be pre-configured to provide CAVA users with an initial set of high-risk patient cohorts for further investigation.

Following the batch-analytics-based initialization stage, the remaining workflow in CAVA is driven by user interaction. After first logging into the CAVA system, a user can browse the cohort panel to see a list of population groups that are available for visual analysis. At first, these will be the cohorts generated during the initialization round of batch analytics. Over time, the list grows to include manually crafted cohorts created via interaction with the CAVA system.

From here, a user can perform one of two operations, both triggered by drag-and-drop manipulation of an entry in the cohort panel. First, a cohort can be dropped onto an item in the analytics panel. This would perform additional computation on the set of patients and store the results in the user-defined section of the cohort panel. (All analytics are treated as batch analytics when the input cohort is taken from the cohort panel. This is because there is no active view to display the results interactively.) Alternatively, and most common, users can drag a cohort to one of the visualization icons in the view panel. This would automatically bind the cohort to the visualization, which checks the view prerequisites $F^{pre}_{V_k}$ against the cohort's $F^{in}_{C_i}$, queries the database to gather data for the cohort members, and renders the interactive view on the visualization canvas. From a rendered view, users can interactively explore the data in various ways. The exact set of interactive capabilities available to the user (e.g. brushing, pan/zoom, annotation) depends on the specific view that was requested. However, all views allow users to select subsets of a cohort's membership and apply filters.

In addition, users can perform one of three subsequent steps. First, users can save a modified cohort so that it can be revisited at a later time. In response, the newly saved cohort appears in the user-defined section of the cohort panel. Second, users can pivot from one type of visualization to another. This allows users to quickly navigate between multiple visualizations, viewing and refining the cohort throughout the process. Finally, users can request that a new round of analytics be performed on the cohort. In response, CAVA first gathers any needed input parameters that the analytic algorithm might require (typically via a dialog box). It then automatically launches execution of the requested analytic module. For interactive analytics, the results are automatically bound to the currently active view which displays the newly created cohort. For longer running batch analytics, the results are persisted to the cohort database for asynchronous review by the user. Throughout this process, the history panel records the user's analysis process (both views and analytics) which lets the user review his/her past steps and compare cohorts from various stages of the analysis.

Prototype implementation

Based on the design proposed above, we have developed a prototype CAVA implementation targeting the health-care domain. Our prototype connects to a population database containing electronic medical data for a set of cardiac patients. For each patient, the dataset contains both demographic information and longitudinal medical data. The medical portion of the database contains time-stamped records of diagnoses, labs, medications, and procedures. This section describes the prototype CAVA platform implementation details and lists the set of available views and analytics.

CAVA platform implementation details

The CAVA prototype is a web-based system built using Servlet technology which can be hosted using

the open-source Apache Tomcat server or commercial alternatives (i.e. IBM WebSphere). Server-side application logic is developed primarily in Java, with small portions authored using JavaServer Pages (JSP) and SQL. Client-side features have been developed using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), the Dojo toolkit for user interface widgets, and basic content elements. The Scalable Vector Graphics (SVG)-based library D3[50] is used for most of the visualization rendering, though some visualization components use a custom JavaScript library built on top of HTML5's Canvas element.

The data model in CAVA is based on the Universal Feature Model (UFM) proposed in Intelligent Care Delivery Analytics (ICDA),[51] a large-scale batch-oriented health-care analytics platform. The data are stored in a relational database (e.g. IBM's DB2) using a standard data model that is optimized for the sparse and high-dimensional nature of electronic medical data. The UFM data model was designed to be performant for large-scale population analytics and works well for CAVA's typical workload. As a preprocessing step, the source medical data were transformed from their original schemas loaded into our UFM-based population database.

In addition to the data model, the ICDA platform provides a powerful runtime environment for data analytics modules. A plug-in oriented framework allows for the quick deployment of additional analytics, and the flexible APIs allow for modules developed in a variety of languages such as Java and Python. However, as originally proposed, ICDA was designed to support only batch analytics. We have therefore extended the ICDA runtime to support interactive analytics as required by the CAVA application controller shown in Figure 3.

Available views and analytics

Building upon this foundation, we developed a number of views and analytics specifically targeted to the health-care domain. More specifically, the prototype system includes, among others, a demographic overview (showing distributions for age, gender, and diagnoses), a table view to show detailed information about a set of patients, a flow diagram based on the Outflow visualization[27] to show how the symptoms progress along the time, a histogram-based treatment comparison view that uses small multiples to compare patient subgroups, and a radial chart designed to show hierarchical data such as different medical coding systems.

