MultiViz: Data Visualization of Patient Records

Francisco Manuel Carvalho Neves Maximiano Nunes
Instituto Superior Técnico, Lisboa, Portugal
Email: francisco.m.nunes@tecnico.ulisboa.pt

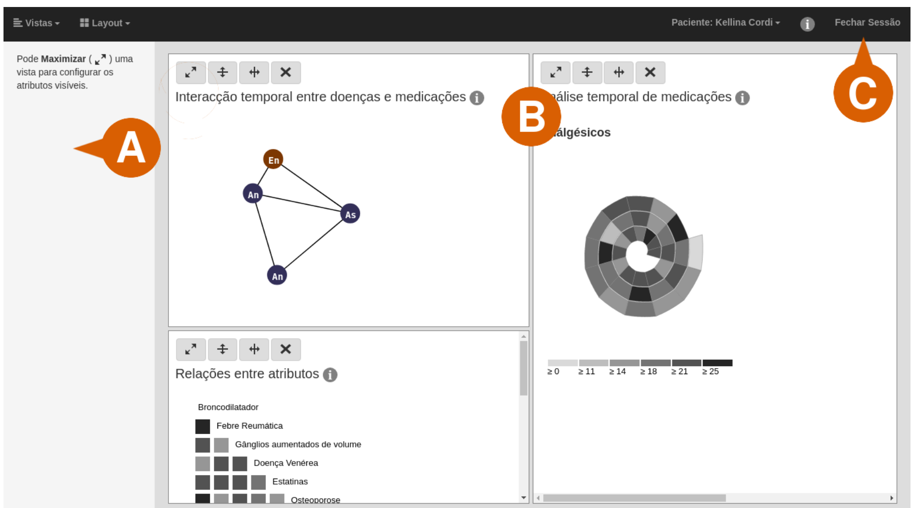


Figure 1: Layout of the interface: Action Panel (A), Panes (B), Main Panel (C).

Abstract

We have developed an interactive visualization tool to improve task efficiency in medical patient record analysis. Large quantities of heterogeneous data need to be processed in order to infer diagnosis and treatments for a given patient. Visualizing this data allows easier creation of medical decisions, by making explicit relevant properties of observed attributes. We have conducted our work in collaboration with Egas Moniz Clinical University (EMCU), developing our tool iteratively with feedback from medical professionals. We attempted to improve discovery and exploration of previously unknown patterns in a given dataset, assisting the university in scaling up their current data system to meet medical professional needs. We discuss the current state of the art in medical visualizations, how we gathered tasks to be addressed by our visualization tool, the architecture of our solution, results from user testing, and suggestions for future work.

Keywords

Information Visualization, Multi-Dimensional, Patient Records, Clinical Diagnosis, Discovery, Exploration

I. Introduction

Collecting patient data is a cheap operation. However, hospitals often struggle to process the vast amounts of data collected. This is due to the unrecognizable structure of stored data [1]. Databases also end up having outdated patient information, forcing clinicians to heavily rely on tedious access to progress notes in order to follow patients' health over time. To approach such complexity, visualizations can be used to support investigation across multiple levels of detail [2], combining human and electronic processing to extract knowledge from a complex set of data [3]. Thus, they can be included in clinical decision-making processes in order to increase situation awareness and maximize utility. Despite these advantages, research has assessed an under-utilization of these graphical representations, comparatively to the use of spreadsheets and notes [4]. Additionally, current practices for sharing these notes is inefficient, leading to inconsistent communications between medical team members. These issues are relevant to consider when analyzing the domain tackled by our project.

We have worked with a group of dentists from the EMCU, who are studying patient cases across multiple pathologies. Several needs were identified during task assessment with these domain experts. Dentists require a comprehensive understanding of the patient's current health state in order to decide on treatment regime. During diagnosis, they will infer this state by looking for relations between dozens of attributes. However, this remains a time consuming process, requiring the dentist to manually check the patient's entire clinical history.

Relations between allergies and medication are used to identify possible incompatibilities, where a certain medication could have negative effects on health due to the presence of an allergy or another medication. Additionally, being able to match medication against diagnosed diseases can highlight situations where the patient is lacking the appropriate medication or is receiving multiple doses for the same affliction. The latter often occurs due to multiple visits to medics at different hospitals or clinics.

From this assessment, we are able to contextualize related work in medical visualizations and develop an understanding of which user tasks we intend to support.

A. *Objectives*

The main goal of our project is to implement an interface that is able to help medical professionals in their tasks. In order to ensure our interface has medical relevance, we have defined the following additional set of goals:

1. Focus on single patient status diagnosis. We have studied different solutions in interactive visualizations applied to medical domains, in order to understand how to enhance current processes and help interpret data.
2. Enhance personalized medicine tasks, where evaluation of a patient status can be done by comparing similar diagnosis from other patients. Clinicians usually consider up to four simultaneous hypothesis [5]. Thus, it is essential for our solution to facilitate discovery and exploration of patterns and relationships in multiple patient record data.
3. Aid a medical professional in making prediction of future evolution of patient status, influencing the current patient's habits. This should be accomplished by enabling the user to look for correlation across multiple patient records. After identifying relevant attributes, it should be possible to figure out similarities in shared attributes and further developments that could happen, thus helping in deciding further tests and medications. An example of such an estimation, in the medical field of Dentistry, would be: "From our dataset of 1000 patients, in 30 years you may lose up to 10 teeth, as it happened to 10% of registered patients sharing the same disease."

B. *Document Structure*

Herein we present related work in Section II, where we discuss the strengths and weaknesses of different approaches to medical visualizations. Afterwards, we present our solution in Section III, showing the evolution of our various prototypes. Here we contemplate identification of user needs, choosing interaction idioms, validating non-functional prototypes and implementation of the functional prototypes. Then, we present an evaluation of our solution's usability in Section IV. We describe the structure of our tests and present the results from user testing with domain experts. Finally, we end with conclusions and propose future work in Section VI.

