Mirrored Dendrograms: an Unsupervised Semi-structured and Feature-based Interactive Data Visualization Tool

Angela Moufarrej
E.C.E. Department,
Lebanese American University
36 Byblos, Lebanon
angela.moufarrej@lau.edu

Abdulkader Fatouh
E.C.E. Department,
Lebanese American University
36 Byblos, Lebanon
abdulkader.fatouh@lau.edu

Joe Tekli[*1]
E.C.E. Department,
Lebanese American University
36 Byblos, Lebanon
joe.tekli@lau.edu

Abstract—**Visualizing the correlations between structured data features is of central importance for effective and efficient data analysis and decision-making. In this paper, we present a new unsupervised semi-structured and feature-based tool for interactive data visualization titled "mirrored dendrograms". It accepts as input semi-structured and multi-featured data, and allows the user to select the target features to be visualized and mapped against each other, and their relative impacts (weights) on the visualization process. It then invokes a hierarchical clustering process to cluster the data following the user-chosen features, and produces a dendrogram structure for each combination of target features. The dendrograms are mirrored against each other by mapping their nodes using the transportation optimization problem. Different from existing solutions like tanglegram and cluster heatmap, mirrored dendrograms offers three main contributions: (i) connecting the dendrograms through their internal nodes to describe their structure relationships (instead of connecting their leaf nodes only), (ii) allowing to zoom-in and out of the data to show their relationships at different granularity levels (compared with existing static solutions), and (iii) identifying the best zooming level between the two dendrograms which highlights the maximum correlation with the minimal amount of details presented to the user (acquiring the most value out of the data, while viewing the least amount of data). We have evaluated our solution using multiple use case scenarios, including Electronic Health Records (EHRs), IMDB publications, IMDB movie entries, and Semantic SVG Graph (SSGs) instances. A number of 60 testers participated in quantitative and qualitative evaluations to assess the data visualization tool, compared with existing solutions namely tanglegrams and cluster heatmap. Testers evaluated visual quality by measuring i) the time needed by a user to identify the matching features between two data entries, and ii) the accuracy of the mapped features identified by the user. Two-sample t-tests were conducted to verify the statistical significance of the results obtained for the sample data groups being compared. A qualitative survey was also conducted to evaluate the tools' usability, interactivity, and data zooming quality. Results are promising and highlight the tool's quality and potential compared with its alternatives.**

Keywords—**Data Visualization, Data Clustering, Dendrogram, Feature Correlation, Similarity Computation, Data Granularity.**

1. Introduction

In a time when data is experiencing a remarkable growth in different fields, extracting and understanding the correlations between different data features is becoming increasingly important in many application areas, ranging over business, demographics, politics, and medicine, among others, e.g., [12, 15, 47]. The proper exploitation of such data introduces many challenges in terms of data analysis and visualization, to allow effective and efficient decision-making. The problem is further aggravated on the Web where the data is often loosely structured and multi-featured. In this context, interactive data visualization comes into play as a promising solution to facilitate data analysis. Data visualization allows unveiling patterns and trends that could be repeated over time and space, and helps experts identify anomalies in the data [47, 56]. It also adds a layer of abstraction, by providing a clear and creative presentation of the data allowing to better reach the target audience.

In this paper, we introduce a new unsupervised semi-structured feature-based tool for interactive data visualization titled *mirrored dendrograms*. It accepts as input semi-structured and multi-featured data and allows the user to select the target features to be visualized and mapped against each other. A hierarchical clustering process is invoked to cluster the data and produce a dendrogram structure for each combination of target features. The dendrograms are then mirrored against each other, where their leaf nodes are displayed at the center, and their root nodes on the sides. Their internal nodes are mapped against each other, identifying the best connections using the transportation optimization problem. The initial design of mirrored dendrograms and its primary results are described in [43]. This paper extends the mirrored dendrograms tool and functionalities, and adds a substantial experimental evaluation to assess its performance in different use cases and with different datasets. The new tool recommends the best zooming level to display the dendrograms, by introducing a new dedicated measure computing the maximum correlation (similarity) and the minimal amount of details (granularity) presented to the user. This is based on our intuition that users wish to acquire the most value out of the data while viewing the least amount of data, i.e., with the least amount of effort. The tool also provides new interactive visualization capabilities, allowing the user to adjust the zooming level and the number and weight of the

[*] Corresponding author.

connections between the mirrored dendrograms. Connection width is automatically adjusted to reflect the mapped nodes' similarity scores. Connection colors can be automatically adjusted to reflect different sub-clusters within the connected dendrograms. Visual snippets can be automatically added from the source datasets to provide a visual description of the connected sub-clusters through their root nodes.