The CAVA prototype also includes a number of analytics. Batch analytics include a demographic module (e.g. to define cohorts for men vs women) and a risk stratification module (to identify various groups of patients at risk of hospitalization based on predictive modeling techniques). Together with the overall population cohort P, the cohorts generated by these analytics provide the initial set of system-generated cohorts from which a user can choose when starting a new analysis. On-demand analytics in our prototype include a patient similarity component,[5] a utilization analysis component,[52] and a heart failure risk assessment component.[10]

The selection of views and analytics described above were chosen to match the initial needs of our target users. As those needs grow, we expect the library of available view, and analytics components will expand as well. Moreover, we note that while the main focus of our prototype is the visual analysis of patient cohorts, a CAVA system can be connected to views of single patients where appropriate. For example, the table view could be extended to allow users to see more information about individual patients or linked to external tools such as traditional EMR systems.

Use cases

The CAVA platform enables a wide range of cohort analysis workflows. To highlight some of the benefits of our approach, we present two CAVA use cases from the health-care domain. Taken together, these use cases show how CAVA supports our motivating scenario and addresses the five requirements identified in section "Motivating scenario in health-care domain." For each use case, we begin with a description of the user's analysis task. We then describe the step-by-step analysis process by which CAVA can help the user complete their investigation. Each use case is illustrated with a figure showing screenshots of CAVA at various stages of the analysis.

Use case: iterative search

In this use case, we follow a clinician who has recently become aware of a new preventive technique that has been shown to help delay or prevent certain types of patients from developing heart disease. (The "new preventive technique" mentioned in this scenario is used to demonstrate a hypothetical scenario for CAVA and is not intended to suggest any novel medical insights. Moreover, this article makes no claims about the existence or efficacy of any treatments—new or old—for any disease. Such claims are beyond the scope of this article and would require rigorous clinical evaluations which are not part of this article.) In particular, the treatment has been studied most in male hypertensive patients between 60 and 80 years of age. Due to limited resources and potential side effects, the clinician wants to focus this new treatment regimen on only those patients who are both (a) at high risk of

developing the disease and (b) best fit the selection criteria for which the treatment is most effective. The clinician uses CAVA to find a cohort of candidates for the treatment following a usage pattern that we call iterative search.

To start, the physician selects a high-risk group from the cohort panel which has been generated by a background risk stratification analytic. The user then drags and drops the cohort onto the demographic overview visualization icon. This results in the visualization shown in Figure 6(a), which displays linked views of age, gender, and diagnosis distributions. The user interactively selects various elements in the visualizations to explore how these three demographic criteria are correlated.

Next, the clinician interacts with the visualization to select and filter the age group in which the treatment has been studied: 60–80 years of age. By selecting the age range in the histogram and clicking the filter button, the user modifies the cohort to exclude those outside the specified range. The clinician then selects the men in the cohort and applies an additional filter. The result is shown in Figure 6(b). As a result of the filters, the initial cohort has been reduced to a group roughly one-third in size. However, the clinician presumes that there are likely additional patients—missing from the current cohort—who are clinically similar to the visualized patients and could benefit from the treatment even if they do not strictly meet the inclusion criteria. Therefore, the clinician decides to search for similar patients by dragging the current cohort from the active view to the Patient Similarity entry in the analytic panel. In response, CAVA binds the visualized cohort to the analytic and presents the user with a dialog box to gather the needed input parameters. In particular, the clinician indicates that she wants to retrieve enough similar patients to double the size of the cohort. After clicking OK, CAVA runs the analytic and updates the visualization with the newly expanded cohort.

The visualization now shows the additional similar patients, but the clinician is still not finished. Because the treatment was designed for patients with hypertension, she selects the hypertension subgroup in the visualization (as shown in Figure 6(c)) and applies one last filter. The clinician has now used a combination of ad hoc filters and analytics to identify an initial set of candidate patients to target with the newly available treatment. Moreover, they have accomplished this without the help of a technology team to write SQL queries, run analytics, or produce reports.

Use case: on-demand analytics

In this use case, a clinician has been told by her medical director that an unusually high number of cardiac patients with hypertension are ending up in the hospital with a specific set of symptoms. Given the increased danger to patients and the expense associated with a hospitalization, the clinician wants to identify a cohort of at-risk patients who would most benefit from a proactive care management plan designed to avoid hospitalization.

The clinician begins by dragging the All Patients cohort to the demographics overview visualization icon. This results in a visualization summarizing age, gender, and diagnosis distribution for the full population of patients. She immediately selects hypertension in the diagnosis treemap and applies a filter to focus on the right subgroup of patients. The result is shown in Figure 7(a).

