Multimedia Tools and Applications (2022) 81:209–236
https://doi.org/10.1007/s11042-021-10990-1

1191: SENTIENT MULTIMEDIA SYSTEMS

Graph-based visualization of sensitive medical data

Ilias Kalamaras[1] · Konstantinos Glykos[1] · Vasilis Megalooikonomou[2] · Konstantinos Votis[1] · Dimitrios Tzovaras[1]

Received: 23 July 2020 / Revised: 15 February 2021 / Accepted: 30 April 2021 /
Published online: 25 June 2021

Abstract

With the increasing amounts of electronic health data being constantly generated in medical examinations and by sensors and mobile applications, data visualization methods can assist medical professionals and researchers in exploring and making sense of the data. Two important challenges faced by data visualization are large data volume and protection of sensitive data. In this paper, we propose a graph-based method that allows the exploration of a patient dataset, while also naturally allowing the summarization of large amounts of data, making it applicable to large datasets and sensitive data. A graph is constructed from the raw data, encoding local similarities among patients, and is visualized on the screen, producing a visual map of the patient distribution. Multidimensional glyphs are put in place of the nodes, revealing the properties that characterize each graph area. The graph construction method is extended to an incremental scheme, allowing federated graph formation. The proposed method is demonstrated in three use cases, regarding frailty in older adults, Sjögren's Syndrome patients, and a large-size diabetes dataset.

Keywords Graph-based visualization · Glyphs · Incremental graph construction

✉ Ilias Kalamaras
kalamar@iti.gr

Konstantinos Glykos
glykos@iti.gr

Vasilis Megalooikonomou
vasilis@ceid.upatras.gr

Konstantinos Votis
kvotis@iti.gr

Dimitrios Tzovaras
dimitrios.tzovaras@iti.gr

[1] Information Technologies Institute, Centre for Research and Technology Hellas,
6th km Harilaou - Thermi, 57001, Thermi - Thessaloniki, Greece

[2] Computer Engineering and Informatics Department, University of Patras,
26504, Patras, Greece

1 Introduction

With the increased availability of large amounts of medical data, methods that allow medical personnel and researchers to explore and make sense of them are necessary. Medical data are increasingly being gathered in Electronic Health Record (EHR) databases, collected either manually by medical personnel, through clinical examinations, laboratory tests, questionnaires, etc., or automatically, through the use, by the patient, of real-time data collection devices (wearable or not), mobile self-management applications, or other types of applications, such as games, that implicitly collect health-related information. The large amounts of collected data pose challenges regarding the presentation of raw data and analysis results to the medical professionals so that they can easily explore them and discern interesting structures, but also regarding the need to protect sensitive information in large multi-cohort analyses.

While the underlying structure of the data can be effectively uncovered by modern statistical modeling and machine learning methods, data visualization still remains an important tool for medical professionals and researchers in understanding the data at hand. Automated methods for patient clustering, summarization and outlier detection can quite effectively compute groups of similar patients with their centroids and show outliers; however, it is with the visualization of a patient distribution or progress through time that this information can be most easily understood by a human and linked to individual patterns or cases that can be further examined. Exploiting the high capacity of the human visual perception, visualization methods can present all individual cases (e.g. patients) at once, leaving the task of pattern detection to the human eye, or facilitate pattern detection after some preprocessing or analysis. However, in critical applications, care should be taken not to reveal sensitive information about individuals, which is especially relevant for multi-cohort studies, where data from multiple patient cohorts need to be examined by researchers who may not be authorized to view individual data from all cohorts. In the machine learning literature, this problem is currently being addressed by *federated learning* methods, a high-level trained model is updated sequentially by each protected dataset, without ever the data of all datasets being processed at the same time or place. Data visualization can borrow from this paradigm to create federated visualization methods which would be advantageous in cases of very large data volumes, where summaries are more appropriate, or in cases of sensitive medical information.

In this paper, a graph-based approach is presented for the visualization of one or multiple patient cohorts, focusing on the visualization of the patient distribution with respect to multiple parameters of interest. A graph, i.e. a network of patients, is constructed from the raw data to encode the similarities among patients and is positioned on the screen using force-directed methods, which consider nodes of the graph as repelling charges and edges between nodes as attractive springs, creating a visual map of the patients that reveals their similarities. The individual characteristics of each patient are presented using multivariate *glyphs*, i.e. small visual representations of multi-dimensional vectors, for each node, which allow the determination of the type of patients occupying the different areas of the graph. The proposed graph-based approach extends naturally to federated scenarios, by maintaining and updating a high-level graph structure based on graph partitioning methods.

An example of the proposed graph-based procedure is displayed in Fig. 1, for a dataset of 400 patients. Figure 1a is the graph of all patients, with each patient being a node and where different groups of patients becoming apparent. In federated usage, what the user would see is the reduced graph of Fig. 1b, where each group of patients is represented by a single node, after graph partitioning. The characteristics of each group are summarized by

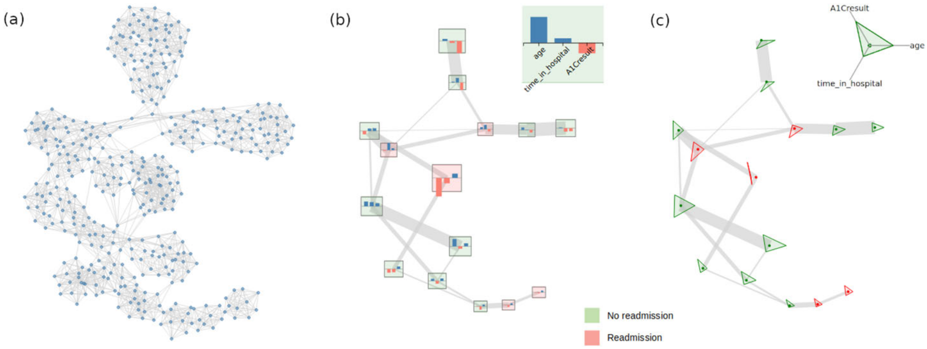


Fig. 1 Example of the proposed graph-based method for a dataset of 400 patients. **(a)** The graph of all patients. **(b)** The reduced graph, using bar chart glyphs. **(c)** The reduced graph, using radar chart glyphs

the node glyphs, here small bar charts. The end-user can select different types of glyphs for the group nodes, according to the clarity of the displayed information. For instance, Fig. 1c shows the same reduced graph, but using radar chart glyphs. The reduced graph can be further combined with additional datasets, to create an incremental representation of data from several cohorts. The applicability of the proposed glyph-enhanced and federated graph visualization is further demonstrated below in three relevant use cases, regarding frailty in older adults, Sjögren's Syndrome and a large-size diabetes dataset.

The rest of this paper is organized as follows. Section 2 reviews related works of the literature regarding graph- and glyph-based visualizations. Section 3 presents the proposed visualization, providing details about the graph construction and positioning, the glyph specification and the federated incremental graph construction procedure. Section 4 demonstrates and discusses the presented method in the context of three relevant use cases, while Section 5 concludes the paper, providing directions for future extensions.

2 Related work

Graph or network layouts have been extensively used in the literature to visualize pairwise relationships among objects of interest, such as social interactions among people, communication among network nodes, gene expression among proteins and disease spread among infected people. Graphs are usually depicted in a two-dimensional layout, using force-directed or other techniques [1], e.g. for social network exploration and community detection [12], although there exist non-planar layouts, such as 3D graph visualizations [13] or spherical graph layouts [15]. Following a different planar layout approach, the BioFabric method [30] displays the graph nodes as horizontal lines and the edges as vertical lines connecting nodes. This allows interesting patterns, such as nodes with similar connectivity, to be easily identified. The recently proposed Parallel Aggregated Ordered Hypergraph (PAOH) [27] method extends the BioFabric method by allowing edges to connect more than one nodes, making it suitable for visualizing hypergraphs (i.e. graphs where each edge connects more than two nodes) and their evolution in time. Another similar extension is DyNetVis [17], where the nodes, each again corresponding to a horizontal line, are restricted to occupy a single column, resulting in a compact representation of the graph. Several such columns are then put next to each other to visualize the evolution of the graph through time.

In the health domain, graph-based data representations have been used to visualize pairwise relationships between patients, patients and medical outcomes, disease spread, etc. For instance, bipartite graphs have been used to visualize cytokines, by representing patients and cytokines with two distinct node groups, while edges exist only between a patient node and a cytokine node, corresponding to a cytokine expression [2]. An extension of this concept is using k-partite graphs, connecting multiple sets of nodes, which have been used to visualize dominant values and common associations between parameters in general-purpose datasets [6]. In a similar manner, a graph-based visualization has been used for visualizing the relationships between Somatic Hypermutation (SHM) associations and Chronic Lymphocytic Leukemia (CLL) patients [23]. Graph visualization has also been used to present the evolution of patients through time [29]. In this case, a node represents a group of patients, while an edge that links two nodes to each other shows a correlation. Positioning the nodes so that the horizontal axis represents time is able to reveal progress through time. Moreover, graph-based visualizations have been used for discovering suspicious activities, such as fraud or waste in the healthcare sector [18]. The multi-objective visualization method [14] constructs a graph by merging multiple Minimum Spanning Trees from multiple modalities of the available data, and positions the nodes using force-directed graph placement algorithms, revealing multimodal similarities among objects of interest. Graph-based visualizations based on multi-objective method have been implemented for visualizing healthcare data [21, 22].

Graphs of pairwise similarities between patients can reveal patient groups of specific characteristics that can be beneficial for clinical predictions. In such cases, each node represents a patient, a group of patients or some other object of interest, while edges connect patients that are similar in some application-specific manner. In addition, visual attributes of the nodes and edges, such as color, size, thickness, etc., can be used to provide additional information, such as gender or connection type. Such a setup has been used, e.g, for prediction of lung cancer risk [20], uncovering subtypes of diabetes [16] and analysis of treatments for nasopharyngeal carcinoma [24].

Methods such as the above represent the graph nodes by simple visual objects, such as circles, while using the color, size or other attributes of these objects to encode additional information, such as a numerical or categorical variable per node. This additional information can be valuable for the viewer, as it shows the distribution of certain attributes across the mapping provided by the graph structure, possibly providing hints and explanations about why particular nodes have been placed at specific positions, or uncovering isomorphisms between the distribution of certain attributes and the graph structure constructed using other attributes. However, using primitive shapes such as circles, rectangles, etc., limit the number of variables that can be encoded in the visual characteristics of the shape, e.g. the color and size of a circle, or the color, width and height of a rectangle. Glyphs instead provide a way to encode multiple attributes in a single visual object.