Different from existing solutions in the literature, mirrored dendrograms: 1) process structured data (in contrast with parallel coordinates which describe the relationships between sets of flat data, and are not designed to compare structured data), 2) build cluster dendrograms to describe the structural relationships between data items (in contrast with graph-based techniques which focus on improving the visualization of graph nodes and connections rather than comparing pairs of datasets), 3) compute the structural similarity between two dendrograms (this is partially achieved with tanglegram and cluster heatmap, which only compare structured data according to their leaf node ordering, disregarding their inner node structural similarities). Furthermore, compared with its most related alternatives namely tanglegram and cluster heatmap, our solution provides three original contributions: i) connecting the dendrograms through their internal nodes to describe their structure relationships (instead of connecting their leaf nodes only - this is often misleading when evaluating the correlation between tree structures, since two trees can have different internal structures, while their leaf nodes are presented in a matching order, and vice versa), (ii) allowing to zoom-in and out of the data to show their relationships at different granularity levels (compared with existing static solutions), and (iii) identifying the best zooming level between the two dendrograms which highlights the maximum correlation with the minimal amount of details presented to the user (acquiring the most value out of the data, while viewing the least amount of data). To our knowledge, the latter three functionalities are not achieved with any existing tool.

We have evaluated our solution using multiple use case scenarios, including Electronic Health Records (EHRs), IMDB publications, IMDB movie entries, and Semantic SVG Graph (SSGs) instances. We used a sample dataset of 114 EHRs obtained from a private medical clinic. The study was focused on the migraine headache disorder, where multiple patient data samples were mapped and visualized against each other. We also built 120 mirrored dendrogram visualizations from the DBLP, IMDB, and SSG databases, along with their corresponding tanglegram and cluster heatmap visualizations in order to perform a comparative evaluation study. A number of 60 testers participated in quantitative and qualitative evaluations to assess the data visualization tool, compared with existing solutions. Results are promising and highlight the quality and potential of the tool.

The remainder of this paper is organized as follows. Section 2 briefly reviews existing visualization tools based on data clustering. Section 3 describes our *mirrored dendrograms* proposal. Section 4 described our experimental evaluation and results, before concluding in Section 5 with ongoing directions.

2. Related Works

We provide an overview of visualization tools based on clustering techniques, including parallel coordinates, dendrogram, tanglegram, cluster heatmap, graph-based and other visualization techniques. They seek to represent the relationships between hierarchical structures, which are mostly related to our study.