However, the clinician still needs to focus her analysis on only those most at risk of hospitalization. She therefore drags the modified cohort from the visualization panel to the Hospitalization Risk analytic. This causes CAVA to bind the selected analytic component to the user-defined group of hypertensive patients and initiate the scoring process. The cohort's patients are then each assigned a hospitalization risk score that predicts the likelihood of hospitalization based on each patient's unique medical history. When the on-demand analytic returns, this new feature is appended to the cohort's inner feature set ($F^{in}_{C_i}$). However, because the currently active view (the demographic overview) does not display this attribute, no visible changes appear in the visualization.

To visualize both the newly calculated risk assessments along with variations in symptom progression within the current cohort, the clinician pivots to the Outflow view[27] via the same drag-and-drop interaction used to launch the hospitalization risk analytic. This view, shown in Figure 7(b), illustrates how different symptom progression pathways lead to different hospitalization risk assessments.

The clinician finds the location in the view that represents the set of symptoms reported by the medical director and sees that it does indeed correspond to a high risk of hospitalization as predicted by the data-driven analytic component. However, the clinician also sees that this view does not tell the full story. Several related paths with similarly high-risk scores branch off earlier in time with somewhat different symptom progressions. The clinician therefore selects an earlier branch in the Outflow diagram and applies a filter to obtain this larger subgroup of high-risk patients. Now that this set of at-risk patients has been identified, the clinician pivots again by dragging the cohort to the table view (Figure 7(c)) to get a detailed list of patient names and ID numbers. As in the first use case, the clinician has succeeded without relying on a team of technologists at each step.