A *glyph* is a compact visual object consisting of several primitive shapes, such as rectangles, polygons, circles, etc., composed in a structure that can be easily processed by the human eye. The individual properties of the glyph's primitive components can be varied according to multiple data attributes, enabling the visualization of high dimensional data [3, 28]. A wealth of glyph designs have been proposed in the literature and have been categorized in terms of their use of position, color and orientation [10]. Different types of glyphs may be more appropriate for different tasks, e.g. reading values from the visualization vs. performing a visual search over an entire set of glyphs [19]. Glyph-based visualizations have been used in medical applications, especially in 3D imaging, where 3-dimensional glyphs are superimposed on 3D models of human organs to provide additional multi-variate information [25]. The family of Z-Glyphs [4] visualize z-score normalized vectors in linear or circular layouts, which has the effect that "normal" vectors are visualized close to regular shapes, such as straight lines or perfect circles, making outliers stand out as deviations from regularity, similar to the interactive stacked histogram visualization of [5].

3 Graph-based patient visualization

In this paper, we propose a graph-based method for visualizing a patient cohort, targeting the visualization of neighborhoods of similar patients, creating a visual map for the exploration of the available data. The contributions of the proposed method with respect to the state-of-the-art can be summarized in the following:

- The enhancement of graph-based visualizations with node glyphs, which can visually indicate the characteristics of each graph area, making the graph visualization a comprehensive visual map of the entities of interest (here patients). The resulting visual map can be of value to end-users (e.g. clinicians or researchers) since it can facilitate data exploration, understanding of patterns and detection of outliers, in an intuitive manner (i.e. similar to viewing a geographical map).
- The proposition of an incremental graph construction procedure based on an iteration of graph partitioning and merging, that can be used to summarize arbitrarily large graphs by iteratively building from smaller components. The proposed procedure can be used to summarize large clinical datasets stored in federated databases, in use cases where personal data of individual patients of a cohort need to be protected from exposure to end-users of other cohorts.

In this section, the proposed method is presented in detail.

3.1 Graph construction

Graph-based visualizations can be used for visualizing the relationships among entities (e.g. persons), or, in our case, patients. Here, we consider that each node of the graph is a patient, while edges encode pairwise similarities between the patients in terms of the data collected for them. Such a graph can lead to the visual separation of a cohort of patients into sub-groups. Patients who belong to a specific sub-group present some similarities, while patients that do not belong to any sub-group may be outliers.

In order to construct such a similarity graph, we have to compute the differences between patients. Each patient can be considered as a numerical vector

$$x = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m,$$

where x_j is the value of a specific attribute, e.g. blood pressure, or existence of lymphoma. To facilitate computations, categorical variables taking two values, such as lymphoma existence with two categories ("yes" or "no"), are encoded as numerical variables, e.g. assigning the value 1 to "yes", and 0 to "no". Categorical variables with more than two categories can be encoded in an one-hot encoding manner, in order to be processed numerically. A patient can now be considered as a point in an m-dimensional space. Data from all patients in a cohort form a matrix $X \in \mathbb{R}^{n \times m}$, where n is the number of patients and each element x_{ij} is the value of attribute j for patient i.

$$X \in \mathbb{R}^{n \times m}, x_i = (x_{i1}, x_{i2}, \ldots, x_{im}), i \in \{1, \ldots, n\}.$$

In order to prevent attributes of large scales to dominate others, which would affect the subsequent distance computation, we normalize all dimensions using z-score normalization, before proceeding to distance computations.

We can compute the distance of the corresponding high-dimensional points in order to compute the difference between two patients, using an appropriate distance metric (e.g. Euclidean, L_1, cosine) in order to compute the distance. In this work, we have used the L_1 metric which is defined as the sum of the absolute differences between vector elements:

$$d_{L_1}(x_p, x_q) = \sum_{j=1}^{m} |x_{pj} - x_{qj}|.$$

The choice of the distance metric is application-specific and different metrics can lead to different results. The L_1 metric is less sensitive to outlier values and in the considered use cases has shown satisfactory results. However, the presented method can be used with any distance metric between vectors.

Taking into account the distance metric between a pair of patients, we create a pairwise distance matrix D which contains the distances between all pairs of patients:

$$D \in \mathbb{R}^{n \times n}, D_{pq} = d_{L_1}(x_p, x_q).$$

This distance matrix encodes relationships between patients that can be used for the construction of the similarity graphs.

We use two methods for graph construction according to the distance matrix: k-nearest neighbours and Minimum Spanning Trees. In a k-nearest neighbours graph, each node is linked to another node if the second one is among the k nearest neighbours of the first one, based on the distance matrix. Two nodes that are connected to each other represent nearby points in the high-dimensional space.

According to Minimum Spanning Tree (MST) method, the distance matrix is regarded as the weighted adjacency matrix of a complete graph, in which all nodes are connected to each other and the edge weights depend on the corresponding elements of the distance matrix. This complete matrix is then reduced to its Minimum Spanning Tree, which is a tree (i.e. no cycles) that visits all nodes and has the minimum possible total edge weight. Two nodes that are connected in the MST represent nearby points in the high-dimensional space, as in the nearest neighbors method, but here the number of edges is much less, leading to less clutter. If multiple sets of attributes are available per object (e.g. one set of attributes for blood-related measurements and another for activity measurements), then multiple MSTs can be constructed and merged, in order to construct a multimodal graph. In this graph, cycles may exist owing to the fact that its edge set is a concatenation of multiple trees. In this case node proximity depends on multiple sets of attributes, thus semantic groups depending on multiple diverse parameters may be easier to form. The individual MSTs can be visualized separately next to the merged tree, in order to show attribute-specific connectivity.

After the construction of the graph structure encoding the similarities among patients, we can embed the nodes on the two-dimensional plane in order to visualize the graph structure on the screen. We have used a force-directed graph placement method, where nodes are considered as repelling charged particles and edges as attractive springs connecting pairs of nodes. An additional force towards the center of the viewpoint has also been imposed to limit large layouts in a compact region, utilizing as much of the available space as possible. Examples of graph visualizations can be found in Fig. 2a and b, using the k-nearest neighbours and the MST methods, respectively. We can observe that, e.g. in the k-nearest neighbor case, some clusters of nodes have been created. These clusters represent groups

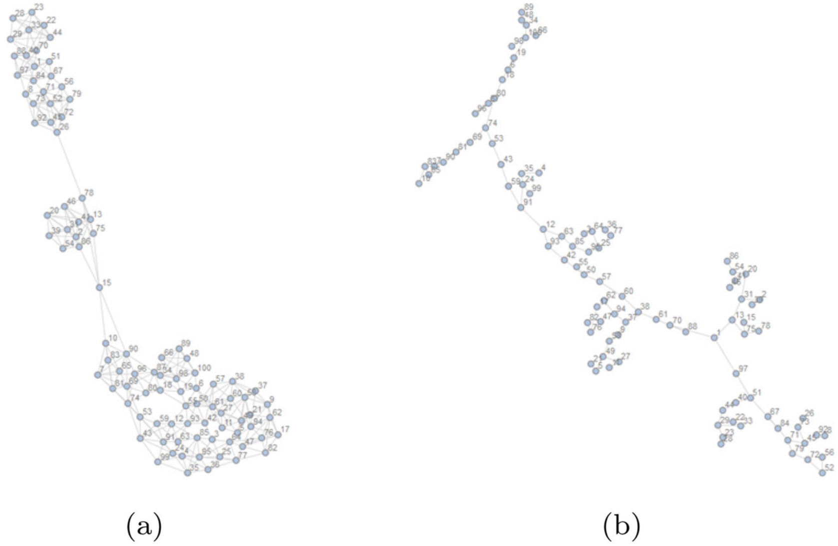


Fig. 2 Graph visualizations of the same dataset. **(a)** Using k-nearest neighbors. **(b)** Using a Minimum Spanning Tree (MST)

of patients that are similar to each other, as demonstrated in Fig. 3, where the data of some representative patients (nodes) are presented in bar charts. The two patients at the top cluster have similar values for the four attributes considered, while the other two patients, in two different clusters, have quite dissimilar vectors. Similar bar charts could be generated for the groups of patients, instead of individual patients, as can be seen further below, in the use cases section.

3.2 Graph node glyphs

The positioning of the graph nodes on the plane according to the above procedure already reveals a lot of information about the set of patients. The user can see areas of patients with similar characteristics, visually spot patient clusters and detect outliers as isolated nodes. However, no information about the kind of patients that each area represents is revealed. To add this information on the chart, we choose to replace each node with a glyph.

A *glyph* is a compact visual representation of a multidimensional item, e.g. a numerical vector, where each dimension in encoded in a different visual element of the glyph, such as the position, size, color, etc. of lines, rectangles, or other shapes. Combining glyphs with a graph-based positioning provides a spatial mapping of the glyphs that allows the user to understand why patients have been positioned in the specific areas, why a node is isolated, etc., by inspecting the visual characteristics of the glyph and its neighbours. It should be noted however that when dealing with graphs with many nodes (e.g. 500 nodes or more), using a glyph for each of the nodes makes the overall visualization quite cluttered and difficult to read. In these cases, it is advised that glyphs are only used for a representative sample of the graph nodes, or only for a summary of the total graph, as is e.g. seen in the use case of Fig. 13.

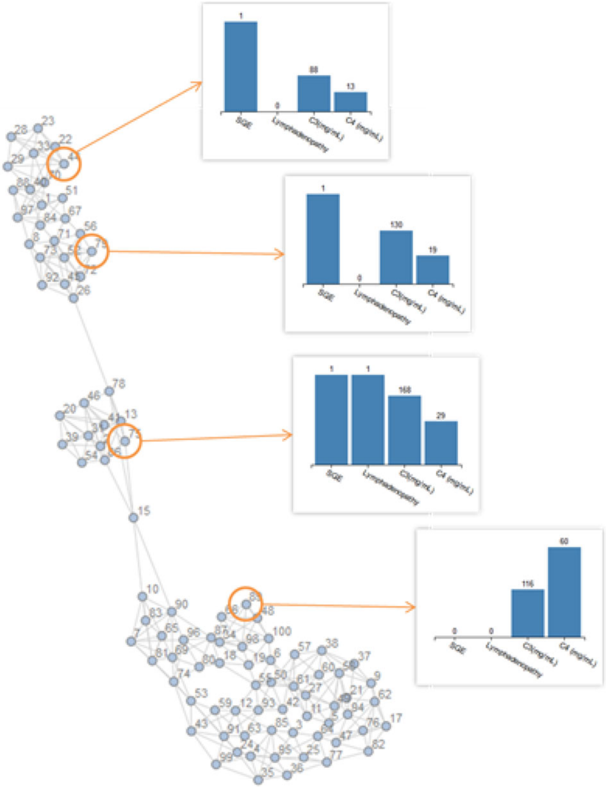


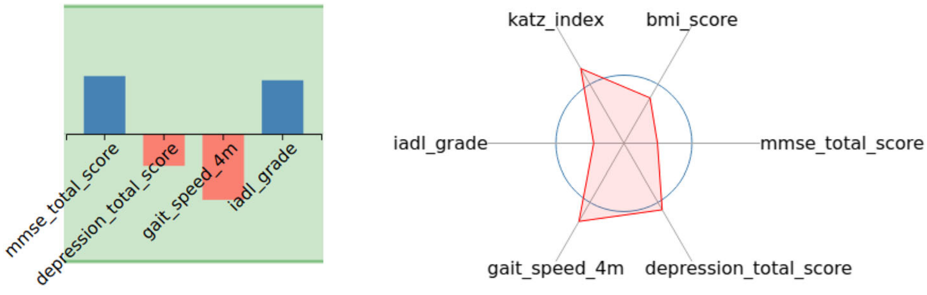
Fig. 3 Nodes that are close to each other correspond to patients that are similar to each other

In the present work, we have considered glyphs based on bar charts and radar plots, in an attempt to see their capacity in terms of compactness and clarity of the visualized information.