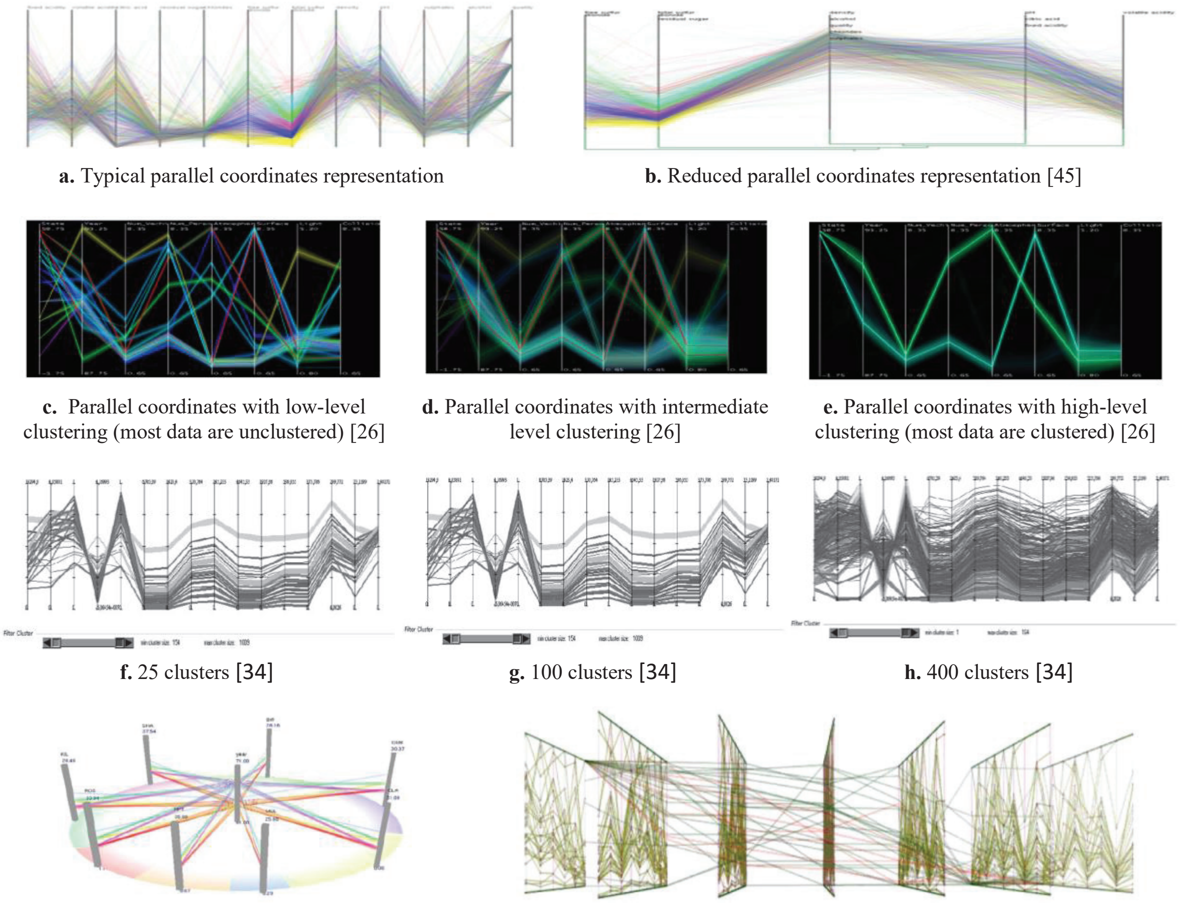
2.1. Parallel Coordinates

Parallel coordinates is a common visualization technique that aims at representing multi-dimensional datasets and extracting the underlying relationships between them (cf. Figure 1.a). Data samples are organized according to their multiple dimensions where each dimension is plotted on a separate vertical axis. In an *N*-dimensional space, a single data element is plotted as a polyline that crosses the *N* vertical axes, where its location on each axis is proportional to its value for the dimension related to that axis. Data points on adjacent axes are linked together, highlighting the correlation between the corresponding dimensions. While effective with relatively small datasets, yet this technique can suffer from cluttering when dealing with large data samples and dimensions [9, 45]. To address this problem, a few studies suggest performing dimension reduction using latent analysis or latent indexing techniques (e.g., LSA/I[2], PCA[3], word-2-vec, etc.) to reduce the number of polylines [37, 39]. A common issue with the latter approaches is the nature of the reduced dimensions which are purely algebraic (i.e., eigen vectors, or word embeddings) and might not provide useful insights for human users. More recent studies suggest reducing the number of polylines based on the visual properties of the data. The authors in [45] propose a solution based on the concept of contractible parallel coordinates, suggesting to merge highly correlated vertical axes together (cf. Figure 1.b). This requires reordering the vertical axes to get the most correlated ones next to each other, by computing pair-wise correlations between all data dimensions, and then merging the most correlated ones together into a single vertical axis.

In [26], the authors describe an extension of the parallel coordinates tool by adding hierarchical enhancements to group similar data points together along every dimension, and thus reduce cluttering along the vertical axes. They perform hierarchical clustering using the Birch algorithm [38] to provide a multi-resolution display of the data at different summarization levels (Figure 1.c-e).

In a similar study, the authors in [34] perform data point clustering using self-organizing maps, and use linked views to simplify the visualization of the resulting clusters, where modifications in one view is reflected in the other linked views (cf. Figure 1.f-h). Users can drill-down and filter clusters to view desired areas, where clusters are distinguished by their color densities. Users can also specify the number of clusters they wish to visualize, along with other parameters.

[2] Latent Semantic Analysis/Indexing
[3] Principal Component Analysis



a. Typical parallel coordinates representation

b. Reduced parallel coordinates representation [45]

c. Parallel coordinates with low-level clustering (most data are unclustered) [26]

d. Parallel coordinates with intermediate level clustering [26]

e. Parallel coordinates with high-level clustering (most data are clustered) [26]

f. 25 clusters [34]

g. 100 clusters [34]

h. 400 clusters [34]

i. Three dimensional visualization by [35]

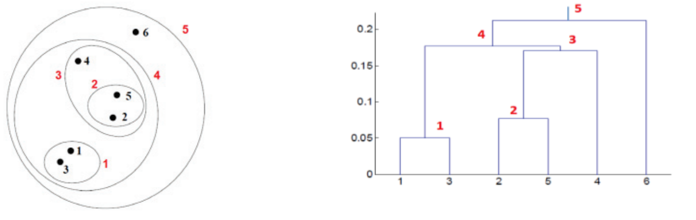
j. Three dimensional visualization by [72]

Figure 1. Sample parallel coordinates representations

In [35], the authors extend the usual two-dimensional display of parallel coordinates and introduce a new three-dimensional visualization tool called CMRPC (Clustered Multi-Relational Parallel Coordinates, cf. Figure 1.i). It allows visualizing the correlations between several features (i.e., dimensions) at a time, compared with the traditional two-dimensional display which can only visualize the correlation between two dimensions at once. CMRPC enables analyzing concurrently one-to-one relations between a central “focus” dimension and the remaining dimensions situated around it, forming a cylinder. The authors situate the different axes in-order, based on their correlation with the central dimension. In [72], the authors extend the parallel coordinates tool to add a three-dimensional visualization considering the time dimension (cf. Figure 1.j). They include multiple planes each showing a certain time stamp. This forms a group of plane clusters where each plane includes the parallel coordinates visualization depending on the timestamp of the data samples, where data sampled at the same time is represented on the same plane.