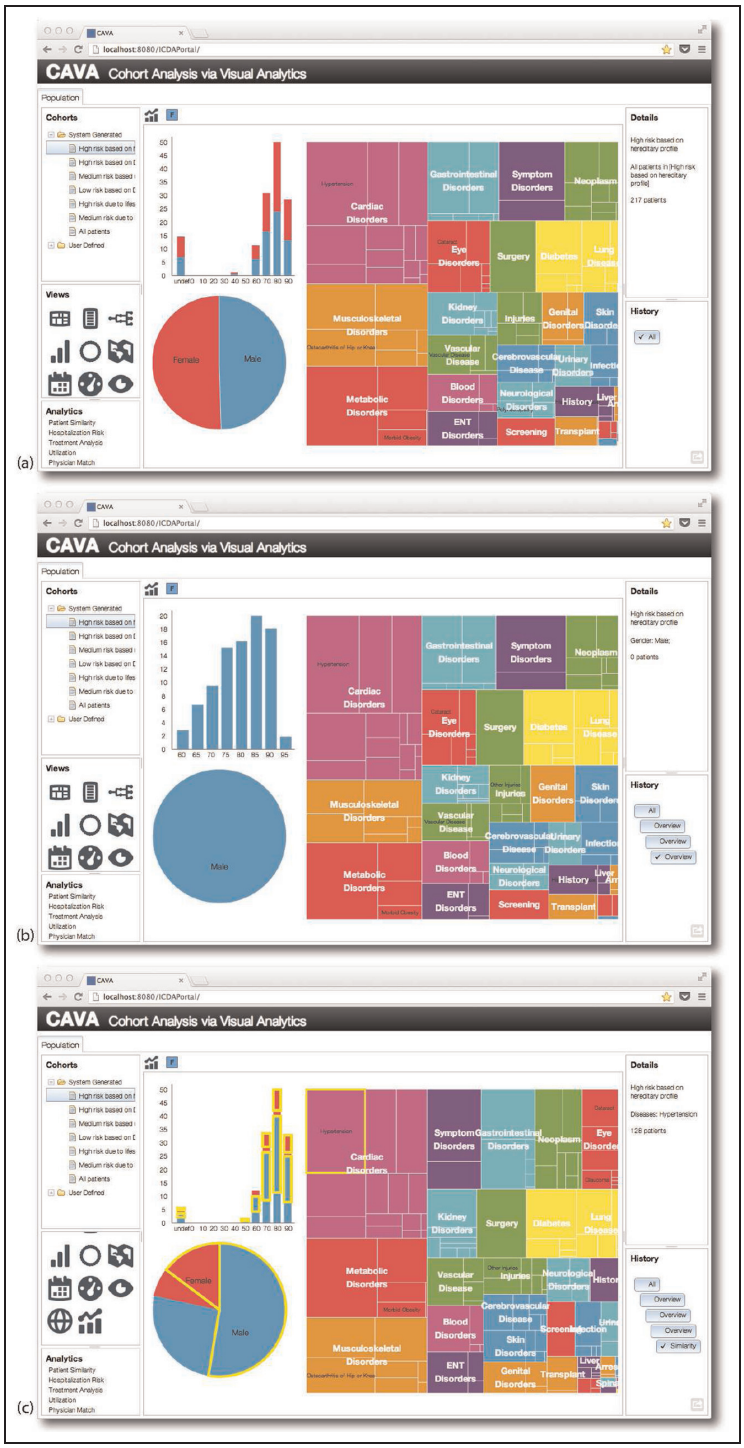

Figure 6. CAVA supports an iterative search process as described in the first use case of section "Use cases." This sequence shows several snapshots from the scenario where a clinician expands and refines an initial high-risk cohort using a mix of visual filters and patient similarity analytics. The end result is a targeted cohort of candidate patients for a new treatment regimen. (a) The sequence begins with a cohort overview showing age, gender, and diagnosis distributions. (b) Interactive visual filters are used to focus the analysis to narrower cohort. (c) Because the filtered group is too small, patient similarity analytics are requested to expand the cohort by retrieving additional clinically similar patients. The newly retrieved patients are visually integrated into the display for further analysis.
CAVA: Cohort Analysis via Visual Analytics.