3.2.1 Bar chart glyph

A bar chart is a common type of chart that is usually used for displaying the distribution of categorical variables. A bar chart consists of a number of bars, with each bar representing a category. The height of each bar is proportional to the value of the corresponding category. A bar chart can be used as a glyph for each node of the graph, by considering the multiple attributes of a patient as categories of the horizontal axis and the values for each attribute as the height of the bars on the vertical axis.

In Fig. 4a, a bar chart glyph example is presented. For more intuitive representation, the height of the bars encodes the z-score normalized values for each attribute, computed considering all values of each attribute in the dataset. In this way, all attributes are in the same scale, with the height of the bars representing the deviation from the average, in the positive or negative direction. The color of the bars is an additional indication for being higher or lower than the average. A blue bar shows that the value of the patient for the



(a)

(b)

Fig. 4 (**a**) Bar chart glyph with four attributes. (**b**) Radar chart glyph with six attributes

corresponding attribute is greater than the average value of all patients, whereas a red bar means that the value of patient is lower than the average value.

As an additional feature of the glyph, the background color is used to optionally encode another variable, usually a categorical one, which may be of special interest. For example, the color could encode the existence or not of lymphoma, while the bars encode attributes that could be related to lymphoma. The vertical limits of the background rectangle are put at the $+2$ and -2 units of standard deviation, so that very large or very low values compared to the average can be distinguished by the corresponding bars exceeding the limits of the background rectangle.

The color used for the background of the bar glyph or the fill of the radar glyph can encode both categorical variables (e.g. the existence or not of lymphoma) and numerical variables (e.g. average body temperature). In case of categorical variables, color is appropriate if the number of groups is small, i.e. smaller than 10 or maybe 20, since the human perception of color cannot distinguish very similar colors. If, however, the values to be displayed can be ordered (e.g. a hypothetical questionnaire response scale with many degrees from strongly disagree to strongly agree), then a sequential discrete color scale can be used, since the viewer does not have to match exactly a color to its interpretation, but can get a rough estimate by the overall color tone. In its extreme, this would approach a numerical variable, for which a continuous color scale would be appropriate.

3.2.2 Radar chart glyph

The radar chart can be used for visualizing high-dimensional data. Each dimension occupies an axis which may have its own scale. Moreover, each axis starts at the intersection of the axes. The values of the multi-dimensional object in each axis form a polygon whose shape is characteristic of the individual object. The comparison of the polygons allows the user to understand easier the differences between two or more datasets. An example can be found in Fig. 4b. The values of the axes are z-score normalized as in the bar chart glyph. The zero-mean level is indicated by the circle around the center of the chart, i.e. this circle indicates the values of the average person. However, the circle has been omitted in the visualizations of Section 4, since they led to more cluttered displays.

Similar to the background of the bar chart glyph, the color of the polygon's area can be used to encode additional information, such as a categorical variable whose distribution on the graph structure is of interest.

Compared to the bar chart glyph, the radar chart glyph is more appropriate when several attributes are present, as increasing the number of bars in a constrained space makes the chart less readable. However, when few attributes are used, the bar chart may be more appropriate, as it is more familiar to the users. Bar charts are also more appropriate than radar charts in cases where the user needs to read the attribute values from the glyph, since humans can perceive relative position and size better than angular position. The choice of when to use one type of glyph or another can be guided by the above principle, but otherwise it is often the result of trial and error, as one type of glyph may reveal specific patterns that another cannot. In interactive systems, the user should be encouraged to select among different glyph types to see the data from different angles.

As the number of attributes increases, e.g. beyond 10-20, it becomes difficult to discern specific attributes by looking at either the bar or the radar glyph. In such cases, ordering the attributes in some manner can help, since the user does not only rely on the absolute position of an attribute in the glyph, but can approximately locate it in its surroundings. Such an ordering can e.g. be accomplished if the horizontal axis of a bar chart is the ordered bins of a histogram. Regardless of the number of bins used, the histogram will show the approximate distribution of the population represented by the glyph, which makes the bar chart (or the radar chart, by ordering its axes in similar cases) appropriate even for large numbers of bins. This applies also to the color used for the background of the bar glyph and the fill of the radar glyph. Nevertheless, when the attributes cannot be ordered, other types of glyphs may be more appropriate, e.g. ones utilizing other visual properties, such as orientation, transparency, shape patterns, etc.

It should be noted that the two types of glyphs used here are only indicative: different types of glyphs can be used if they are more appropriate for an application. One glyph may be more appropriate when the number of attributes is high, compared to another (e.g. we have chosen the radar chart to show more attributes, because the resulting shape is more compact and clear compared to the bar chart). The choices made for the presented glyphs, e.g. the use of color in the background of the bar chart glyph or for the fill of the radar chart, are meant to demonstrate the possibilities that glyphs can offer. The designer of a glyph can choose to use the available visual characteristics (position, size, color, orientation, etc.) in novel ways in order to achieve clarity and comprehension.

3.3 Incremental graph construction

In the presented graph visualizations, each node represents an individual patient. This allows the visualization of patients that belong to one cohort. However, we may need an extension of the method if we have to visualize patients data that come from different cohorts, since in this case the use, e.g. a clinician, may have access only to data from individual patients of only one of the cohorts. We can implement the graph visualization method in a federated operation mode in order to address this problem. The concept is that only summarized information of a cohort is revealed to other cohorts. Thus, no sensitive information of individual patients is exposed.

The main idea is that a summarized graph can be constructed from data of one cohort, containing only aggregated information about the patient types appearing. This graph can be presented to an unauthorized user with no risk of revealing sensitive information. The user can merge this graph with data of individual patients for which they have access, in order to see how their patients fit within the overall population of patients. Furthermore, the aggregated graph can be updated with the new data, to construct a more accurate representation of the distribution of the whole population. This concept is similar to federated learning in the machine learning literature, where a central model is updated using individual datasets that are never stored at the same place. In our case, the aggregated graph corresponds to the federated model and is incrementally updated with data from multiple patient cohorts. Such an approach can also be used to visualize data of a single cohort that change over time. In this case, the federated model would need to maintain aggregated data from a specified time window, discarding old measurements, in order to track the overall pattern change through time. Such a consideration is not included in the current paper and is left for future work.

A description of the federated graph construction method follows. Figure 5 depicts the whole procedure for an example of two cohorts. The graph of Fig. 5a visualizes data from patients who belong to cohort #1. Initially, each node represents an individual patient. The graph is divided into groups, with the use of a spectral graph partitioning method. Spectral graph partitioning is a graph partitioning method, i.e. one that splits the graph into components so that they are connected between them with as few edges as possible. The problem of graph partitioning is generally hard to solve (NP-hard), so usually approximations or

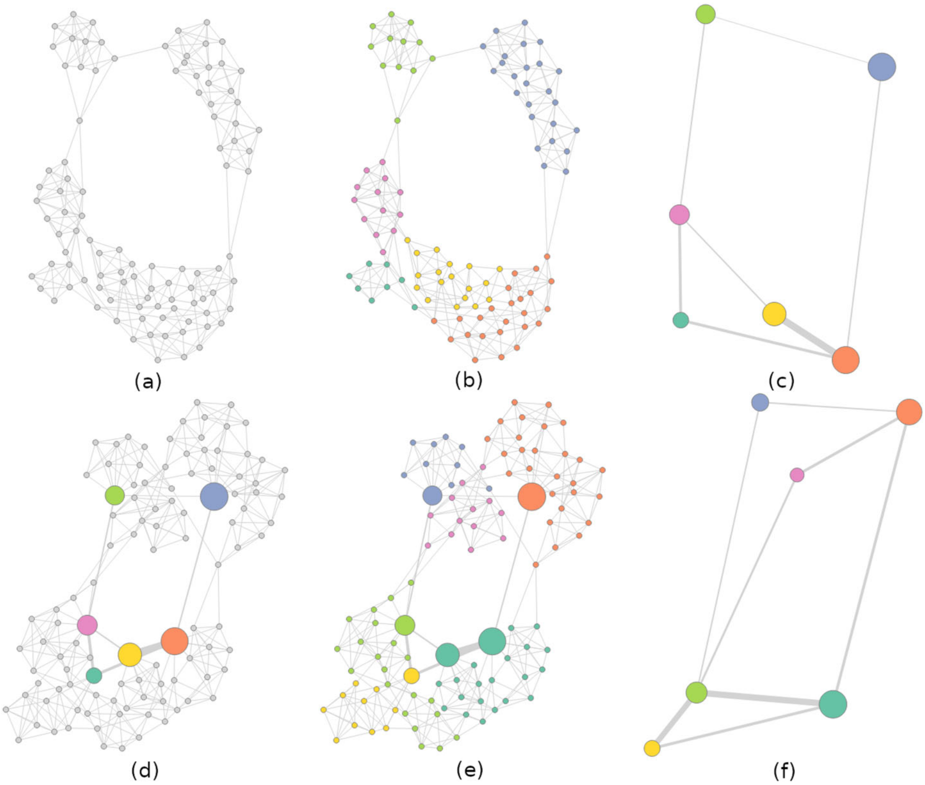


Fig. 5 Constructing a graph incrementally. (**a**)-(**c**) Reduced graph constructed from cohort #1. (**d**)-(**f**) The reduced graph is merged with cohort #2 to construct a total reduced graph

heuristics are used instead. Spectral graph partitioning is a commonly used approximation that achieves sufficient quality for several cases. Spectral graph partitioning splits the graph by first constructing an embedding of its nodes in a vector space, so that nearby vectors correspond to nodes that are connected with several edges. Applying common clustering methods to such an embedding is equivalent to splitting the graph into loosely connected components. Although the resulting partition is not optimal, the constructed embedding follows the structure of the graph so that the split is a good approximation, sufficient for our purposes of graph summarization.