2.2. Dendrogram

A dendrogram is a diagram representing a tree that shows the hierarchical relationships between data points or objects. It is commonly used to describe the output from hierarchical clustering, and illustrates the arrangement of the clusters produced by the clustering algorithm [3]. A dendrogram consists of a hierarchy of clusters where the leaf nodes represent individual data points, the internal nodes represent clusters of data points, and the root node represents the entire data set (cf. Figure 2). Data point or object similarity can be deduced from the height of their lowermost interior node, revealing outliers through the most isolated branches. Dendrograms represent one of the main advantages of performing hierarchical clustering, since they provide a visual description, i.e., an explanation of the clustering process and how the clusters were formed, compared with other clustering techniques like partitional clustering or spectral clustering where no such explanation or visualization exists to describe the clustering process [61, 63].



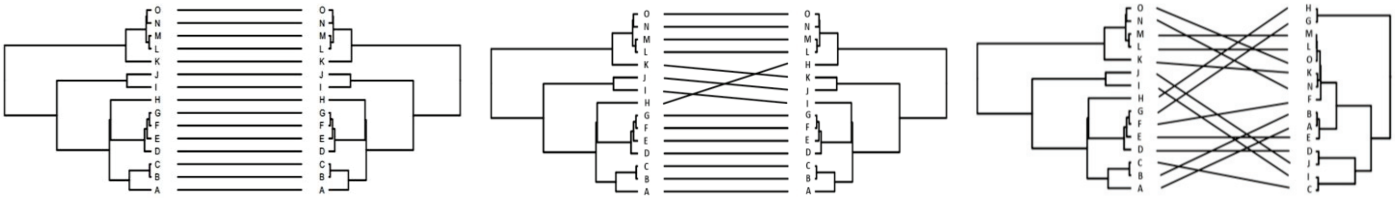
a. Hierarchical clustering

b. Dendrogram structure describing the hierarchical clustering in (a)

Figure 2. Sample hierarchical clustering and corresponding dendrogram structure

2.3. Tanglegram

Tanglegram is another representation that allows comparing two pairs of data samples having tree-like structures such as dendrograms (cf. Figure 3). Both trees should have identical leaves, representing the individual data points. Trees are visualized face-to-face, and an edge is drawn between pairs of matching leaves to connect them together. This allows depicting the spatial relationships between the connected leaves. Most of the work done on tanglegrams aims at reducing the number of line crossings (known as entanglements), to make the visualization clearer and easier to understand [13, 18]. Also, fewer (higher) crossings between the tree leaves might indicate higher (lower) correlation between the tree structures (cf. Figure 3.b and c). Nonetheless, the relationship between number of crossings and tree structure correlation is not always applicable. The trees being compared can have different internal structures or topologies, while their leaf nodes are presented in a matching order, thus producing zero crossings. In other words, tanglegrams do not visualize the similarities/disparities within the tree structures themselves; they only visualize leaf node mappings. Therefore, this visualization can be misleading when evaluating correlation between tree structures or hierarchies [18].