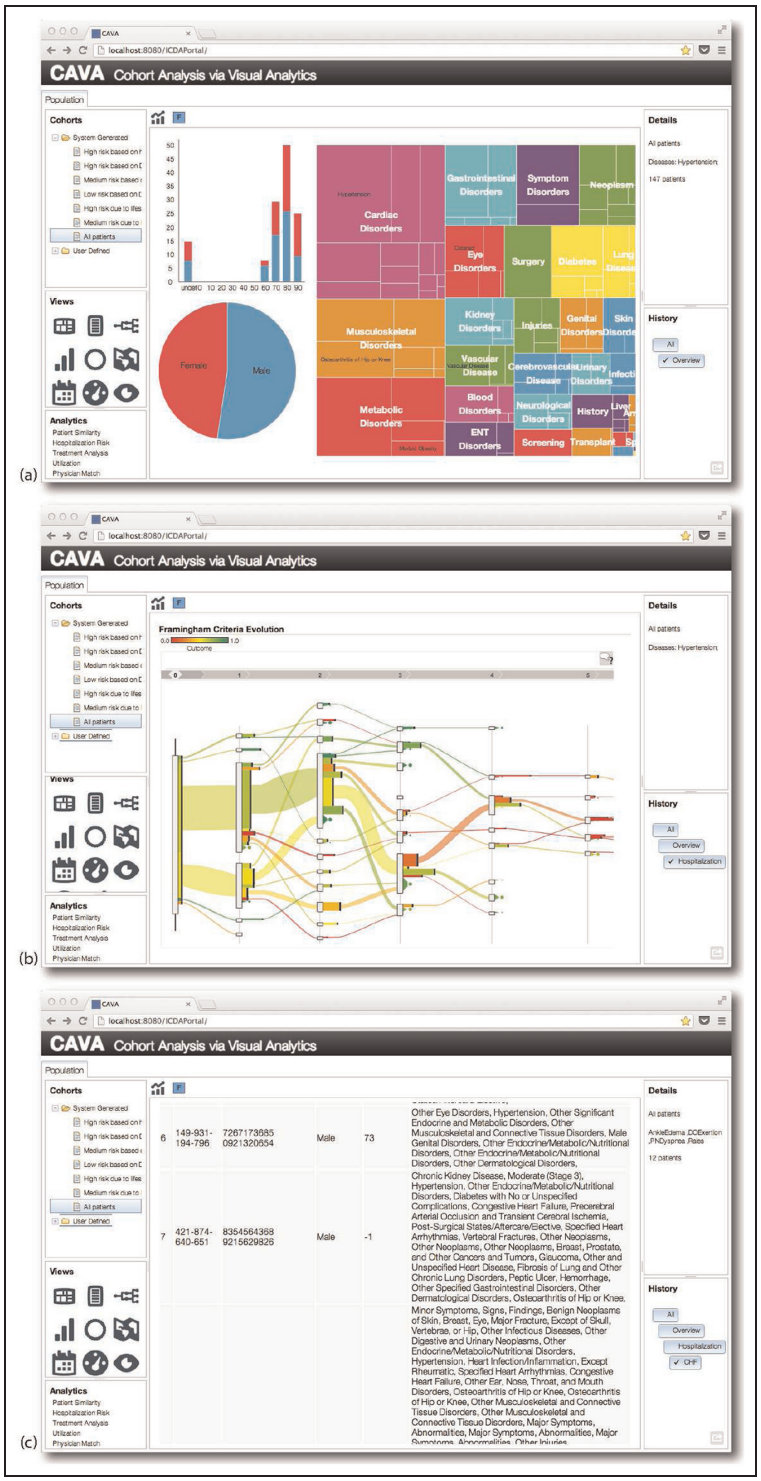

Figure 7. CAVA supports on-demand analytics as described in the second use case of section "Use cases." This set of screenshots shows a user taking advantage of an interactive hospitalization risk analytic to expand the set of available features associated with her cohort. This allows her to filter down to patients who are evolving along high-risk clinical pathways. (a) The sequence begins with a cohort overview showing age, gender, and diagnosis distributions. (b) After applying filters to select a cohort of interest, the user switches to a temporal visualization to identify a high-risk patient pathway. (c) Finally, the user switches to a tabular view to retrieve detailed individual patient information.
CAVA: Cohort Analysis via Visual Analytics.

Domain-expert evaluation

To evaluate our approach, we shared the CAVA system with an emergency room physician with over 15 years of experience practicing medicine and 20 years as a medical executive. His history with both bedside care and health-care management gives him valuable perspective regarding individual patient care and population management.

The doctor spent roughly 1 h reviewing our CAVA prototype applied to a population of over 32,000 cardiac patients. The dataset analyzed with CAVA contains both demographic and longitudinal data extracted from the EMR system of a US-based care provider. The patient data were de-identified to preserve patient privacy, but were otherwise used as stored within the EMR. The data therefore contain real-world distributions and challenges with respect to noise, missing data, and potential bias.

After examining the patient population through the CAVA interface, the doctor was interviewed about his experience. During the interview, he was asked to comment about both (1) the design and usability of the system from the domain expert's perspective and (b) the applicability of the CAVA system to problems faced by major health-care organizations.

Usability and design

Overall, the doctor found CAVA to be "pretty intuitive" and "very useful." He found the ability to manipulate cohorts by dragging and dropping them between analytics and views interchangeably valuable, stating that it provides "speed and validation in a hurry." When asked to provide more detail, he respond by saying that "You can picture someone sitting down over a couple of days with these tools" to complete a detailed set of population studies. He contrasted this to existing workflows using spreadsheets and charting capabilities as currently used, which would require "two weeks of analysis at a minimum" to answer some basic questions. He described it as a much more manual process which would benefit from the ease-of-use and computational power in CAVA. "I think people will be very very impressed at the easy of use factor compared to what they are doing."

One area where he suggested improvements is in the integration of more statistical capabilities within the views. In particular, when discussing the patient similarity analytics and their ability to expand cohorts, he asked "how do you know if there are not enough patients?" This implies that quantitative statistical metrics about a cohort should be available to compliment the graphical views. As currently designed, the CAVA framework can easily support such statistics as integrated pieces of an individual view. However, the general nature of this requirement suggests that building statistical capabilities directly into the representation of a cohort would be an interesting extension to the existing CAVA design.

Applicability to health-care challenges

In terms of the applicability of CAVA to health-care challenges, the doctor was generally very positive. When describing how new medical procedures or practices are discovered, he pointed out that "somebody had the idea that doing things a little differently might help a different group of patients. [CAVA is] a tool to help you figure that out" because it allows you to quickly and easily experiment with analyses on different subgroups.

The primary area where the doctor suggested potential problems for medical use cases was the limited amount of patient detail provided even in CAVA's table view. He suggested that after deriving a cohort of interest, clinicians will need more than just a summary of diagnoses and basic demographics. "All of this makes sense, except for the details" of the patients in the table. "That's useful only up to a point … This is where you tie back to the unstructured data" such as discharge summaries and case summaries. The "narrative unstructured data will interest physicians more than just looking at the table."

In terms of the workflow that CAVA supports, he called it a great match for how things are done in practice. He is planning a study with a collaborator regarding dietary changes in different patient cohorts and said "this would be perfect." Describing how he would use CAVA, he stated,

> Within the initial working cohort, you identify a subcohort of interest but that may be too small. So you go back to the original dataset and find more people like this to fill out the dataset [with similarity analytics] … That's what happens in the real world.

He went on to say that "tools to do that and make it happen like this [snaps his fingers] is very useful … I'd love to show this to [my collaborator], she'll go nuts."

When asked to comment on the benefits of having the ability to run analytics on demand over a given cohort of patients, the doctor recounted a story from his medical training. Patients were arriving at the hospital with meningitis and seizures. "I saw the first person I ever saw die," in part because the "doctors did not recognize what was happening" soon enough. They were looking at data manually, individually, one patient at a time. He felt that when doctors see a trend, a tool like CAVA could let them quickly discover what a group of patients have in common.