According to spectral graph partitioning, the nodes inside a group are connected to each other with many edges, while there are few edges among different groups. Taking into account the graph adjacency matrix $A \in \{0, 1\}^{n \times n}$, where $a_{ij} = 1$, if nodes i and j are connected with an edge, spectral graph partitioning creates the following graph Laplacian matrix L:

$$L = D_A - A,$$

where D_A is a diagonal matrix in which element d_{ii} is the degree of node i. The degree of the node is the sum of the corresponding row in the adjacency matrix.

The eigenvectors of L with the smallest eigenvalues constitute an embedding of the graph nodes in a space of low dimensionality. Nearby nodes in this space are also close to each other in the graph structure, according to geodesic distances. The points on the space are then clustered with the use of clustering methods such as k-means, dividing the graph into groups. With the use of spectral clustering, the graph of our example is partitioned, as shown in Fig. 5b, where each group has its own color.

Each group consists of nodes that are similar to each other, while nodes which belong to different groups do not have enough similarities. For each group C_k, the following summary statistics are computed:

- The cardinality of the group, which is the number of nodes (patients) in the group.

$$n_k = \sum_{i \in C_k} 1$$

- The group centroid, which is a representative (albeit artificial) patient of the group. This representative patient has the average value of the group in each attribute. Regarding binary attributes with values in {0, 1} a decimal value that belongs in [0,1] is calculated.

$$\overline{x_k} = (\overline{x_{k1}}, \overline{x_{k2}}, \ldots, \overline{x_{km}}), \overline{x_{kj}} = \frac{1}{n_k} \sum_{i \in C_k} x_{ij}$$

- The group standard deviation, which is a vector of the standard deviations of all attributes from the group nodes.

$$s_k = (s_{k1}, s_{k2}, \ldots, s_{km}), s_{kj} = \sqrt{\frac{1}{n_k - 1} \sum_{i \in C_k} \left(x_{ij} - \overline{x_{kj}}\right)^2}$$

This summary does not reveal sensitive information about individual patients data due to the fact that it is cumulative in the group level. Each group can be represented by a single node that has the above collective statistics. This single node that represents the whole group can be merged with the nodes of another graph in an incremental way. Taking into consideration the above statistics, we can say that the nodes of the initial graph may be considered as “groups” that have cardinality 1. In this case, the group centroid is equal to the vector with the patient values, while the vector of standard deviations is the zero vector.

The group centroid is relatively robust to outliers, since if a point is far from the other points in a group, it would probably have been assigned to a different component of the graph partition. However, there may still be cases where the graph partitioning leaves groups that contain outliers (e.g. a point that does not fit to any component, yet is not assigned to a separate component on its own). The existence of such outliers may affect the group centroid so that it no longer is representative of the majority of points. In such cases, the definition of the group centroid above can be modified, e.g. leaving out points that are further than 2 or 3 times the standard deviation from the center (a common straightforward outlier detection technique), or using more advanced outlier detection techniques (e.g. Local Outlier Factor).

To complete the graph construction, we need information about connectivity between groups as well. An edge between two groups exists if there is at least one edge between any nodes of the two groups. One more parameter, the edge multiplicity w_{kl}, is calculated for the edge connecting groups k and l. This parameter counts the number of edges between the group nodes. The $1/2$ factor is necessary owing to the fact that the adjacency matrix A is symmetric.

$$w_{kl} = \frac{1}{2}\sum_{i \in C_k}\sum_{j \in C_l} A_{ij}$$

The multiplicity of an edge that exists between two individual patients is considered as 1.

Taking into account the above notions, the partitioned graph can be transformed into a reduced graph that contains only the group nodes, associated with their summarized statistics, and the group edges with their multiplicities. The reduced graph visualization for the example can be found in Fig. 5c. Each node has the colour of the corresponding group. The size of each node depends on the cardinality of each group, while the thickness of the edges depends on the multiplicity of the reduced edges. The length of the edges, resulting from the force-directed layout, implicitly denotes the amount of similarity between two nodes: nodes that are connected with long edges are put far apart from each other due to their low similarity. It should also be noted that in Fig. 5c and f, the forces of the force-directed layout have been suppressed, to keep the group nodes in roughly the same positions as their corresponding graphs in their previous screenshots, for ease of reference.

The reduced graph provides collective information about the cohort. In this example, it consists of 6 nodes, meaning that the cohort can be divided into 6 groups or sub-cohorts. Each group is represented by a group centroid. The similarities among them are represented by the graph edges. The reduced graph introduces, of course, a loss of information. Not all information of the original data is preserved in the reduced graph, which is, as already mentioned, beneficial for de-identifying individual patients. The amount of information loss can be controlled by modifying the number of components in which to split the original graph. A larger number of components leads to a fine-grained graph containing several nodes, each representing a small group of patients. In principle, this is similar to data quantization and compression methods (e.g. Self-Organizing Maps), where the original data are reduced in size, without losing much information about the overall structure of the data. On the other hand, a small number of components leads to a coarse graph with only a few nodes, each representing a large group of patients. This is similar in principle to clustering methods, where the goal is to split the data to a handful of groups that characterize large areas of the input space. Allowing the user to choose the reduced graph granularity is beneficial in understanding the structure of the available data.

The reduced graph of one cohort is ready to be merged with the nodes of another cohort, without exposing any sensitive patient data. The merged graph consists of the individual patient nodes of the new cohort and the group nodes of the previous cohort. For the calculation of the edges, we take into account both individual and grouped nodes and we consider them as patient vectors. We use the centroid vectors for the group nodes. A distance matrix is calculated, followed by the calculation of the graph edges. Any edges between the group nodes are replaced by the edges of the reduced graph, to maintain the multiplicity information.

The merged graph resulting from this procedure can be seen in Fig. 5d. The gray nodes represent the individual patients of the new cohort, while the coloured nodes are the group nodes of Fig. 5c. The merged graph presents a comparison between the new patient nodes and the group nodes. Some new nodes present similarities with the group nodes of cohort #1. These nodes are located near the corresponding group nodes to which they are similar. Moreover, some new nodes are located far away from all group nodes, which shows that the corresponding patients are not similar to the groups of cohort #1.

The merged graph can itself be partitioned, taking into account all types of nodes (both individual and group nodes). Figure 5e depicts the result of the partitioning. Note that the colors of the nodes do not correspond to the colors of the previous partitioning (i.e. the one in Fig. 5b-d. This happens because a new partitioning is generated, using only the structure of the graph at stage (d). The algorithm does not try to match new groups with old ones, since there is no one-to-one correspondence: new groups may be created that did not exist, or groups may be merged into larger groups. In the example of Fig. 5e, we can observe that most nodes that were close to group nodes belong to the same partition. However, there is a case in which two group nodes belong now to the same partition (blue nodes), as well as a case in which groups of individual nodes of cohort #2 have created their own partition (pink nodes). The explanation is that the additional data cause the update of the distribution of nodes in the partitions. Finally, this partitioned graph can be itself reduced, in case it has to be merged with other cohorts. This final reduced graph visualization can be seen in Fig. 5f.

3.4 Implementation details

The implementation of the above incremental graph construction method in a federated health data analysis platform (e.g. in the platform of the HarmonicSS project) can be performed in the following workflow, depicted in Fig. 6. The user requests that a collective graph of the patients of n cohorts, $C_1, \ldots, C_n$, is constructed and presented. The algorithm starts at cohort C_1, running on the local computing space and database of this cohort and

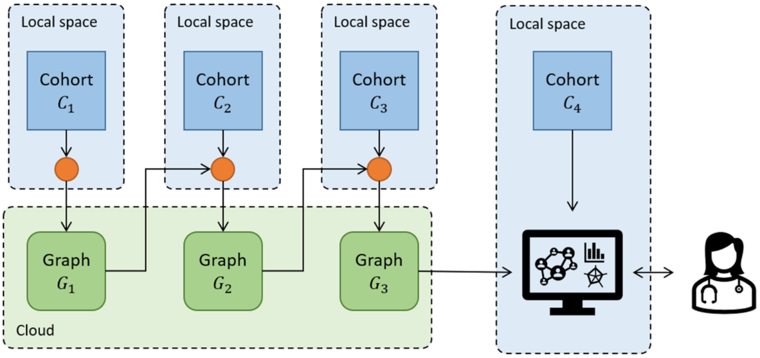


Fig. 6 Workflow of the proposed method in a federated setting, as part of a clinical platform

constructs a reduced graph G_1. The graph G_1 is stored in a cloud repository, and since it is reduced, no data about individual patients are exposed. The algorithm then moves on to cohort C_2, running on its local database, using graph G_1 as a starting point and building a patient graph around it, and producing graph G_2 by merging the summarized information of G_1 with the information of C_2. The procedure is repeated with the algorithm moving from cohort to cohort, gradually building a cumulative graph containing all information available in the cohorts. The resulting graph G is returned to the user, at which point it may be inspected as it is or merged with per-patient information of the cohort for which the user has full access to. The user can guide the procedure by specifying parameters such as the number of nearest neighbors for graph construction, the granularity of the partitioning (i.e. the number of groups to partition the graph into), etc. The user can also select among the available types of glyphs for the final graph nodes, in order to inspect the characteristics of the displayed patients. At no point throughout the whole procedure is information about individual patients of a cohort exposed to the premises of other cohorts, ensuring confidentiality. The only information stored in the cloud, i.e. available to all cohorts, is the reduced summary graphs.

Regarding the technologies used, the proposed method has been implemented as a web application that can be imported in relevant health platforms. The implementation consists of two main parts:

- Back-end graph construction and incremental update: The construction of the graph from the raw data, either in k-nearest neighbours or MST formats, has been implemented as a set of web services in a Node.js[1] server, written in JavaScript and Python. The web services expose a RESTful JSON-based interface, where a user or application can call the service providing the data to be analyzed and a set of options required by each method, e.g. the number of components of the graph partitioning.
- Front-end graph and glyph visualizations: The force-directed graph positioning and visualization, as well as the glyphs for the nodes have been implemented in D3[2]. D3 is a JavaScript library for generating data-driven documents and is commonly used in conjunction with SVG graphics to create custom interactive visualizations. The flexibility provided by D3 facilitated the creation of the glyph-based visualizations, with easily interchangeable (and extensible) types of glyphs.

The web application will be released as part of the data analytics platform of the HarmonicSS project [8].