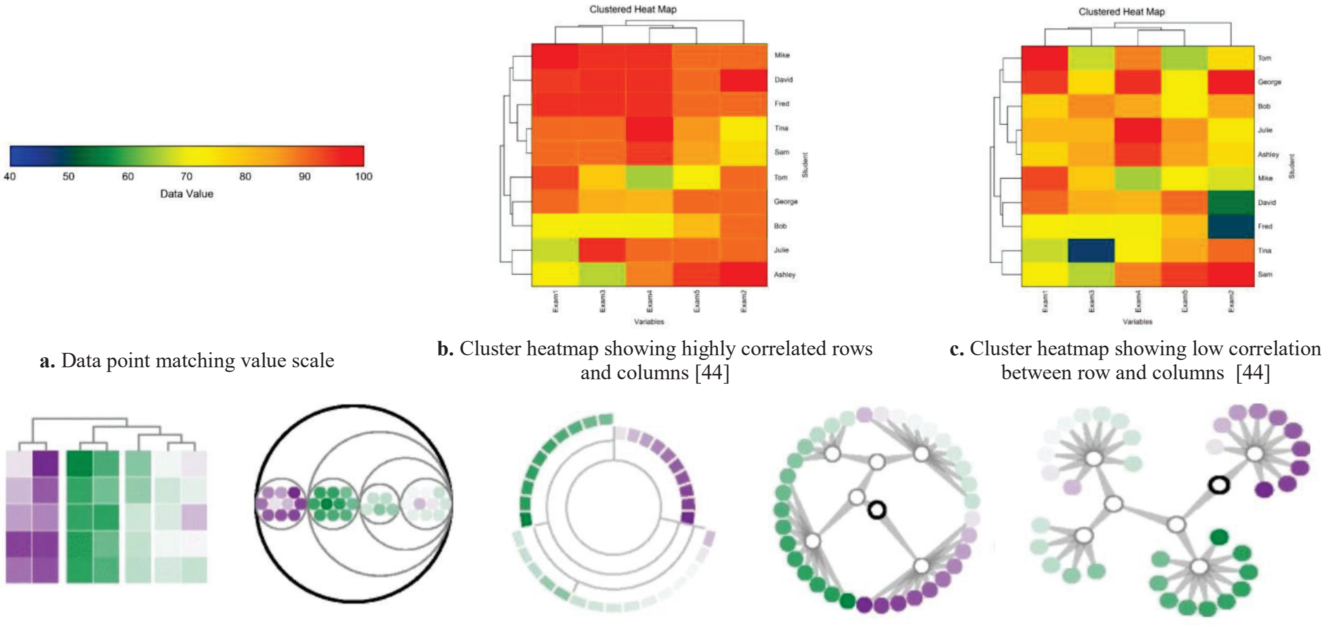

O N M L K J I H G F E D C B A
O N M L K J I H G F E D C B A
O N M L K J I H G F E D C B A
O N M L H K J I G F E D C B A
O N M L K J I H G F E D C B A
H G M L O K N F B A E D J I C

a. No crossings (matching leaves)

b. Few crossings (high correlation between leaves)

c. Many crossings (low correlation between leaves)

Figure 3. Sample tanglegram representations based on [18]



a. Data point matching value scale

b. Cluster heatmap showing highly correlated rows and columns [44]

c. Cluster heatmap showing low correlation between row and columns [44]

d. Gapmap [51]

e. Circle packing [65]

f. Sunburst [57]

g. Radial dendrogram [24]

h. Force directed tree [20]

Figure 4. Sample cluster heatmap visualizations and alternatives

2.4. Cluster Heatmap

Cluster heatmap is another representation that shows two dendrograms in a data matrix, one positioned as row and the other one positioned as column (cf. Figure 4.a-c). Within the matrix formed by the two clusters, a rectangular tiling is displayed to connect the leaf nodes of the dendrograms. Each tile is colored following a predefined color scale to reflect the value or amplitude of the matching data points in the data matrix. The rows and columns are ordered in a way such that similar leaf nodes are close to each other and the tiling colors are more visually appealing. The tiling is bordered from the top or bottom by one dendrogram and from the side by the other dendrogram [24]. Therefore, in a small display area, it simplifies the examination of row, column, and combined cluster structure, and allows showing large data matrices [28].

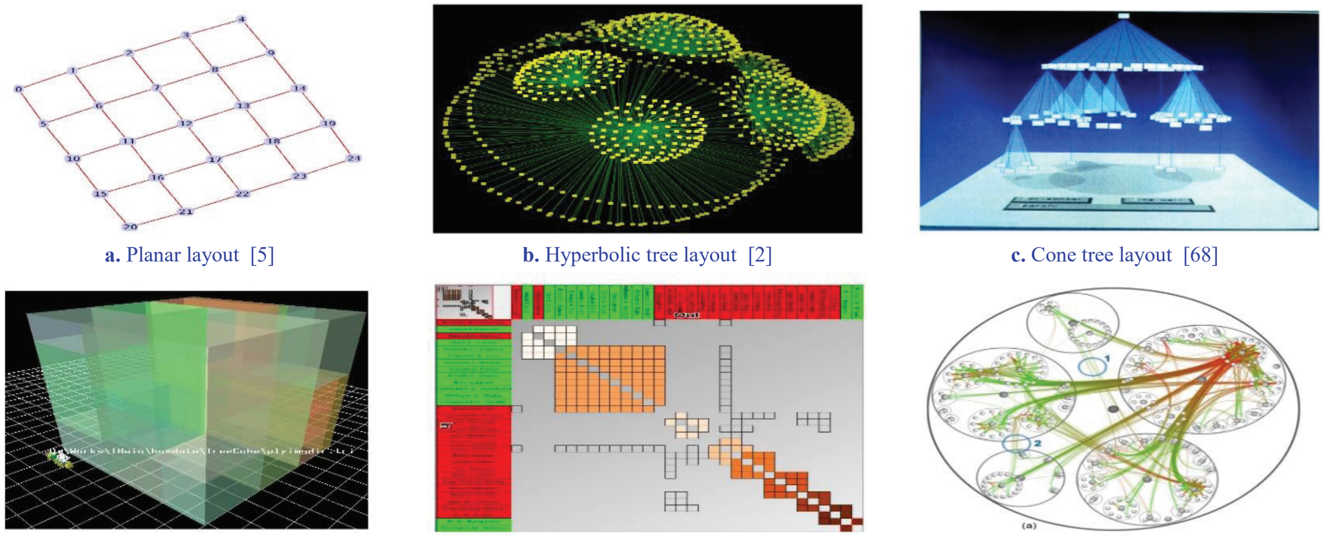
Nonetheless, rows and columns may be perceived to be highly or poorly correlated according to the ordering of their dendrogram leaf nodes, which can be misleading [50] (similarly to tanglegrams, cf. Section 2.3). Also, when clusters are formed close to the root of the dendrogram, cells that are not closely clustered must still be placed adjacent in the heatmap due to the rigid grid structure. Hence, rows or columns that are closely clustered can also end up non-adjacent in large clusters [24, 66].