II. Related Work

During our research, we found that medical visualizations can be classified according to the following approaches: **Temporal**, **Multi-Dimensional** and **Graph-Based**. Where visualizations combine multiple approaches, we chose to classify each by what we consider to be the most representative approach.

A. *Temporal*

Graphical time series can be used to navigate personal history records. Probably the most significant contribution to this approach started with LifeLines [6]. The medical record is summarized as a zoomable set of lines and events. Attributes are grouped in facets (such as problems, diagnosis, medications), each represented by a line. Correlation is perceived by scanning events vertically and horizontally.

LifeLines2 [7] aims to complement query formulations from the previous iteration. A control panel is available for aligning records by a sentinel event, ranking by alphabetical order or number of occurrences, and filtering tasks by presence of a certain sequence of events. This alignment feature aims to reduce the number of context switches that zooming and panning cause. Intervals of validity provide a visual reminder of the possible duration of an event, which eliminates the need to remember and estimate.

Another LifeLines iteration introduced temporal summaries [8], addressing the need to specify temporal range constraints in users' searches, and view multiple records as an aggregate. The implementation consists of stacked bar charts over a time frame, aggregating event counts or record counts. To enhance alignment tasks, this solution allows specification of sequence filters containing both absent events and present events. Selection boxes can be drawn over the temporal summaries in order to filter by temporal range. Four different distributions can be represented by occurrence relative to a sentinel event (i.e. before, after, both before and after, or neither before nor after).

Alternatively, narratives can be extracted out of patient's health data [9]. In this solution, navigation is implemented with selection and zoom of a range of years. Common time periods, such as time since last visit or a week ago, are available for automatic zooming to a predefined section of data. Identification of patterns and relationships (episode-medications-investigations) is eased through user defined or medical rules defined highlights (e.g., selecting an episode highlights symptoms, medication changes, and other related episodes).

We consider temporal approaches to be indispensable in medical visualization, due to the always present temporal nature of typical medical data. However, we find that these approaches alone are not good to communicate thoroughly other types of attributes, because they only enhance detection of temporal patterns. Therefore, other idioms are needed to complement this kind of visualizations in our work.

B. Multi-Dimensional

It is hard to find methods that can encode all desired aspects of a medical setting. This has lead to multiple coordinated views becoming an established paradigm [10]. Generally, these solutions provide views to communicate both an overview and a details-on-demand approach to available data [11]. They resort to mechanisms such as filtering and combing to control the subset of attributes being considered.

One such visualization is TimeSpan [12], which offer a simultaneous overview and detailed information readout of patients by following a set of design goals: Keeping familiarity (through the use of common visualizations such as bar charts and stacked graphs); Representation of multi-dimensional data on demand (with adjustable Bertin-style matrices [13]); One holistic view (heterogeneous data is integrated into stacked bars, allowing comparison of patients and finding outliers). Temporal tasks are addressed in a histogram view, by brushing and linking with selections that overlap.

ForeTell [14] used a different strategy aimed at improving doctor-patient conversation. A user is able to manipulate the view in several ways: *Rollover* displays normal and abnormal ranges for the hovered parameter; *Single click* selects the parameter (with mouse drag for several parameters), used for performing group operations such as filtering, fading unrelated parameters, or collapsing parameters; *Double click* allows edition of values. The doctor can use this information to plan their conversation with the patient, by saving the state and showcasing it as an ideal state of parameters with simulated values. This use case is further addressed with linking parameters that remind the user of attribute dependencies (ex.: cholesterol and exercise).

Another solution [15] aims to display multivariate patterns across a small time scale, divided into hours. It alerts the doctor of some critical patient status without the use of preset thresholds. A star plot idiom is used, for its ability to display multiple variables in a small area. Such idiom allows visualization of averages, values relative to a target value, or parameter specific scales to compare with normal ranges. Filtering is done across time, with highlighting of contained values.

We recognize that multi-dimensional approaches provide good communication, informing on the types of attributes present in datasets. However, we believe they have shortcomings in showing relationships and patterns, due to their overall homogeneous approach to display multi-variate data. According to our user analysis with dental medicine professionals, we have concluded that some attributes are subject to dynamically variable importance. Therefore, we must provide more heterogeneous views to outline different kinds of patterns.

C. Graph-Based

There are solutions that organize data by abstracting multiple databases into force-directed graphs [16], revealing previously unknown clusters. However, they do not take into consideration the multi-dimensional nature of node attributes, often leading to considerable cluttering. To solve these issues, one solution [17] abstracted data into an entity-relation graph, where nodes contain multiple attributes. Associations correspond to links in the graph, which can be filtered. Graph nodes are placed according to user-defined sets of attributes. These reduce the need to know exact search terms and allow a user to view multiple nodes in parallel, avoiding context switches.

A different solution [18] claims that cross-filtering and brushing is the key for data exploration, thus avoiding scrolling and deep selection hierarchies. For patient overview, the user interface provides a hierarchical radial display, together with a body outline, and a time occurrence histogram. A sunburst layout allocates areas with relative placement between adjacent nodes, to reveal relationships in the hierarchy. The root node contains a body outline, such that nodes point to a dot in body locations when relevant. Details-on-demand are obtained by hovering the dot. This system also has a sequential display for diagnosis reasoning. Each node in the internal graph structure corresponds to a column of equally colored boxes, which reduces cognitive load of visual search. Back edges appear when current symptoms are caused by previously prescribed drugs. Each node contains a length bar to encode the quantitative effectiveness of treatment. Scalability is addressed by fading unselected nodes. Aggregations are established according to similarity of the sorting variable. Edge bundling is used to reduce clutter.

DecisionFlow [19] aims to address visualization of large numbers of distinct event types. Similar research focused on aggregate data structures that capture statistics for each event [20, 21]. However, they don't scale to a high number of event types. Users frequently either filter down to a smaller group of event types or collapse multiple low-level types into higher-level categories, resulting in semantical information loss. In order to avoid data analysis with query languages, DecisionFlow opts for user defined constraints, partitioning returned data into sub-sequences. These are composed of preconditions and outcomes. Query results are aggregated and transformed into graphs for visualization. Directed edges between nodes capture the order by which events occurred. Episodes are split into intermediate episodes containing only the events observed within a given pair of conditions. These derived episodes are expressed in the corresponding edges.