The current presentation of the proposed methods focuses on their functional components of graph partitioning, incremental graph construction and glyph-based visualization. The interactive usage of these components by the end-user in a final product will enhance the data exploration value of the proposed method in the context of a clinical platform, towards the discovery of groups of similar patients and relationships in their characteristics. In a typical course of work of an analyst (researcher or decision maker), the user would interact with the visualization by specifying the cohorts to combine, the attributes to use for graph construction, the properties of the graph construction procedure (MST or k-nearest neighbors, number of nearest neighbors), the granularity of graph partitioning, the type of glyphs to use and the attributes to show on the glyphs. Furthermore, the usage of widely used web libraries in the implementation (D3) allows the easy integration of the proposed methods in

[1] https://nodejs.org/

[2] https://d3js.org/

a clinical or visual analytics platform, and the implementation of richer means of interaction, e.g. providing further details for selected patients or groups of patients, brushing and linking selections on the graph with other types of visualizations, etc.

4 Use cases

The developed methods have been evaluated in the context of three use cases. The first two deal with the visualization of older adults in relation to frailty status, and visualization of Sjögren's Syndrome patients. The goal in both use cases is to explore the distribution of persons within the data, to understand the different types of users present and detect interesting patterns and outliers. The third use case deals with diabetic patients, while the focus is on how the proposed method can handle a larger dataset.

4.1 Visualization of older adults

The first use case considered deals with the visualization of older adults with regard to their frailty status. Frailty is a condition of reduced functioning present usually in older adults that is related to the physiological and cognitive status of a person.

The data used in this use case have been collected during the course of the European project FrailSafe [7, 31]. The dataset consists of 200 persons, each described by a multitude of attributes related to physiological, cognitive, activity, psychological, etc., characteristics. The data were collected using wearable and ambient sensors, as well as through administered questionnaires. Not all persons have information for all attributes. For the purposes of the current demonstration, we have focused on activity and cognitive characteristics, for which almost all persons have values, collected using wearable sensors and questionnaires, and how they relate to the frailty status of the individual. In particular, the following attributes have been used:

- **MMSE score**: The Mini-Mental State Examination (MMSE) score, evaluating cognitive functionality.
- **Depression score**: The score achieved in a depression questionnaire.
- **Gait speed**: The measured gait speed at a short walking distance.
- **IADL grade**: The Instrumental Activity of Daily Living (IADL) score, collected trhough questionnaires.
- **Katz**: The Katz Index of Independence in Activities of Daily Living, collected through questionnaires.
- **BMI**: The Body Mass Index (BMI), measured using scales.
- **Frailty status**: The frailty status of the individual, measured using the Fried scale [9]. The attribute takes three values: “Non-frail”, “Pre-frail”, “Frail”.

Not all parameters are used in all the visualizations that follow. In each visualization, the parameters used are shown in the legend.

Figure 7 depicts the visualization of all persons of the FrailSafe dataset (excluding cases with missing data for the attributes of interest), using the proposed graph-based method. Each graph node corresponds to an older adult. The connections among them and their positions are computed based on the method described in Section 3, using the attributes that appear at the legend, having numerical values. The number of nearest neighbors for the graph construction has been experimentally set to 5, in an attempt to achieve a compromise between a disconnected graph and a very compact graph, both of which would fail to show

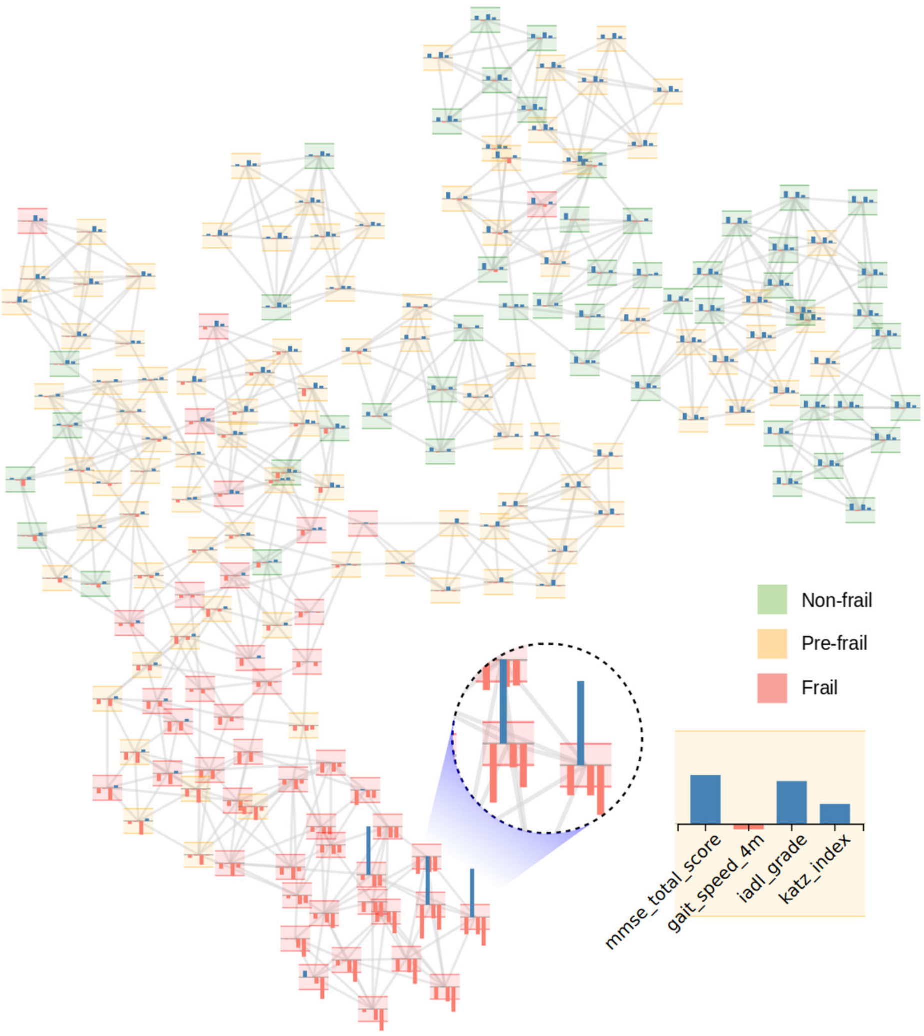


Fig. 7 Visualization of the FrailSafe dataset, using bar chart glyphs

much structure in the data. The above attributes have also been used to create the glyph for each node. The categorical "frailty status" attribute has been used to color the background of each glyph according to the value for each user: green for non-frail, orange for pre-frail and red for frail individuals.

The overall structure of the graph reflects the distribution of the older adults with respect to the considered attributes. The far right area contains mostly non-frail users, who, from the node glyphs, appear to have large values in cognitive-related scores and in the activity-related index. Moving to the left, the middle part of the graph is mostly occupied by pre-frail individuals, with rather average values for all attributes. Various substructures are apparent in this area, such as the cluster at the very top, consisting of persons with relatively high cognitive and activity scores, as reflected by their glyphs. The bottom left part of the graph is covered mostly by frail individuals, with lower than average cognitive and activity scores. This overview might indicate to an end-user that persons with low cognitive ability and low activity are probably indicators of frailty, and could lead to further investigation of such a relationship, to see if it is of statistical significance. Or it might indicate that a person of interest, for whom there is no frailty status available, is positioned in a place where nearby patients are frail, so there is a higher probability that this person might suffer from frailty as well.

Three cases stand out at the bottom (at the zoomed area) with their unusual glyphs showing very high values for the gait-related attributes. Although not spatially separated from the rest of the graph, the glyphs of these cases denote that they are quite different from their nearest neighbors. The very large gait values, significantly larger than two standard deviations, might be attributed to malfunction of the sensors, or some other cause. Other cases of interest might include frail adults appearing among many non-frail individuals, as in the top part of the graph. In all cases, the viewer of the visualization can be alerted from their presence, to investigate them further.

Figure 8 shows a visualization of the FrailSafe dataset using radar glyphs. Radar glyphs are most appropriate when there are more attributes, so we have considered two additional attributes: Katz index and BMI score. We have used the MST graph construction method in this case. The different areas of the graph have different characteristic shapes for the radar glyphs, which reflect the profiles of the corresponding patients. We can see, for example,

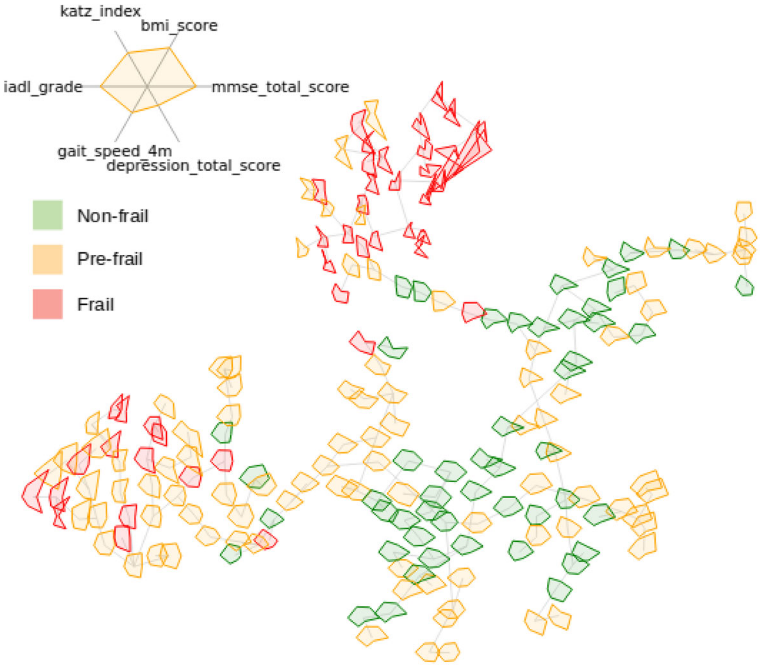


Fig. 8 Visualization of the FrailSafe dataset, using radar chart glyphs and the MST graph construction method

that the left and top parts of the graph, which are mostly occupied by frail people (red color) tend to have shapes that divert from the average shape, with low values for MMSE score (left part of the graph) and low values for IADL grade (top part of the graph).