Few alternatives have been suggested to compensate for the limitations of cluster heatmaps [24], including gapmap [51], circle packing [65], sunburst [57], and radial dendrogram [24] (cf. Figure 4.d-h). Yet most of them aim at improving the visualization of the clusters within an individual dataset, and do not allow comparing pairs of datasets.

2.5. Graph-based Visualization Techniques

Various techniques have been put forward to perform graph-based visualization, also referred to as link analysis or network visualization, which focus on visually describing connections between entities in graph data. A graph can be defined as a data structure used to represent relations among a set of data entities. The size and complexity of graphs can easily reach dimensions at which the task of exploring and navigating them becomes extremely difficult, hence the need for adapted graph visualization techniques. The most common approach to visualize graphs is through a node-link model, where nodes represent entities and links represent connections. The nodes and links can represent any kind of data, from transactions between business clients, to computers on a network, or posts between friends on social media. Different visualization layouts have been proposed, ranging over topological feature-based [71], planar graphs [5], tree-based [2], matrix-based [31], and cluster-based [7] (cf. Figure 5).

While they have been proven effective and practical in multiple application domains, yet graph-based visualizations mostly focus on improving the visualization of entities and connections within an individual graph, and do not specifically address the comparison of pairs of variables or pairs of datasets.



a. Planar layout [5] **b.** Hyperbolic tree layout [2] **c.** Cone tree layout [68]

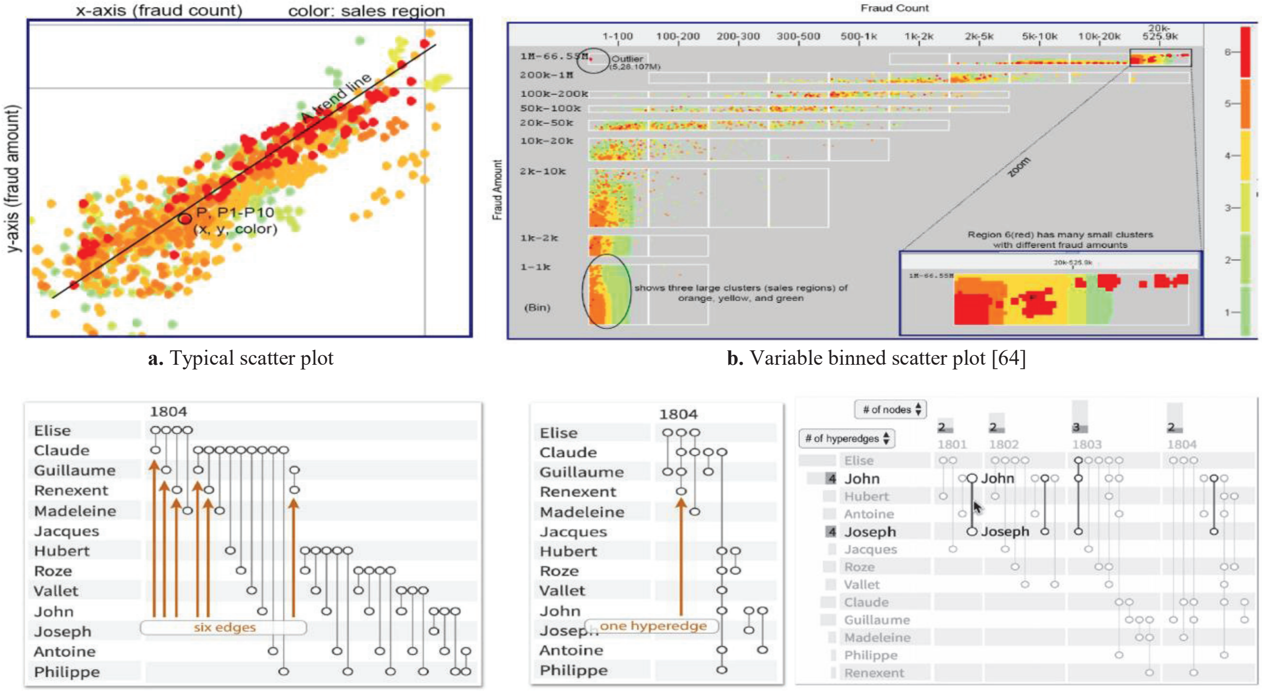
d. TreeCube layout [2] **e.** Matrix layout [31] **f.** Cluster layout [7]