Conclusion and future work

This article presented CAVA, a platform for Cohort Analysis via Visual Analytics. CAVA is designed to help domain experts work more independently and more quickly when performing retrospective cohort studies. To motivate our work, we began with a review of a sample scenario from the health-care domain where analysts need to manipulate and explore groups of patients and their associated data to derive insights. Using this example application, we then distilled a set of five important requirements that drove our design decisions when developing the CAVA platform. These include (1) simple cohort definition, (2) flexible visualization, (3) flexible analysis, (4) cohort refinement and expansion, and (5) iterative analysis capabilities.

We then presented the design of the CAVA platform itself. The proposed architecture achieves each of the five identified requirements through a design centered around three primary types of artifacts. Cohorts are the primary data artifact, representing a group of individuals and their associated attributes. Cohorts are then manipulated through two different types of manipulations: analytics and views. CAVA treats both of these types of operational artifacts as equivalent in terms of their abstract functionality. More specifically, both analytics and views process an input cohort and, in response to input parameters and/or user interaction, produce a new output cohort. Given this common formulation, CAVA allows users to chain together arbitrary sequences of visual and analytical cohort manipulations via its drag-and-drop user interaction model. In addition, we described the typical CAVA workflow and presented the details of our initial prototype implementation. Then, we presented two use cases from the health-care domain that show the types of analyses made possible by the CAVA platform. Finally, we conducted a user study to justify the design and usability of our approach. Results showed that our framework met the domain requirements and was helpful to address specific health-care challenges.

While the provided use cases show that CAVA can support a range of analysis tasks, many topics remain for future work. First, a comprehensive user evaluation is essential. We are currently working with domain experts to gather initial feedback on our prototype system and plan to conduct more formal user studies as our prototype evolves. Second, we are working to expand the set of analytics and views available to users. This will help make a wider range of analyses possible using CAVA. Finally, we are looking at ways to expand the functional model we use to represent analytics and views. For example, many cohort study tasks involve the comparison of multiple subgroups. Currently, this is handled via sub-grouping that is performed within a given analytic or view. However, allowing cohorts and views to work with a *set* of cohorts as input and output parameters (rather than the current single cohort formulation) can enable more powerful workflows.

Acknowledgements

The majority of the work described in this article was performed at IBM Thomas J. Watson Research Center in Yorktown Heights, NY, USA. Zhiyuan Zhang participated in a summer internship with the IBM Research Healthcare Analytics Research Group and is an IBM PhD Fellowship recipient. In addition, prior to moving to UNC-Chapel Hill, David Gotz was a Research Staff Member at IBM Research where he was a member of the Healthcare Analytics Research Group.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Glenn ND. *Cohort analysis*, vol. 5. Thousand Oaks, CA: SAGE, 2005.
2. Porta M (ed.). *A dictionary of epidemiology*. 5th ed. New York: Oxford University Press, 2008.
3. Power C and Elliott J. Cohort profile: 1958 British birth cohort (national child development study). *Int J Epidemiol* 2006; 35(1): 34–41.
4. Orthuber W and Sommer T. A searchable patient record database for decision support. *Stud Health Tech Informat* 2009; 150: 584–588.
5. Ebadollahi S, Sun J, Gotz D, et al. Predicting patients trajectory of physiological data using temporal trends in similar patients: a system for near-term prognostics. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, D.C., USA, 13–17 November 2010, vol. 2010, p. 192. Bethesda, MD: American Medical Informatics Association.
6. Chattopadhyay S, Ray P, Chen H, et al. Suicidal risk evaluation using a similarity-based classifier. In: Tang C, Ling C and Zhou X (eds) *Advanced data mining and applications*. Berlin and Heidelberg: Springer, 2008, pp. 51–61.
7. Huxley R, Barzi F and Woodward M. Excess risk of fatal coronary heart disease associated with diabetes in men and women: meta-analysis of 37 prospective cohort studies. *BMJ* 2006; 332(7533): 73–78.
8. Seid M, Varni JW, Segall D, et al. Health-related quality of life as a predictor of pediatric healthcare costs: a two-year prospective cohort analysis. *Health Qual Life Outcome* 2004; 2(1): 48.
9. Varni JW, Burwinkle TM, Seid M, et al. The PedsQL™* 4.0 as a pediatric population health measure: feasibility, reliability, and validity. *Ambul Pediatr* 2003; 3(6): 329–341.