Figure 9 shows an example of a federated graph visualization using the incremental graph construction presented in Section 3.3. The available data have been split in two cohorts. The first set is visualized using a k-nearest neighbors graph, in Fig. 9a. This graph is partitioned and reduced to the graph of Fig. 9b. Each node corresponds to a group of patients, and the corresponding glyphs describe the average values of each group. The size of the glyphs is determined by the number of patients in each group, while the edge thickness is determined by the number of edges between the corresponding patient groups. The colors for the glyphs represent the most frequent value for the fried score variable in each group. The group nodes correspond to the different areas in the graph of individual patients, in Fig. 9a. The reduced graph does not include information for any individual patient, but only aggregated information. This federated graph can be merged with the data of another cohort, as depicted in Fig. 9c. The group nodes appear in a gray border. The individual patients of the new cohort are mostly positioned near the existing group nodes, denoting that the new cohort patients

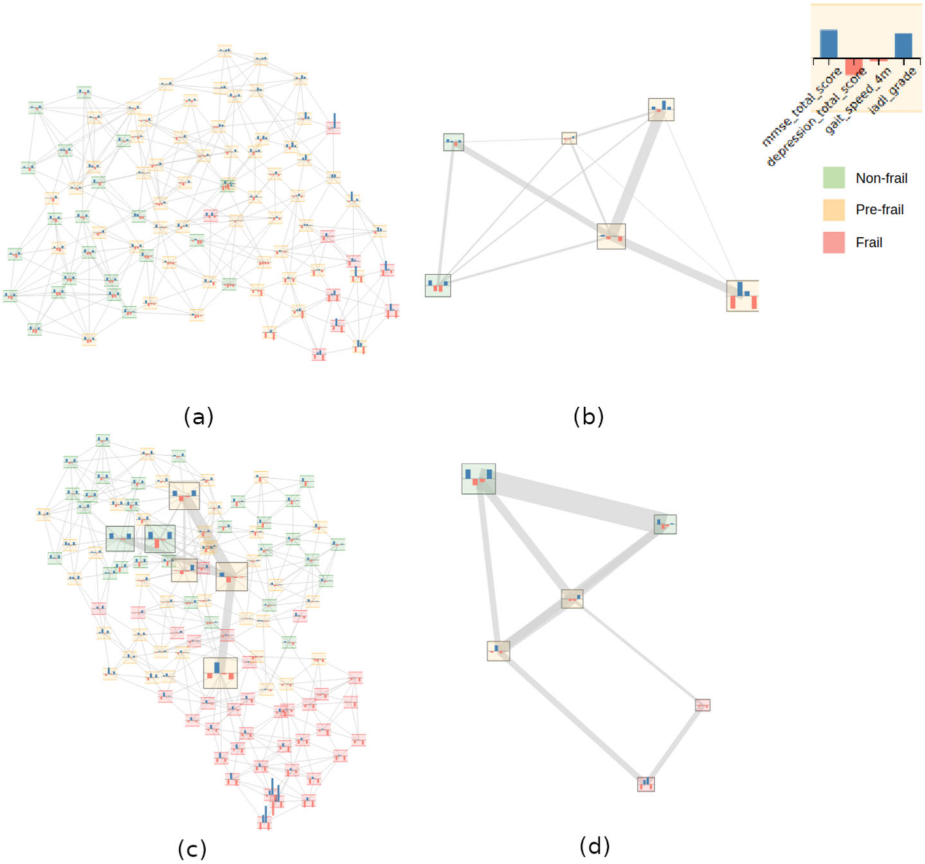


Fig. 9 Incremental graph construction for the FrailSafe dataset. **(a-b)** Graph of cohort #1. **(c-d)** Merging cohort #1 with cohort #2

follow roughly a similar distribution. However, there are several nodes at the bottom, corresponding to several frail people existing in the second cohort, which seem to be far from the group nodes. Further reducing this graph in Fig. 9d shows that the federated groups have been updated, with the previous groups aggregating most of the new patients, while there two new group nodes at the bottom, corresponding to the new group of frail patients that has appeared in the second cohort. This is an expected behaviour for the update of the federated graph, whenever new types of patients are added.

4.2 Visualization of Sjögren's Syndrome patients

The second use case concerns the visualization of patients with Sjögren's Syndrome. Sjögren's Syndrome is a disorder of the immune system, causing mostly dryness in the eyes and mouth of the patients, while it is also associated with cancer of the lymph nodes (lymphoma).

The data used in this case have been collected during the European project HarmonicSS [8]. The data consist of 200 patients, each described with a variety of clinical and laboratory measurements. Not all patients have information for all attributes. For the purposes of the current demonstration, we have used the following attributes:

- **SGE**: The Salivary Gland Echography measurement.
- **Lymphadenopathy**: Whether the person has lymphadenopathy.
- **C3**: Concentration of Complement C3 proteins.
- **C4**: Concentration of Complement C4 proteins.
- **Dryness upper resp.**: Dryness of the upper respiratory system, as subjectively qualified by the patient.
- **Reynaud**: Existence of Reynaud's disease.
- **Lymphoma**: A binary categorical variable, indicating whether the person has lymphoma, i.e. cancer of the lymph nodes.

Not all parameters are used in all the visualizations that follow. In each visualization, the parameters used are shown in the legend.

The visualization of the HarmonicSS dataset is depicted in Fig. 10. All the above attributes, apart from "lymphoma" have been used for the graph construction and the node glyphs. The number of nearest neighbors for the graph construction has been set to 7, similar to the FrailSafe use case. The "lymphoma" attribute has been used to color the glyphs: red for presence of lymphoma and green for its absence. The most apparent characteristic of the visualization is the separation of the patients in three clusters. As deduced from the node glyphs, the large cluster at the right consists of patients with low SGE values (left-most bar), while the other two clusters are separated based on the lymphadenopathy values, with the bottom cluster consisting of people with high lymphadenopathy values. With regard to the lymphoma scores, the bottom cluster contains most of the lymphoma cases, relatively to its size, but individual lymphoma cases can also be seen in the other clusters, especially in areas of low C3 and C4 values (top left of left cluster and top of right cluster).

Figure 11 shows another visualization of the whole HarmonicSS dataset, using radar glyphs. Similar to the FrailSafe scenario, we have added two extra attributes for this case: dryness of the upper respiratory system and Raynaud's index. The various areas and visual clusters formed correspond to different patient profiles, as denoted by the characteristic

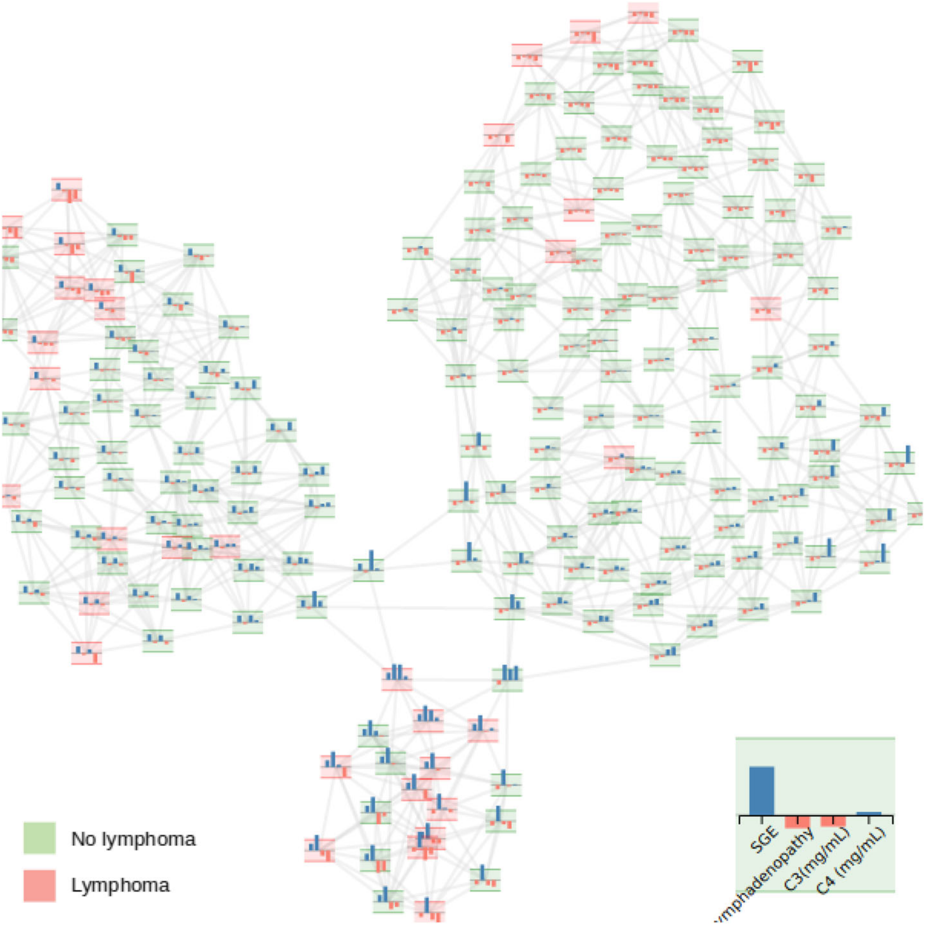


Fig. 10 Visualization of the HarmonicSS dataset, using bar chart glyphs and the nearest neighbors graph construction method

shapes of the glyphs. As examples, the middle part of the graph consists mostly of average users, with regular polygon shapes, while the top left and bottom right part consist of rather irregular shapes, elongated in the C3 and lymphadenopathy directions (top left) and the respiratory system dryness direction (bottom right). Such an overview could provide an indication to a medical professional that e.g. people with non-zero SGE values (top right part of the graph) seem to have an increased chance of developing lymphoma, since there are more persons with lymphoma among them compared to the middle part of the graph, whose glyphs differ mostly in their SGE axis. This can lead an end-user to further investigation, using statistical methods to see if such a relationship is significant.