PhenoBlocks [22] uses a differential hierarchy comparison algorithm to analyze sets of measurable deviations (phenotypes) between patients. By using existing diagnosis as baselines, PhenoBlocks mitigates errors of probabilistic reasoning. Data is categorized into an ontology hierarchy, represented as a directed acyclic graph where nodes with multiple inheritance are duplicated. The presence of a specific phenotype implies existence of all ancestor phenotypes in the graph. This kind of approach is in accordance with clinical tasks, where more specific phenotypes with higher influence will return fewer results, thus reducing the number of possible syndromes to consider in diagnosis. PhenoBlocks chooses the sunburst space-filling idiom for its layout of nodes, where leaf nodes around the periphery correspond to more granular descriptions of an anomaly.

Overall, graph-based approaches are effective in making explicit which relationships are present in dataset attributes. While some are complemented by temporal or multi-dimensional views, we observe that implementations can offer a low number of separate idioms for encoding data. Also, a graph requires data to be presented as relationships. Thus, it is difficult in our case to directly employ these strategies, since the different patterns contained in our dataset cannot always be mapped as relationships.

D. Discussion

We will now focus on common idioms used by visualizations from each of the presented approaches, specially the operations which the user can do when interacting with a visualization. These abstractions help in defining user goals [23].

Temporal representation of data, inspired by the LifeLines [8] approach, is one of the most common concerns and features prominently among most researched visualizations. This may be due to the fact that analysis of the evolution of patient health state being a need shared by several medical fields. Also worth noting is the heavy use of filters, usually used for query restrictions, alongside user selections. It seems to be a popular strategy to deal with multiple attributes in a more constrained way but remains easy to work with. A few visualizations add the ability to collapse elements. We have also noticed that holistic views tend to be preferred over multiple view strategies in order to reduce context switches.

We conclude that hierarchies generated by graph-based approaches should be complemented with additional idioms, since not all medical data seems to be successfully encoded that way. For example, visualizations that support direct representation of relationships don't necessarily support an overview or temporal manipulations.

With these observations in mind, we designed an interface with different encodings for different attribute subsets, presented through multiform views [24].

III. Designing the Interface

The following subsections describe the evolution of our interface. Our design process followed an incremental and iterative approach, with feedback from EMCU medical professionals in each phase. We began by interviewing EMCU's staff, creating user tasks. These tasks determined which idioms we chose for our visualizations. Implementation of our solution followed an incremental and iterative method through the elaboration of continuously improved prototypes.

A. User Tasks

From our meetings at EMCU, we were able to extract the following insights: Attributes which are more commonly observed include medications, diseases, allergies, and hygiene habits. Temporal patterns can be observed at different time scales. It is relevant to identify absences or multiple co-occurrences of a given attribute type. The most relevant pairings are between diseases and medications, either crossing both categories or crossing a single category. They are usually ordered by frequency of occurrences.

With this information, we defined our user tasks. Tasks are described through *Actions* (user goals related to consumption or production of data) and *Targets* (aspects of data which are relevant for users) [23]. The chosen tasks and questions are the following:

Task 1 *Actions:* Discover, Explore; *Targets:* Dependencies

Question 1 Are there any incompatibilities between allergies and medications associated with a given patient?

Question 2 Are there any complaints or risk habits associated with diagnosed pathologies?

Task 2 *Actions:* Compare; *Targets:* Trends, Outliers
Question 1 Is there a correlation between diagnosed pathologies and prescribed medication among patients?

Task 3 *Actions:* Discover, Explore; *Targets:* Dependencies
Question 1 Is a patient not receiving medication for a given pathology?
Question 2 Is a patient receiving unnecessary multiple medications for a given pathology?

Task 4 *Actions:* Identify, Summarize; *Targets:* Features
Question 1 What is the patient's overall current state?
Question 2 Which are the patient's most relevant attributes?

B. Low-Fidelity Prototype

After defining tasks and questions to support, we proceeded to work on paper prototypes. This kind of prototype is used to evaluate the expected general behavior of the system [25]. Users are able to detect issues with possible events or sequence of events in the new system, allowing us to make design changes in a flexible and cost efficient way.

Validation of these prototypes consisted of testing multiple possible visualizations for our idioms. We wanted to evaluate if users could understand the encodings, in addition to being able to discover and use functionality with relative ease. We collected information from direct observation of user interaction. Our protocol consisted of a user speaking out loud their intentions, while he or she performed tasks from our testing scenarios [25].

In the end, we defined a tiled layout for our visualizations, in order to implement multiform views. We also considered linked highlighting between views, so that a user could complete tasks requiring data subsets with different encodings in a more efficient manner. Thus, this layout should help satisfying **Objective 2**, defined in Section I-A. Four views were designed to address user tasks defined in Section III-A. Their visualizations are later described in Sections III-C4, III-C5, and III-C6. We define idioms and their association with tasks in the following list:

Temporal Analysis Exploration of temporal patterns in patient medications; Supports **tasks 1 and 3**.

Attribute Relationships Comparison of co-occurrences in multiple patient attributes; Supports **task 2**.

Temporal Interactions Summarization of patient status progression across time; Supports **tasks 1, 3 and 4**.

C. Functional Prototype

After initial validation of the low-fidelity prototype, we started the development of a high-fidelity prototype. It was meant to provide interactive visualizations with a working back-end, alongside the intended look and feel.

1) Architecture: We designed an interactive web interface, using technologies such as Hypertext Markup Language 5 (HTML5), Cascading Style Sheets (CSS), and JavaScript. In particular, we used the Model-View-Controller (MVC) software architectural pattern, as provided by the Angular.js framework [26]. Visualizations were built with the D3.js software library [27]. Parsing and computation of dates and time ranges was provided by the Moment.js software library [28]. Data was read from static JavaScript Object Notation (JSON) files, using builtin D3.js functions.