10. Sun J, Hu J, Luo D, et al. Combining knowledge and data driven insights for identifying risk factors using electronic health records. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, Chicago, 3–7 November 2012.
11. Keim DA and Kriegel HP. Visualization techniques for mining large databases: a comparison. *IEEE T Knowl Data En* 1996; 8(6): 923–938.
12. Keim D, Mansmann F, Schneidewind J, et al. Visual analytics: scope and challenges. In: Simoff SJ, Böhlen MH and Mazeika A (eds) *Visual data mining*. Springer Berlin Heidelberg 2008, pp. 76–90.
13. Tatu A, Albuquerque G, Eisemann M, et al. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In: *IEEE conference on visual analytics science and technology (IEEE VAST)*, Atlantic City, NJ, 11–16 October 2009, pp. 59–66. D.C.: IEEE.
14. Keim DA, Kohlhammer J, Ellis G, et al. *Mastering the information age—solving problems with visual analytics*. Goslar: Eurographics Association, 2010.
15. Van Liere R, Mulder JD and Van Wijk JJ. Computational steering. *Future Generat Comput Syst* 1997; 12(5): 441–450.
16. Balzer M, Deussen O and Lewerentz C. Voronoi treemaps for the visualization of software metrics. In: *Proceedings of the 2005: ACM symposium on software visualization*, Saint Louis, Missouri, 14–15 May 2005, pp. 165–172. New York: ACM.
17. Graham M and Kennedy J. Using curves to enhance parallel coordinate visualisations. In: *Proceedings of seventh international conference on information visualization (IV 2003)*, London, UK, 18 July 2003, pp. 10–16. D.C.: IEEE.
18. Inselberg A and Dimsdale B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: *Proceedings of the 1st conference on visualization '90*, San Francisco, 23–26 October 1990, pp. 361–378. DC: IEEE Computer Society Press.
19. Zhang Z, McDonnell KT and Mueller K. A network-based interface for the exploration of high-dimensional data spaces. In: *IEEE Pacific visualization symposium*, Songdo, Korea, 28 February–2 March 2012, pp. 17–24. D.C.: IEEE.
20. Shneiderman B. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans Graph* 1992; 11(1): 92–99.
21. Wood J and Dykes J. Spatially ordered treemaps. *IEEE T Vis Comput Gr* 2008; 14(6): 1348–1355.
22. Burch M, Beck F and Diehl S. Timeline trees: visualizing sequences of transactions in information hierarchies. In: *Proceedings of the working conference on advanced visual interfaces*, Napoli, Italy, 28–30 May 2008, pp. 75–82. New York: ACM.
23. Fails JA, Karlson A, Shahamat L, et al. A visual interface for multivariate temporal data: finding patterns of events across multiple histories. In: *IEEE conference on visual analytics science and technology (IEEE VAST)*, Baltimore, MD, 31 October–2 November 2006, pp. 167–174. D.C.: IEEE.
24. Havre S, Hetzler B and Nowell L. ThemeRiver: visualizing theme changes over time. In: *IEEE symposium on information visualization 2000 (InfoVis 2000)*, Salt Lake City, UT, 9–10 October 2000, pp. 115–123. D.C.: IEEE.
25. Vrotsou K, Johansson J and Cooper M. ActiviTree: interactive visual exploration of sequences in event-based data using graph similarity. *IEEE T Vis Comput Gr* 2009; 15(6): 945–952.
26. Weber M, Alexa M and Müller W. Visualizing time-series on spirals. In: *Proceedings of the IEEE symposium on information visualization*, San Diego, CA, 21–26 October 2001, p. 7. DC: IEEE.
27. Wongsuphasawat K and Gotz D. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE T Vis Comput Gr* 2012; 18(12): 2659–2668.
28. Wongsuphasawat K, Guerra Gómez JA, Plaisant C, et al. LifeFlow: visualizing an overview of event sequences. In: *Proceedings of the 2011 annual conference on human factors in computing systems*, Vancouver, Canada, 7–12 May 2011, pp. 1747–1756. New York: ACM.
29. Zhang Z, Wang B, Ahmed F, et al. The five W's for information visualization with application to healthcare informatics. *IEEE T Vis Comput Gr* 2013; 19(11): 1895–1910.
30. Bruls M, Huizing K and van Wijk J. Squarified treemaps. In: *Data Visualization 2000*, Vienna, Austria, 26–28 May 2000, pp. 33–42. Springer Vienna.
31. Gotz D and Wongsuphasawat K. Interactive intervention analysis. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, Chicago, 3–7 November 2012. Bethesda, MD: American Medical Informatics Association.
32. Perer A and Gotz D. Data-driven exploration of care plans for patients. In: *CHI'13: extended abstracts on human factors in computing systems*, Paris, France, 27 April–02 May 2013, pp. 439–444. New York: ACM.
33. Jankun-Kelly TJ, Ma K and Gertz M. A model and framework for visualization exploration. *IEEE T Vis Comput Gr* 2007; 13(2): 357–369.
34. Perer A and Shneiderman B. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. In: *Proceedings of the 13th international conference on intelligent user interfaces*, Canary Islands, Spain, 13–16 January 2008, pp. 109–118. New York: ACM.
35. Shrinivasan YB and Van Wijk JJ. Supporting the analytical reasoning process in information visualization. In: *Proceedings of the twenty-sixth annual SIGCHI conference on human factors in computing systems*, Florence, Italy, 5–10 April 2008, pp. 1237–1246. New York: ACM.
36. Gotz D and Zhou M. Characterizing users' visual analytic activity for insight provenance. *Inform Visual* 2011; 8(1): 123–130.
37. Rind A, Wang TD, Aigner W, et al. Interactive information visualization to explore and query electronic health