Similar to the FrailSafe use case, we also demonstrate an incremental graph construction scenario, in Fig. 12. The dataset has been split to two cohorts, in order to construct a federated graph from one cohort and update it using the other. The graph of the first cohort

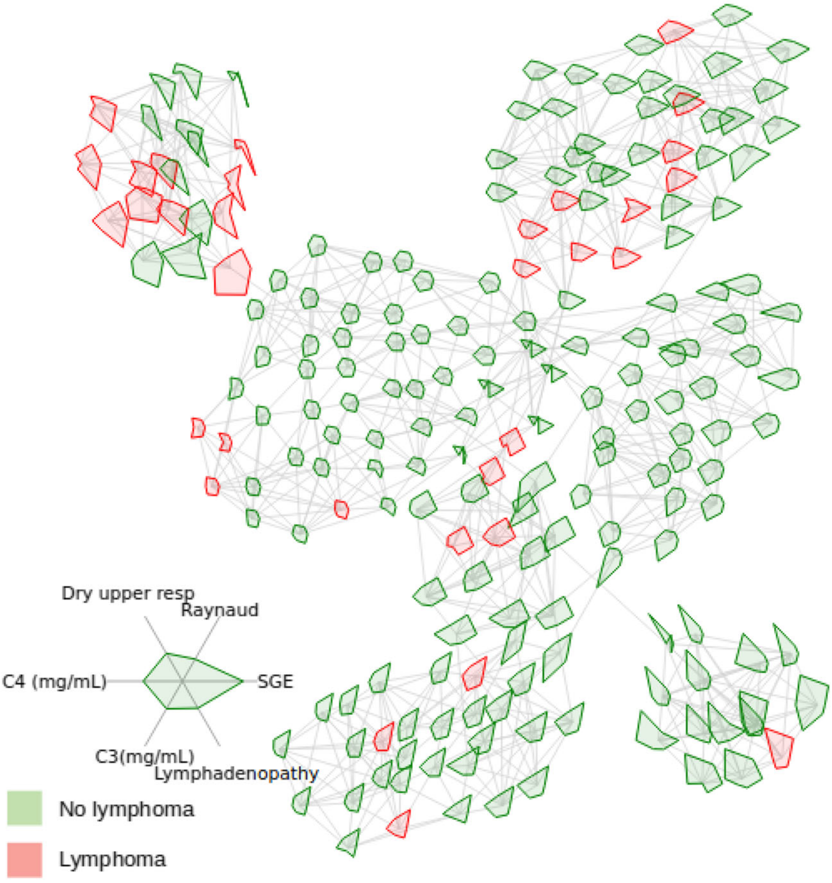


Fig. 11 Visualization of the HarmonicSS dataset, using radar chart glyphs and the nearest neighbors graph construction method

in Fig. 12a is partitioned to the graph of Fig. 12b, where the group nodes correspond to the various areas of the original graph. Since there are only a few patients with lymphoma, all group nodes are colored in green, since the majority of patients in each group does not have lymphoma. This situation is changed when the second cohort is introduced, in Fig. 12c. There are several patients with lymphoma in this cohort (red nodes), creating their own group in the updated federated graph of Fig. 12d.

4.3 Visualization of a large-scale diabetes dataset

As a final use case, we examine how the proposed method can handle large datasets of thousands of patients. The dataset used is a subset of the data used for a 2014 study for the effect of haemoglobin measurements on hospital readmission rate [26], available on Kaggle[3]. We have used a subset of the original data, i.e. the patients for whom there is an

[3] https://www.kaggle.com/brandao/diabetes

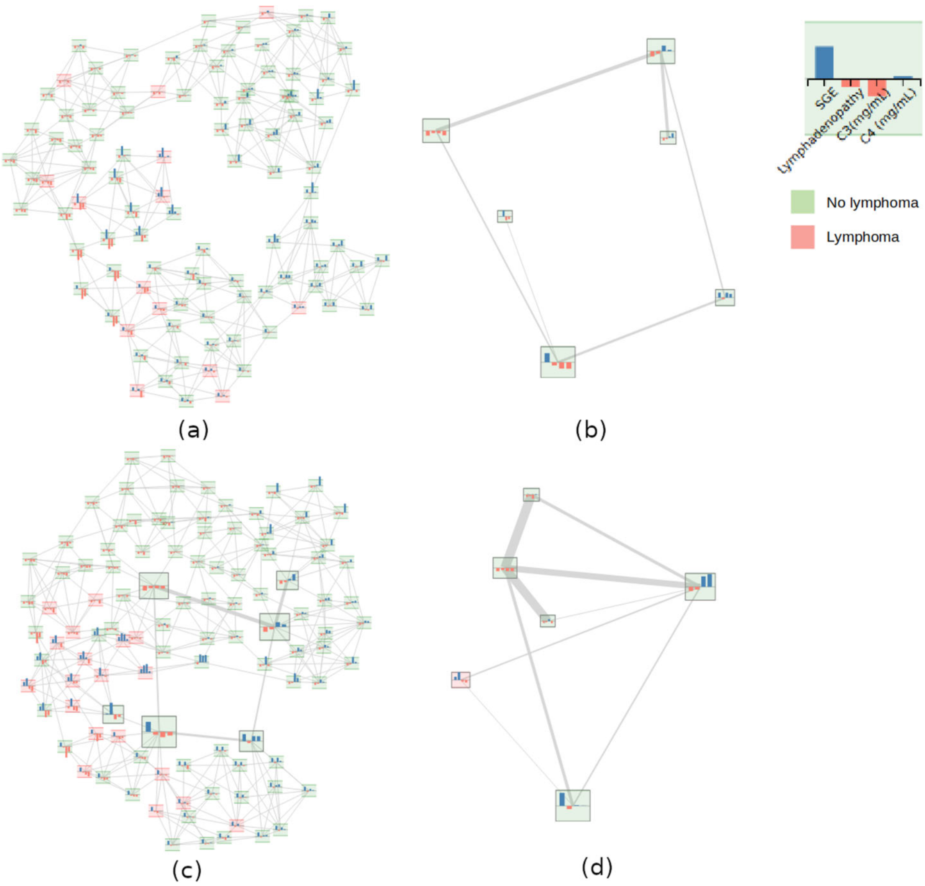


Fig. 12 Incremental graph construction for the HarmonicSS dataset, using a k-nearest neighbors graph. **(a-b)** Graph of cohort #1. **(c-d)** Merging cohort #1 with cohort #2

HbA1C measurement available. The size of this subset is 17000 patients. The attributes used for this use case are the following:

- **Age**: The age of the patient in years, binned into ten-year wide bins.
- **Time at hospital**: The hospitalization duration, in days.
- **HbA1C value**: The result of the HbA1C measurement of the patient.
- **Readmission**: Whether or not the patient is known to have been readmitted to the hospital.

The current implementation of the proposed method allows for the visualization of graphs of up to about 2000 nodes without significant overhead in time or memory consumption. As an example, Fig. 13a shows the visualization of the first 2000 patients of the dataset, using the k-nearest neighbors method with $k = 150$, while Fig. 13b shows the result of graph partitioning using 20 components. In the glyphs, the age, hospitalization time and HbA1C values were used for the glyph bars (as well as for the graph construction), while the readmission value was used for the background color (red for readmission, green for no readmission).

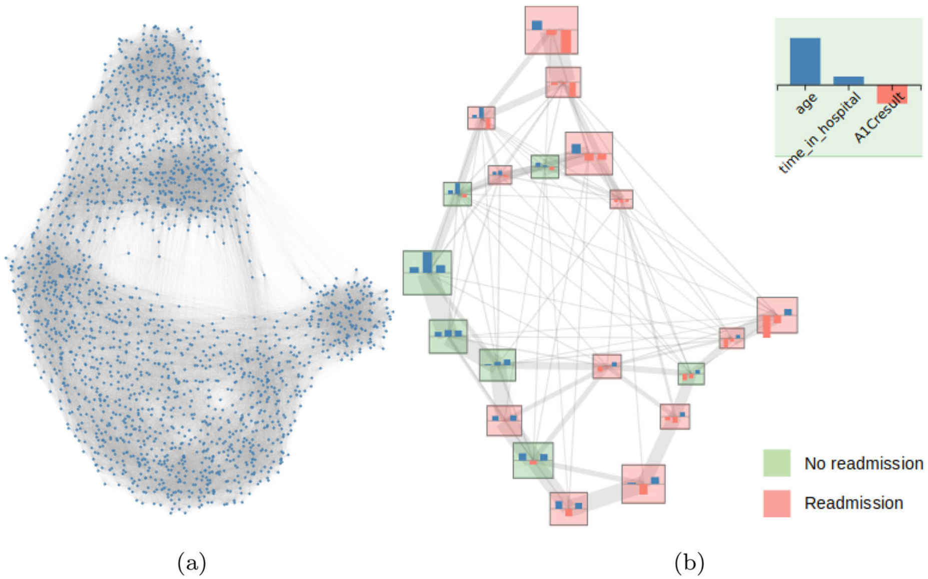


Fig. 13 Visualization of the first 2000 patients of the diabetes dataset. **(a)** The graph of all patients. **(b)** The partitioned graph

However, moving to graph sizes beyond 2000 nodes requires large amounts of memory and computational time to process, making it hard to produce a direct visualization of the whole dataset. The bottlenecks in the computation are the graph construction using the pairwise distance matrix, the placement of the graph on the 2D plane using force-directed methods and the computation of the spectral embedding, requiring the eigendecomposition of the Laplacian matrix. Modifications of the proposed method can be applied in order to address these bottlenecks:

- The construction of the graph using k-nearest neighbors consists of the computation of the pairwise distance matrix, which requires $O(N^2)$ time (the dimensionality of the points is considered constant, for simplicity of presentation), and the computation of the k nearest neighbors for each point, which, using a naive linear search, requires $O(kN^2)$ time, for a total complexity of $O(kN^2)$. Using $k-d$ trees as a data structure for storing the points of the data space, this can be reduced to an average of $O(kNlogN)$, which is much faster for large numbers of points.
- The force-directed placement of a large graph can be achieved efficiently using layout algorithms such as the Fast Multipole Multilayer Method (FMMM) [11], which are able to visualize graphs of several thousands or millions of nodes in a small amount of time (∼5min).
- The Laplacian matrix is a symmetric matrix that is sparse for small enough values of k, for a k-nearest neighbour graph. The fact that it is sparse can be exploited to optimize the storage memory required and the computation time for spectral decomposition. Moreover, not all eigenvectors of the Laplacian are needed, but only those corresponding to the smallest eigenvalues. Fast algorithms such as Chebyshev–Davidson [32] can be used to efficiently compute the graph embedding.

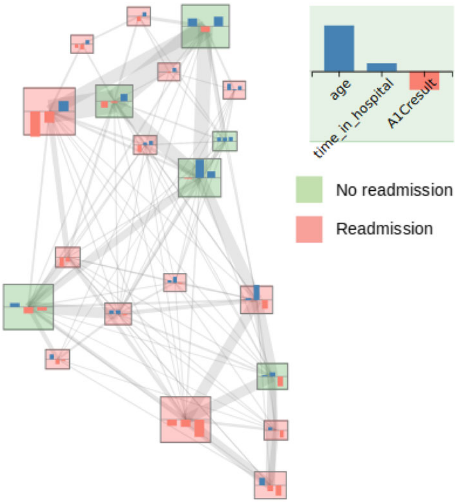


Fig. 14 Visualization of the partitioning of the whole diabetes dataset of 17000 records

Nevertheless, the incremental graph construction procedure presented in Section 3.3, allows for an alternative approach to the problem. Instead of visualizing and partitioning the whole graph, we can split the dataset into smaller batches and construct the graph and its partitioning incrementally, taking each batch in sequence. In this manner, we can summarize arbitrarily large graphs with only a time latency due to the repeated application of the algorithms in each batch.