Figure 5. Graph-based visualization tools

2.6. Other Visualization Techniques

A few other visualization techniques have been introduced to showcase the relationships between pairs of variables in certain application domains. The authors in [64] introduce variable binned scatter plots to visualize pairs of two-dimensional continuous variables (Figure 6.a, b). They use a non-uniform variable binning of the *x* and *y* dimensions and plot all the data points that fall within each bin into corresponding squares. They use a third color attribute for visualizing data distribution and clustering. The authors applied their technique on two use case scenarios revolving around credit fraud and data center energy consumption, comparing with traditional scatter plot visualizations. In [10], the authors introduce parallel aggregated ordered hypergraph as a visual technique to describe dynamic hypergraphs (Figure 6.c, d). The tool represents vertices as parallel horizontal bars and hyperedges as vertical lines, using dots to depict the connections to one or more vertices. The authors evaluate their tool using two digital humanities use cases revolving around legal document authorship and publication data authorship, showcasing the tool's effectiveness for medium size dynamic hypergraphs (50-500 vertices) commonly generated by digital humanities projects [10].

While the above-described visualizations showed promising results in their respective application use cases, yet they mainly aim at improving the visualization of flat data variables and do not address the comparison of structured data.



a. Typical scatter plot

b. Variable binned scatter plot [64]

c. Parallel aggregated ordered graph [10]

d. Parallel aggregated ordered hypergraphs [10]

Figure 6. Other cluster-based visualization tools

Table 1. Recap of data visualization tools based on clustering techniques

Category	**Approach**	**Contributions**	**Limitations**
Parallel Coordinates	Bok J., et al. [9]	– Original and reduced parallel coordinate representations	– Effective with relatively small datasets, – Suffers from cluttering when dealing with large data samples and dimensions
	Koren Y. [37] Lou J., et al. [39]	– Using latent analysis and latent indexing techniques to reduce the number of polylines	– Reduced dimensions which are purely algebraic (i.e., Eigen vectors, or word embeddings), which do not provide useful insights for human users
	Nohno K., et al. [45]	– Contractible parallel coordinates, suggesting to merge highly correlated vertical axes together	– Needs reordering the vertical axes to get the most correlated ones next to each other, which is not a trivial task
Dendrogram	Ahmad A. & Khan S. [3]	– Describe hierarchical clustering output – Illustrate cluster arrangement	– Does not address the comparison of pairs of variables or pairs of datasets
Tanglegram	Buchin K., et al. [13] De Vienne D. [18]	– Reduce the number of line crossings, i.e., entanglements, to make the visualization clearer and easier to understand	– Does not visualize the similarities/disparities within the tree structures; only visualizes leaf node mappings – Can be misleading when evaluating correlation between tree structures or hierarchies
Cluster Heatmap	Engle S., et al. [24] Galili T., et al. [28]	– Original cluster heatmap representations – Show relations between two structured datasets	– Rows and columns may be perceived as highly/poorly correlated according to the ordering of their dendrogram leaf nodes, which can be misleading – Clusters formed close to the root must be placed adjacent in the heatmap due to the rigid grid structure
	Sakai R. [51] Wang W., et al. [65] Stasko J. & Zhang E. [57] Engle S., et al. [24]	– Gapmap, circle packing, sunburst, and radial dendrogram – Solutions attempting to compensate for the limitations of cluster heatmap	– Aim at improving the visualization of the clusters within an individual dataset, and do not allow comparing pairs of datasets
Graph-based	Tarawaneh R., et al. [5] Tanaka Y., et al. [2] Henry N. & Fekete J. [31] Holten D. [7]	– Graph visualizations using node-link model and related techniques: – Planar, tree-based, matrix-based, and cluster-based	– Focus on improving the visualization of entities and connections within an individual graph – Do not specifically address the comparison of pairs of datasets
Other techniques	Hao M., et al. [64]	– Variable binned scatter plots to visualize pairs of two-dimensional continuous variables	– Does not address the comparison of structured data
	Valdivia P., et al. [10]	– Parallel aggregated ordered hypergraph to describe dynamic hypergraphs	– Does not address the comparison of structured data

2.7. Recap

Table 1 summarizes the properties of data visualization tools based on clustering techniques. Parallel coordinates and other visualization techniques describe the relationships between sets of flat data, and are not designed to compare structured data. Tanglegram and cluster heatmap compare structured data according to their leaf node ordering. They do not visualize the similarities within the tree structures themselves, but rather visualize their leaf node mappings. This is often misleading when evaluating the correlation between tree structures, since two trees can have different internal structures, while their leaf nodes are presented in a matching order, and vice versa. Graph-based visualization techniques focus on improving the visualization of entities and connections within an individual graph and do not specifically address the comparison of pairs of datasets. Different from existing solutions, we introduce a new tool for visualizing the correlation between two structured datasets, which computes the structural similarity between their dendrogram trees, and identifies the best structural matching between them according to their structural properties.