2) Data: Our dataset followed the structure of the Clinical History document used by EMCU. We randomly generated attribute values for approximately one hundred patient records, using scripts made with the Node.js runtime environment [29]. Even though we weren't able to populate our dataset with real patient records, we consider the generated dataset to be adequate for giving an idea of possible patterns that could emerge in the visualizations, such as outliers or correlations. It should also be able to help detecting scalability issues produced by large datasets.

3) Layout: Each element of the interface's layout is described in Fig. 1. These consist of the following: **Action Panel**, which offers a list of available views, data to select, filters to apply, or contextual help for further actions; **Panes**, which correspond to evenly split containers. Each pane contains the visualizations themselves and widgets to set presentation of data. Informational tooltips are available next to the view's title, describing what kind of data can be observed. Buttons for manipulating the view are also present, providing the following functions: maximize/collapse the view, split the view vertically or horizontally, and remove the view; **Main Panel**, where the user can jump between views, create a new layout from scratch, view the current patient's profile, or close the session, in order to choose another patient.

4) Attribute Relationships: To explore relationships between attributes across multiple patient records, we implemented two different types of Co-Occurance Matrix.

The first matrix was inspired by Bananacom's *Similarity Matrix* [30], which in our case combines all possible pairs of attributes across all available categories (Fig. 2). A cell's color saturation encodes the number of patients with a co-occurrence of a given pair of attributes. Since the order of the pair doesn't influence the number of co-occurrences, we are able to avoid drawing half of the matrix. This allows us to position labels to the right of cells, in such a way that vertical and horizontal alignment matches each cell. To aid the user in perceiving these alignments, visual guides are added around cells when the user mouse overs a cell or a label. In addition, the corresponding labels are also highlighted with a filled rectangle. These visual aids are also drawn whenever the user places the mouse over one of the entries in the attribute list.

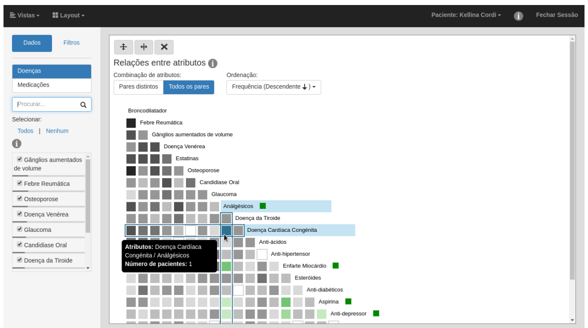


Figure 2: Co-Occurance Matrix, arranged for all pairs of attribute categories.

To compute cell positions (as well as labels), we started by flattening all attribute categories into a single array. Then, we calculated two different indexes, one for each of the two attribute names involved in a given co-occurrence. For the "X" axis, we computed the minimum between two relative indexes in the flattened array, while for the "Y" axis, we used the maximum. Each position was then multiplied by the cell size. Similar computations were done for positioning visual guides.

The cell's tooltip describes the attribute pair involved and the number of patients that share that co-occurrence. It's also positioned in a way that doesn't occlude visual guides. Attributes in the matrix can be sorted alphabetically or by frequency. We added empty bordered cells for missing values, instead of leaving them with white-space. This improves visual alignment of labels when no guides are drawn.

The second matrix focused on showing co-occurrences for distinct attribute categories (Fig. 3). The cell layout was arranged in a diamond shape, which allowed labels to be placed for each dimension without rotations, since they could be visually aligned across each diagonal line of cells. Mouse hovering behaviours are identical to the first matrix.

To compute cell positions (as well as labels), we started by computing the largest possible position a cell of a given category could have. Consider categories "A" and "B", corresponding to the two categories present in a co-occurrence. We fix one of the categories as the baseline (for example, "A"). All atributes of "A" have the absolute position calculated according to the length of array "A" times the cell size. All atributes of "B" use both the lengths of arrays "A" and "B", since the position can now vary across two dimensions. After this absolute position is calculated, we subtract the relative position of a cell, according to the index in the corresponding category array. Similar computations were done for positioning visual guides.

Attributes can be filtered in lists present in the Action Panel. Each entry contains a filled bar, where length encodes the percentage of patients in the dataset which have at least one incidence of that attribute. Each attribute list was ordered by number of incidences, in order to ease identification of common attributes in the dataset.

To inform the user of their filters' impact on affected patients, we made the Patient Distributions visualization, to be located next to the heatmap. To simplify, we only considered intersections of sets between filters. The Action Panel also contains widgets for creating filters (Fig. 3). To save space in the panel and avoid some mouse scrolling, these filters are contained in an accordion widget. The user can choose between biomedical data (manipulated with brushes in histograms) and hygiene habits (manipulated with radio lists, i.e., only one of the entries is set at a time). Histograms show the range of values whenever a brush interval is set. They also contain a green diamond glyph, encoding which bar in the histogram corresponds to the current patient. Radio lists contain filled bars analogous to those described in the attribute lists. Whenever one of these widgets is set, the corresponding distribution bar is added to the Icicle Plot, located to the right of the matrix. The matrix is also updated accordingly, reducing the number of patients in cells to those contained in the filters' set conditions. If one attribute contains no patients with occurrences, it is omitted from the matrix, which automatically collapses any generated empty space. This can be observed in Fig. 3, where the matrix's diamond shape has become much smaller. Filters can be reordered by dragging one of the distribution bars to the new target position. Filled bars for each filter are updated, according to the new hierarchical order.