records. *Found Trends Hum–Comp Interact* 2013; 5(3): 207–298.
38. Steenwijk MD, Milles J, van Buchem MA, et al. Integrated visual analysis for heterogeneous datasets in cohort studies. In: *IEEE VisWeek workshop on visual analytics in health care*, Salt Lake City, 24 October 2010.
39. Cao N, Gotz D, Sun J, et al. DICON: interactive visual analysis of multidimensional clusters. *IEEE T Vis Comput Gr* 2011; 17(12): 2581–2590.
40. Gotz D, Sun J, Cao N, et al. Visual cluster analysis in support of clinical decision intelligence. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, D.C., USA, 22–26 October 2011, vol. 2011, p. 481. Bethesda, MD: American Medical Informatics Association.
41. Lins L, Heilbrun M, Freire J, et al. VisCareTrails: visualizing trails in the electronic health record with timed word trees, a pancreas cancer use case. In: *IEEE VisWeek workshop on visual analytics in health care*, Providence, RI, 23 October 2011.
42. Chui KKH, Wenger JB, Cohen SA, et al. Visual analytics for epidemiologists: understanding the interactions between age, time, and disease with multi-panel graphs. *PloS One* 2011; 6(2): e14683.
43. Perer A and Sun J. MatrixFlow: temporal network visual analytics to track symptom evolution during disease progression. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, Chicago, 3–7 November 2012, vol. 2012, p. 716. Bethesda, MD: American Medical Informatics Association.
44. Zhang Z, Gotz D and Perer A. Interactive visual patient cohort analysis. In: *IEEE VisWeek workshop on visual analytics in health care*, Seattle, 14 October 2012.
45. Ferreira N, Lins L, Fink D, et al. BirdVis: visualizing and understanding bird populations. *IEEE T Vis Comput Gr* 2011; 17(12): 2374–2383.
46. Dou W, Wang X, Chang R, et al. ParallelTopics: a probabilistic approach to exploring document collections. In: *IEEE conference on visual analytics science and technology (IEEE VAST)*, Providence, RI, 23–28 October 2011, pp. 231–240. DC: IEEE.
47. Xu W, Esteva M, Jain SD, et al. Analysis of large digital collections with interactive visualization. In: *IEEE conference on visual analytics science and technology (IEEE VAST)*, Providence, RI, 23–28 October 2011, pp. 241–250. DC: IEEE.
48. Center for Disease Control and Prevention. Healthy weight: assessing your weight: BMI: about adult BMI, http://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/ (2011, accessed 24 October 2013).
49. Gotz D, Zhou MX and Aggarwal V. Interactive visual synthesis of analytic knowledge. In: *IEEE conference on visual analytics science and technology (IEEE VAST)*, Baltimore, MD, 31 October–2 November 2006, pp. 51–58. DC: IEEE.
50. Bostock M, Ogievetsky V and Heer J. D3 data-driven documents. *IEEE T Vis Comput Gr* 2011; 17(12): 2301–2309.
51. Gotz D, Stavropoulos H, Sun J, et al. ICDA: a platform for intelligent care delivery analytics. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, Chicago, 3–7 November 2012. Bethesda, MD: American Medical Informatics Association.
52. Hu J, Wang F, Sun J, et al. A healthcare utilization analysis framework for hot spotting and contextual anomaly detection. In: *American Medical Informatics Association (AMIA) annual symposium proceedings*, Chicago, 3–7 November 2012.