To create a partitioning of the whole diabetes dataset, we split the 17000 patients into 17 batches of 1000 patients each, and performed incremental graph construction in each batch in sequence, building the graph of one batch around the reduced graph of the previous batch. Each graph was built using the k-nearest neighbor method with $k = 75$ and 20 components were used for graph partitioning. Figure 14 shows the partitioned graph resulting from the final step of this procedure. Nearby nodes correspond to similar groups of patients, so the end-user can distinguish the different types of users that exist in the whole dataset. For instance, the bottom part of the graph corresponds to people of low HbA1C values, indicated by the third column of the bar chart glyphs. These patients tend to have higher probabilities of readmission, indicated by the red background color, although other areas of the graph have high readmission rates as well (e.g. the top-left part). A clinician can use such a visualization of a large dataset to have an overview of the kinds of patients in it. Interaction can also be used to allow the end-user to select the attributes to use for the bars and the color of the glyphs, facilitating exploration.

5 Conclusion

In this paper, a graph-based method for visualizing a set of patients has been presented. In the proposed method, a graph is constructed that encodes local similarities among the patients in a high-dimensional space determined by a set of health parameters of interest.

The graph is presented on the two dimensional screen using force-directed placement methods, forming a visual map of the manifold spanned by the patients and allowing the visual detection of neighborhoods and clusters of similar characteristics. In addition, the graph nodes are visualized using glyphs able to depict multiple attributes for each single patient, thus providing visual cues for the type of patients present in each graph area and for the reason that certain patients are put near or away each other. The graph construction procedure is further extended in an incremental graph construction procedure, where a high-level graph of groups of nodes can be sequentially updated with data from new patient cohorts. This high-level representation is valuable in case of very large volumes of raw data, where summarized information is more appropriate, as well as in cases of sensitive data visualization, where access to individual patients is restricted. The proposed method has been demonstrated in three use cases, where the end results provide insight in the raw data, allowing the detection of patterns and outliers.

The envisioned end-users of the proposed method are decision makers and researchers of a domain of interest. The above presentation has been focused on health applications, where the end-users are clinicians and medical researchers. The value of the proposed method for the end-users is that it aims to assist in data exploration and comprehension, by providing a two-dimensional "map" of entities of interest (e.g. patients), that can serve as an overview of the available dataset and a guide towards further investigation. The entities of interest are positioned according to their similarities, which allows the formation of regions of interest, corresponding to different types of entities present in the data. On top of this, the node glyphs allow the end-user to see which features characterize each area. A clinician, for instance, can locate a patient in this map and find similar patients or groups of patients, that can in turn lead to selection of appropriate medication or intervention strategy. Or they can spot patients that do not belong in one of the coarse categories identified, which may be a hint for an abnormality requiring special attention.

Part of the presented work, related to the visualization of the raw graph of patients, prior to any reduction procedure, has been evaluated by medical professionals and researchers in the context of the FrailSafe project [31]. The feedback collected by the end-users was positive in terms of the usefulness and usability of the developed approach, which was an incentive to improve and extend the original approach. The evaluation of the complete method, involving the construction of the incremental graph and the addition of node glyphs, will be performed in the context of the currently running HarmonicSS project [8]. Since the outcome of the methods is a set of visualizations with the aim to assist the end-users in their analyses, the evaluation will be mostly based on feedback by the end-users. The evaluation process will involve the presentation of the methods to the end-users (clinicians and clinical researchers) in the context of an integrated platform, the usage of the methods by the end-users for an indicative period of time (e.g. a week) and the collection of feedback through questionnaires regarding the usefulness and usability of the methods. Through the questionnaires, we will focus on parts of the methods that might be novel to the end-users, such as the graph presentation of patients or the use of glyphs, and on how they could potentially be improved. The goal of the evaluation will be to identify the strong and weak points of the proposed solutions and to try to address the shortcomings in future releases and evaluations.

The presented method is a work in progress. The usage of the method in real-world scenarios will dictate the directions for future improvements. In any case, future work will also be directed towards testing and fine-tuning different types of glyphs that may provide more information while also leading to less clutter. Variations of the force-directed node positioning methods will also be considered, in an attempt to exploit all available space, possibly by forcing the nodes to lie on a two-dimensional grid. Effort will also be put to enhance the federated graph construction scheme with better handling of vectors of both numerical and categorical data.

Acknowledgements This work has been supported by the EU H2020 projects FrailSafe (H2020-PHC-21-2015, grant agreement no. 690140) and HarmonicSS (H2020-SC1-2016-RTD, grant agreement no. 731944).

References

1. Beck F, Burch M, Diehl S, Weiskopf D (2017) A taxonomy and survey of dynamic graph visualization. In: Computer graphics forum, vol 36, pp 133–159. Wiley Online Library
2. Bhavnani SK, Drake J, Divekar R (2014) The role of visual analytics in asthma phenotyping and biomarker discovery
3. Borgo R, Kehrer J, Chung DavidHS, Maguire E, Laramee RS, Hauser H, Ward M, Chen M (2013) Glyph-based visualization: Foundations, design guidelines, techniques and applications. In: Eurographics (STARs), pp 39–63
4. Cao N, Lin Y-R, Gotz D, Du F (2018) Z-glyph: Visualizing outliers in multivariate data. Inf Vis 17(1):22–40
5. Dix A, Ellis G (1998) Starting simple: adding value to static visualisation through simple interaction. In: Proceedings of the working conference on advanced visual interfaces, pp 124–134
6. Drosou A, Kalamaras I, Stavros P, Dimitrios T (2016) An enhanced graph analytics platform (gap) providing insight in big network data. J Innov Digit Ecosyst 3(2):83–97
7. EU H2020 (2018) European project FrailSafe: Sensing and predictive treatment of frailty and associated co-morbidities using advanced personalized patient models and advanced interventions. https://frailsafe-project.eu/
8. EU H2020 (2020) European project HarmonicSS: Harmonization and integrative analysis of regional, national and international Cohorts on primary Sjögren's Syndrome (pSS) towards improved stratification, treatment and health policy making. https://www.harmonicss.eu/
9. Fried LP, Tangen CM, Jeremy W, Newman AB, Hirsch C, Gottdiener J, Seeman T, Tracy R, Kop WJ, Burke G (2001) Frailty in older adults: evidence for a phenotype. J Gerontol A Biol Med Sci 56(3):M146–M157
10. Fuchs J, Isenberg P, Bezerianos A, Keim D (2016) A systematic review of experimental studies on data glyphs. IEEE Trans Visual Comput Graph 23(7):1863–1879
11. Hachul S, Jünger M. (2005) Large-graph layout with the fast multipole multilevel method, Spring, V, (December), 1–27
12. Jeffrey H, Danah B (2005) Vizster: Visualizing online social networks. In: IEEE symposium on information visualization, 2005. INFOVIS 2005., pp 32–39. IEEE
13. Jiawei L, Si YW (2020) Clustering-based force-directed algorithms for 3d graph visualization. J Supercomput 76(12):9654–9715
14. Kalamaras I, Drosou A, Tzovaras D (2014) Multi-objective optimization for multimodal visualization. IEEE Trans Multimed 16(5):1460–1472
15. Kwon O-H, Muelder C, Kyungwon L, Kwan-Lium M (2016) A study of layout, rendering, and interaction methods for immersive graph visualization. IEEE Trans Visual Comput Graph 22(7):1802–1815
16. Li L, Wei-Yi C, Glicksberg BS, Gottesman O, Tamler R, Chen R, Bottinger EP, Dudley JT (2015) Sci Transl Med. In: Identification of type 2 diabetes subgroups through topological analysis of patient similarity, vol 7, pp 311ra174–311ra174
17. Linhares CDG, Travençolo BAN, Paiva JGS, Rocha LEC (2017) Dynetvis: a system for visualization of dynamic networks. In: Proceedings of the symposium on applied computing, pp 187–194
18. Liu J, Bier E, Wilson A, Guerra-Gomez JA, Honda T, Sricharan K, Gilpin L, Davies D (2016) Graph analysis for detecting fraud, waste, and abuse in healthcare data. AI Mag 37(2):33–46
19. Opach T, Popelka S, Dolezalova J, Ketil JR (2018) Star and polyline glyphs in a grid plot and on a map display: which perform better. Cart Geogr Inf Sci 45(5):400–419
20. Pai S, Bader GD (2018) Patient similarity networks for precision medicine. J Molec Biol 430(18):2924–2938

21. Polychronidou E, Kalamaras I, Votis K, Tzovaras D (2018) Towards visualizing primary sjögren's syndrome data from heterogeneous cohorts. In: Proceedings of the 10th hellenic conference on artificial intelligence, pp 1–4
22. Polychronidou E, Kalamaras I, Votis K, Tzovaras D (2019) Health vision: An interactive web based platform for healthcare data analysis and visualisation. In: 2019 IEEE Conference on computational intelligence in bioinformatics and computational biology (CIBCB), pp 1–8. IEE
23. Polychronidou E, Xochelli A, Moschonas P, Papadopoulos S, Hatzidimitriou A, Vlamos P, Stamatopoulos K, Tzovaras D (2017) Chronic lymphocytic leukemia patient clustering based on somatic hypermutation (shm) analysis. In: GeNeDis 2016, pp 127–138. Springer
24. Ribassin-Majed L, Marguet S, Lee AnneWM, Ng WT, Ma J, Chan AnthonyTC, Huang P-Y, Zhu G, Chua DTT, Chen Y et al (2017) What is the best treatment of locally advanced nasopharyngeal carcinoma? an individual patient data network meta-analysis. J Clin Oncol 35(5):498
25. Ropinski T, Oeltze S, Preim B (2011) Survey of glyph-based visualization techniques for spatial multivariate medical data. Comput Graph 35(2):392–401
26. Strack B, DeShazo JP, Gennings C, Olmo JL, Ventura S, Cios KJ, Clore JN (2014) Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. BioMed research international 2014
27. Valdivia P, Buono P, Plaisant C, Dufournaud N, Fekete J-D (2020) Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. IEEE transactions on visualization and computer graphics
28. Ward M (2008) Multivariate data glyphs: Principles and practice. In: Handbook of data visualization, pp 179–198. Springer
29. Widanagamaachchi W, Livnat Y, Bremer P-T, Duvall S, Pascucci V (2017) Interactive visualization and exploration of patient progression in a hospital setting. In: AMIA annual symposium proceedings, vol 2017, pp 1773. American Medical Informatics Association
30. William JRL (2012) Combing the hairball with biofabric: a new approach for visualization of large networks. BMC Bioinform 13(1):1–16
31. Zacharaki EI, Deltouzos K, Kalogiannis S, Kalamaras I, Bianconi L, Degano C, Orselli R, Montesa J, Moustakas K, Votis K et al (2020), An ict platform for unobtrusive sensing of multi-domain frailty for personalized interventions. IEEE Journal of Biomedical and Health Informatics, Frailsafe
32. Zhou Y, Saad Y (2007) A chebyshev–davidson algorithm for large symmetric eigenproblems. SIAM J Matr Anal Appl 29(3):954–971