Our design decisions to include patient distribution bars for attribute lists and filters aimed to reduce selection bias created in user selections. As described in the *Adaptive Contextualization* approach [31], there is a mismatch between the small number of dimensions displayed simultaneously in visualization methods compared to the high-dimensional nature of real-world datasets. Applied filters commonly correspond to a narrow view of the dataset they are manipulating. Their approach tries to reduce bias through a contextualized breadcrumb panel, containing filters that have been applied at each step of the data selection process. It also shows how the resulting dataset compares to previously visualized datasets in terms of underlying variable distributions. In our case, we show the chain of changes with the Patient Distributions visualization.

Context of user selection impact in the original dataset is shown with bars in attribute lists and filters with radio lists, so that the user has an idea of the size of the data subset that is visualized.

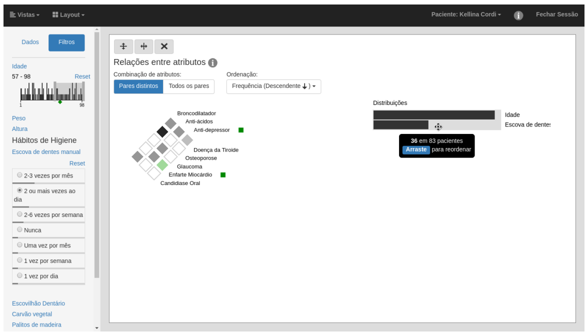


Figure 3: Co-Occurance Matrix, arranged for distinct pairs of attribute categories. The Action Panel contains user set values, while the Patient Distributions Icicle Plot shows the hierarchical application of filters.

5) Temporal Analysis: In order to provide better insight into the temporal nature of the current patient's medications, we chose an idiom that encodes data in a cyclic arrangement. This is accomplished with Spirals (Fig. 4). Each sector corresponds to a bin, delimited by a time interval contained in the corresponding attribute's time range of recorded frequencies. Color saturation encodes the number of dates present in a bin. This number depends on the currently set periodicity. Periods were defined with values that could better show cyclic patterns in a given set of bins, such as 30 for "daily", 7 for "weekly", and 12 for "monthly".

We anticipated cases where recorded dates may be missing. For example, this could point to a patient not taking medication at the expected frequency. If no recorded dates are present in a given bin, it is still drawn but encoded with the color white, similarly to missing values in the Co-Occurance Matrix. A bin's time interval ends when the current date being compared exceeds the last possible date of the interval. Special cases are made for the first and last bins, which are respectively limited by the first and last recorded dates.

To draw the Scalable Vector Graphics (SVG) paths of each spiral sector, we reused some code from *SpiralJS* [32], in particular the computation of the 4 vertices of a sector plus control points for the curvature of edges. We adapted several variables to be dynamically set according to time intervals in our data. For example, a large time range will result in a larger number of sectors in more fine-grained binnings. To ensure reasonable target acquisition, the default binning for a given spiral is the most fine-grained binning that results in less than 150 sectors. Another example is the binning chose by a user, which will influence not only the number of sectors drawn, but also the periodicity used. The overall size of the spiral followed an approximated space-filing algorithm, where sector size was computed based on number of bins and periodicity. Smaller number of bins and smaller periods lead to larger sector sizes.

Regarding time intervals, we also included a line chart next to each spiral, which provides an alternative encoding of the same dataset. It aids the user during brushing of the time range, by providing context about fluctuations in attribute frequencies. The bottom axis contains the start and end dates of the time range, with a few extra dates to provide a better idea of the time interval contained in a brushing operation.

To ease comparisons between two or more attributes, the user can join their corresponding spirals. This function is activated through a button in the source spiral. The Action Panel informs the user to select a target spiral. Candidates are marked with a dashed border around them. After the user selects one of the candidates, the two original spirals are removed from the pane, replaced with a single new spiral. Recorded dates between the two attributes are merged and a new binning is computed. The final spiral will contain the most fine-grained binning interval among both spirals. Tooltips are updated to describe which portion of frequencies in a bin are contributed by which attributes. Sectors with frequencies from two or more attributes are encoded with a different color hue (red).

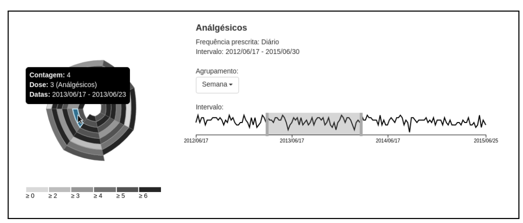


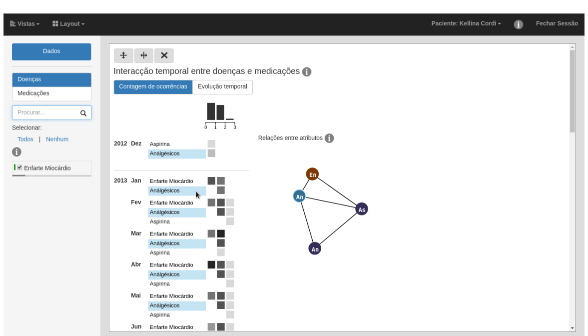
Figure 4: Spiral visualization. User applies brushing to the time interval, reducing the number of sectors displayed

6) Temporal Interactions: This view consists of an overview of the evolution of attributes in the current patient's medical record. We used both Heatmaps and Timelines to encode this data (Fig. 5). Either approach uses small multiples, where segments correspond to months with recorded frequencies in present attributes. When the user places the mouse over one of the labels, all labels matching the same attribute are also highlighted, allowing faster recognition of changes across months.

Heatmaps summarize single or multiple occurrences between attributes in a month. They are aligned with a histogram, displayed on the top. Each bar of the histogram corresponds to the cardinality of simultaneous occurrences among attributes. For example, if two attributes appear isolated in a given month, but also have overlapping occurrences in some dates of that month, this will result in a heatmap with all cells filled, and two bars in the histogram.

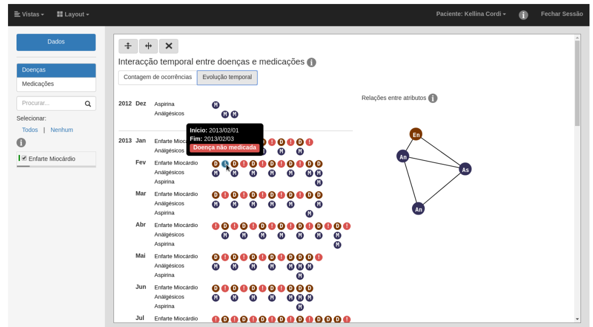
Timelines show time intervals grouped by a tuple of attribute names and cardinality of occurrences among attributes. Each column of points in a month corresponds to a different tuple. Attributes are represented by circular glyphs, labelled with the initial of the corresponding attribute category. Color hue encodes different categories and also warnings for certain cases. For example, diseases that were recorded without any medication being applied are shown in red with an exclamation point. Mouse over a glyph will show a tooltip describing the considered time interval and any dangerous situation, in the case the glyph is encoded with a warning.

Both visualizations are supported by a dependency graph. Nodes correspond to attributes, labelled by their initials, while links correspond to recorded overlapping occurrences between two attributes. The link's length encodes the frequency of these overlaps, where a smaller length corresponds to a higher frequency. We used a force-directed approach, where position of nodes encodes how related attributes are between each other. However, we used only initial strength conditions for positioning nodes, running the force simulation in a single iteration. In practice, this means that link length is proportional to link strength. We can compare this design with the work done in Dust & Magnet [33], which applied a multivariate visualization technique to ease understanding of multi-dimensional data. Similar to a force-directed layout, positioning of points is defined by a certain measure of "attraction". However, this approach differs by using control points for positioning of data points. These control points correspond to each dimension present in the dataset. We make no such distinction in our case, since nodes define the "attraction" among themselves solely based on link strengths.



(a) Heatmap with summarized frequencies

Figure 5: Two types of encodings for attribute occurrences



(b) Timelines for time periods between occurrences

Figure 5: Two types of encodings for attribute occurrences (cont.)

IV. Evaluation

After developing the final iteration of our functional prototype, we submitted it to an evaluation, consisting of usability testing with end-users. These consisted of 17 EMCU dental medicine students. We believe these users are representative of the target professionals for which our solution would be of interest, since they can provide significant feedback on the use cases of our problem domain.

For each question, we collected quantitative metrics, such as time taken to complete tasks and number of errors during task execution. We verified if users were able to correctly accomplish their given questions, which we based on the tasks formulated in Section III-A.

A. Tasks

We came up with two different sets of questions, distinguished by their difficulty level. No mention of their difficulty was present in the question sheet given to users. This allowed us to see if number of errors and time taken by users would increase significantly with the complexity of questions.

The easier questions are the following: **(1)** "Which is the disease with the highest number of patients?" **(2)** "How many patients have *Osteoporosis* and take *Statins*?" **(3)** "Did the current patient take *Anti-Depressants* in July 2 of 2014?" **(4)** "Identify how many diseases the current patient had where some dates had no overlaps with medications." **(5)** "Provide a date where the current patient was poli-medicated (i.e. took more than one medication simultaneously)."

The harder questions are the following: **(6)** "Check if the current patient took *Anti-Depressants* and *Analgesics* in January of 2015." **(7)** "Is there any correlation between patients with a *Weight* higher than 180kg and administration of *Anti-Acids*?" **(8)** "Which month had higher administration of *Analgesics* without any recorded diseases?" **(9)** "The current patient has two attributes with many overlaps. Is this pair of attributes present in most patients?" **(10)** "Which is the pair of attributes with the highest number of patients who use a manual tooth brush two or more times per day?" **(11)** "Apply two filters in this order: *Age* higher than 65 years; *Weight* higher than 160kg. Check how many patients are covered in total." **(12)** "Change the order of the filters applied in the previous question. Explain the change that occurred in the filled bar of *Age*."

B. Questionnaires

Towards the end of the evaluation, we presented two questionnaires: the first consisted of user profiling; the second collected contextual satisfaction criteria, according to the System Usability Scale (SUS) [34]. The former allows us to characterize our observed population in terms of demographics, medical experience and interactive visualization experience. The latter quantifies the perceived usability of our system, focusing on three features: *Effectiveness*, *Efficiency*, and *Satisfaction*.

V. Results

In the following subsections, we analyse how well our users fared when answering the provided questions. We identify the main difficulties they experienced, and provide some statistical assessment of our collected metrics. We also present results from both questionnaires.

A. User answers

Questions where most users failed to answer correctly were **question 11** (14 failed, 1 skipped), **question 5** (13 failed, 1 skipped), **question 7** (11 failed, 2 skipped), and **questions 9 and 10** (7 failed).

We observed many misreadings of Timelines, where users assumed dates between different tuples of occurrences always contained registered occurrence dates for the entire interval. When using the Co-Occurance Matrix, users frequently left filters applied to previous questions, leading to incorrect interpretations of data. Patient distributions in filter bars were rarely understood. In some cases users simply didn't read the questions carefully, providing unrelated answers.

B. Recorded times

Users generally took more time in **questions 3, 4, 7 and 11** (mean time in seconds for each was 99, 112, 110, and 107). **Questions 1, 3, 8 and 11** had the largest deviations between 1st and 3rd quartiles (ranges in seconds for each were [31,101], [82,165], [45,140], [70,185]).

A more in-depth statistical analysis was also conducted. For each question, we had a sample of 17 users. To see which questions followed a normal distribution, we used the *Shapiro-Wilk* test, since our sample was smaller than 50. Given a 5% significance level, **questions 5, 7, 9 and 10** showed evidence for normality ($W_5 = 0.92, W_7 = 0.94, W_9 = 0.95, W_{10} = 0.96$), while the remaining questions showed evidence against normality. Hence, to find significant differences between easier and harder questions, we used a *Paired T-Test* for the first set of questions, and a *Wilcoxon Signed Rank Test* for the second set of questions.

Given $p \leq 0.05$ for both types of tests, we observed that **question 5** was significantly faster than **questions 7 and 9** ($t_{5-7} = 8.40, t_{5-9} = 4.20$), **question 1**, was significantly faster than **questions 6 and 11** ($z_{1-6} = -3.34, z_{1-11} = -2.06$), **question 2** was significantly faster than **questions 8 and 11** ($z_{2-8} = -2.65, z_{2-11} = -3.25$), **question 3** was significantly faster than **question 6** ($z_{3-6} = -3.62$), and **question 4** was significantly faster than **questions 6 and 12** ($z_{4-6} = -2.82, z_{4-12} = -2.81$).

Since only 10 out of 19 tested question pairs were significant, it is inconclusive if the interface was actually able to support harder questions with the same efficiency as easier questions.

C. Recorded errors

Users made a higher number of errors in **questions 1, 7, and 11** (total errors for each were 19, 23, and 24). Common errors across different questions included the following: choosing a view that doesn't have the right visualizations to answer the current question; looking at the current patient's profile when the needed data isn't there; typing unrelated words in the search bar of attribute lists. Widgets in the interface also had limitations regarding precision, leading to off-by-one values.

In an attempt to find a correlation between performance time and number of errors, we computed *Pearson Coefficients*. A strong positive correlation was found for **question 2** ($R = 0.81$) and **question 3** ($R = 0.84$), and a moderate positive correlation for **question 9** ($R = 0.61$). This is consistent with our observations, as **question 2** could be answered by simply checking the tooltip of the correct cell, so any other method that resulted in errors took more time to access. **Question 3** had users jumping to different views and widgets unrelated to the asked question, which adds delays due to view manipulation and context changes.

D. User Debriefing

We encountered considerable difficulties with the learning curve of our interface, due to our fast demonstration and hard to find information and functionality. Few users created split views, leading to wasted time exploring the views because they couldn't recall previous information about them.

Some users felt confused by views presenting simultaneously data from the current patient and data from multiple patients. One user suggested a split view with two panes: one with statistics, the other with the current patient's profile.

Regarding the utility of the iterface, it was generally agreed that the system eased answering some possible hypotheses made by a medical professional, specially when making predictions based on multiple patient data (**Objective 3** defined in Section I-A).

E. User Profiling

All of our users were taking or had a Master's degree in a medical field. There was some variety in the number of years of medical experience these students had. Most users weren't experienced with medical visualization software. Three users had plenty of experience with interfaces for Three Dimensional (3D) models, related to scientific visualization. Therefore, this experience may not be applicable to our interface, which uses interactive charts and figures, related to information visualization.

F. System Usability Scale

The final SUS score was 44.85. According to a report [35] which sets the average SUS score at 68, we can see that our system was ranked below average. This is consistent with user feedback and recorded metrics, which both evidenced considerable difficulties using the interface.

G. Discussion

Our results show mixed success in supporting end-user tasks. While users considered that the interface had utility, there are important aspects that should be revised regarding it's usability.

It is clear that some separation between the current patient data and multiple patient data should be considered. Users have expressed that queries could be made directly on the current patient's attributes. The fact that we mixed different concerns across views lead to unexpected locations containing some given information, causing considerable delays.

Lack of experience with these kind of visualizations may have affected the users' performance, given that they only had a brief demonstration to get used to the provided visualizations.

VI. Conclusion

We have presented an interface for supporting multidimensional data visualization of patient records. Three main approaches were identified in Chapter II regarding visualizations with similar goals. We focused on providing multiform views, in order to offer an interface with the best features of these approaches. Through iterative and incremental development of prototypes, we were able to design some new idioms, aiming to support user tasks identified with EMCU staff. Our interface also provided some integration between visualizations, where changes and other interactions made in one of them were reflected in another.

User testing proved essential for evaluating both the usability and utility of our system. We believe that further design iterations should be considered, in order to solve issues pointed out by users. Nevertheless, we consider that we were able to satisfy most of the objectives established in Section I-A.

A. Future Work

One essential feature to be implemented would be integration with real data. Scalability issues should be addressed with implementation of semantic zoom for attribute categories or *Focus+Context* idioms.

Regarding feedback from user tests, the layout of the interface needs to be adjusted. The patient dropdown should be given a pane of it's own, so that single patient status diagnosis can be done more explicitly. The possibility of including medical rules alongside patient records should also be explored, so that alerts can be provided for more abnormal patient situations. We should try to avoid errors resulting from previously applied filters, possibly with a persistent indicator of the number of patients being visualized. Intervals in Timelines should also be adjusted to map directly to attributes' recorded dates. Brushes should have two input fields for an user to set intervals with better precision. These could also have predefined common intervals suggested to the user. Previews could be added to visualizations to communicate how a certain filter impacts shown data, in isolation from other filters.

Given that our utility assessment was only done during an informal conversation, case studies should be performed after future iterations of the system. These allow us to extract more comprehensive insights from medical professionals on the interface's efficacy and quality in supporting user goals.

References

[1] T. D. Wang, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman, "Visual information seeking in multiple electronic health records: design recommendations and a process model," *the ACM international conference*, no. 1, pp. 46–55, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1882992.1883001$\delimiter"026E30F$npapers3://publication/doi/10.1145/1882992.1883001

[2] T. Munzner and E. Maguire, *Visualization analysis and design*, ser. AK Peters visualization series. Boca Raton, FL: CRC Press, 2015, ch. 1.

[3] D. Keim, G. Andrienko, J.-d. Fekete, G. Carsten, G. Melan, D. Keim, G. Andrienko, J.-d. Fekete, and G. Carsten, "Visual Analytics : Definition , Process and Challenges To cite this version : Visual Analytics : Definition , Process , and Challenges," *Information Visualization - Human-Centered Issues and Perspectives*, pp. 154–175, 2008. [Online]. Available: http://hal-lirmm.ccsd.cnrs.fr/lirmm-00272779

[4] M. Craft, B. Dobrenz, E. Dornbush, M. Hunter, J. Morris, M. Stone, and L. E. Barnes, "An Assessment of Visualization Tools for Patient Monitoring and Medical Decision Making," vol. 00, no. c, pp. 212–217, 2015.

[5] E. U. Weber, U. Böckenholt, D. J. Hilton, and B. Wallace, "Determinants of diagnostic hypothesis generation: effects of information, base rates, and experience." *Journal of experimental psychology. Learning, memory, and cognition*, vol. 19, no. 5, pp. 1151–1164, 1993.

[6] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, "LifeLines: using visualization to enhance navigation and analysis of patient records." *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pp. 76–80, 1998. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/9929185

[7] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, and S. Murphy, "Aligning Temporal Data by Sentinel Events : Discovering Patterns in Electronic Health Records," *CHI 2008 Proceedings · Health and Wellness*, pp. 457–466, 2008.

[8] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith, "Temporal summaries: Supporting temporal categorical searching, aggregation and comparison," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1049–1056, 2009.

[9] V. Nair, M. Kaduskar, P. Bhaskaran, S. Bhaumik, and H. Lee, "Preserving narratives in electronic health records," *Proceedings - 2011 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2011*, pp. 418–421, 2011.

[10] C. North and B. Shneiderman, "Snap-together visualization: can users construct and operate coordinated visualizations?" *International Journal of Human-Computer Studies*, vol. 53, pp. 715–739, 2000. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S107158190090418X

[11] B. Shneiderman, "The eyes have it: a task by data type taxonomy for informatio nvisualizations," *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[12] M. H. Loorak, C. Perin, N. Kamal, M. Hill, and S. Carpendale, "TimeSpan: Using Visualization to Explore Temporal Multi-dimensional Data of Stroke Patients," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 409–418, 2016.

[13] J. M. St Card and B. Shneiderman, "Readings in information visualization," 1999.

[14] P. Bhaskaran, M. Kaduskar, P. Saluja, S. Talimani, S. Bhaumik, and S. Yoo, "ForeTell: Facilitating doctor-patient conversation through interactive information visualization of risk prediction index," *Proceedings - 2012 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2012*, pp. 615–618, 2012.

[15] P. Ordóñez, M. Lombardi, and C. U. Lehmann, "An Animated Multivariate Visualization for Physiological and Clinical Data in the ICU Categories and Subject Descriptors," *Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10)*, pp. 771–779, 2010.

[16] T. M. J. Fruchterman and E. M. Reingold, "Graph Drawing by Force-directed Placement," *Software-Practice and Experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[17] M. D. Lieberman, S. Taheri, H. Guo, F. Mirrashed, I. Yahav, A. Aris, and B. Shneiderman, "Visual exploration across biomedical databases," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 2, pp. 536–550, 2011.

[18] Z. Zhang, B. Wang, F. Ahmed, I. V. Ramakrishnan, R. Zhao, A. Viccellio, and K. Mueller, "The five Ws for information visualization with application to healthcare informatics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 11, pp. 1895–1910, 2013.

[19] D. Gotz and H. Stavropoulos, "DecisionFlow: Visual analytics for high-dimensional temporal event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1783–1792, 2014.

[20] K. Wongsuphasawat, G. John Alexis Guerra, P. Catherine, W. Taowei David, S. Ben, and M. Taieb-Maimon, "LifeFlow: visualizing an overview of event sequences," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems May 7–12, 2011*, pp. 1747–1756, 2011. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979196$\delimiter"026E30F$nhttp://dl.acm.org/citation.cfm?id=1979196

[21] K. Wongsuphasawat and D. Gotz, "Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2659–2668, 2012.

[22] M. Glueck, P. Hamilton, F. Chevalier, S. Breslav, A. Khan, D. Wigdor, and M. Brudno, "PhenoBlocks: Phenotype Comparison Visualiza-

tions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 101–110, 2016.

[23] T. Munzner and E. Maguire, *Visualization analysis and design*, ser. AK Peters visualization series. Boca Raton, FL: CRC Press, 2015, ch. 3.

[24] ——, *Visualization analysis and design*, ser. AK Peters visualization series. Boca Raton, FL: CRC Press, 2015, ch. 12.

[25] M. J. Fonseca, P. Campos, and D. Gonçalves, *Introdução ao Design de Interfaces*. Lisbon, Portugal: FCA Editora, 10 2012, ch. 7.

[26] Google. AngularJS — Superheroic JavaScript MVW Framework. [Online]. Available: https://angularjs.org/

[27] M. Bostock, J. Heer, and V. Ogievetsky. D3.js - Data-Driven Documents. [Online]. Available: https://d3js.org/

[28] JS Foundation. Moment.js — Home. [Online]. Available: https://momentjs.com/

[29] R. Dahl. Node.js. [Online]. Available: https://nodejs.org/en/

[30] Bananacom. Bananacom Demo Results - OPEN — Optimal Workshop. [Online]. Available: https://www.optimalworkshop.com/optimalsort/bananacom/bananacom-demo-results-open/shared-results#/t/analysisTools/similarityMatrix

[31] D. Gotz, S. Sun, and N. Cao, "Adaptive Contextualization: Combating Bias During High-Dimensional Visualization and Data Selection," *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16*, pp. 85–95, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2856767.2856779

[32] W. Tyler. WesTyler/SpiralJS: Create spiral graphs of time-series data using D3.js. [Online]. Available: https://github.com/WesTyler/SpiralJS

[33] J. S. Yi, R. Melton, J. Stasko, and J. a. Jacko, "Dust & Magnet: multivariate information visualization using a magnet metaphor," *Information Visualization*, vol. 00, no. April, pp. 239–256, 2005. [Online]. Available: http://ivi.sagepub.com/lookup/doi/10.1057/palgrave.ivs.9500099

[34] J. Brooke, "SUS - A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996. [Online]. Available: http://hell.meiert.org/core/pdf/sus.pdf

[35] J. Sauro. MeasuringU — Measuring Usability with the System Usability Scale (SUS). [Online]. Available: https://measuringu.com/sus/