3. Proposal: *Mirrored Dendrograms*

We design a new tool for interactive visualization of structured data titled *mirrored dendrograms*. The overall process is depicted in Figure 7. It accepts as input two sets of semi-structured and multi-featured data, and allows the user to select the target features to be visualized. The data is then hierarchically clustered to produce a dendrogram for each combination of target features. The tool evaluates the structural similarity between the produced dendrograms to identify the best zooming level to display the data. The dendrograms' internal nodes are mapped against each other according to their structural properties, using an adaptation of the transportation optimization problem. The tool provides interactive visualization capabilities, allowing the user to adjust the zooming level, and the number and weight of the connections, allowing to adapt the visualization accordingly.

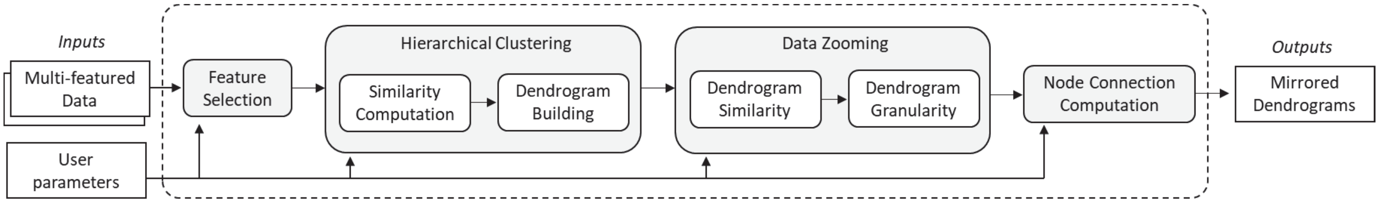
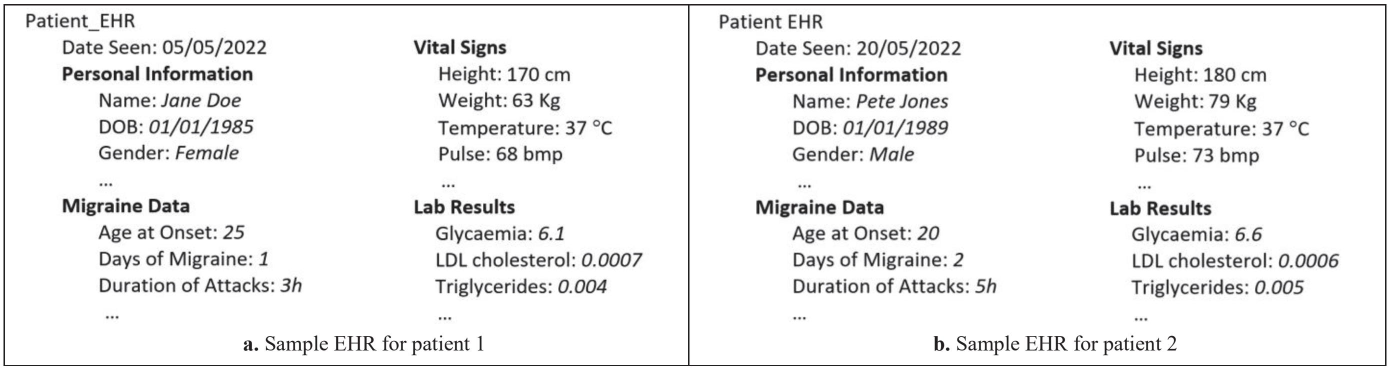


Figure 7. Simplified activity diagram describing our approach's overall architecture

3.1. Data Representation

We consider semi-structured and multi-featured data, where users choose their features of interest by assigning different weights to different features according to their preferences. In this paper, we use real-world EHRs to describe our running examples, yet any other multi-featured data can be utilized. Figure 8 shows extracts of two EHRs providing atomic feature elements (e.g., *DOB*, *days of migraine*, *age at onset*) and aggregate feature elements (e.g., *personal information*, *migraine data*, *vital signs*).



Patient_EHR
Date Seen: 05/05/2022
Personal Information
Name: *Jane Doe*
DOB: *01/01/1985*
Gender: *Female*
...
Migraine Data
Age at Onset: *25*
Days of Migraine: *1*
Duration of Attacks: *3h*
...

Vital Signs
Height: 170 cm
Weight: 63 Kg
Temperature: 37 °C
Pulse: 68 bmp
...
Lab Results
Glycaemia: *6.1*
LDL cholesterol: *0.0007*
Triglycerides: *0.004*
...

a. Sample EHR for patient 1

Patient EHR
Date Seen: 20/05/2022
Personal Information
Name: *Pete Jones*
DOB: *01/01/1989*
Gender: *Male*
...
Migraine Data
Age at Onset: *20*
Days of Migraine: *2*
Duration of Attacks: *5h*
...

Vital Signs
Height: 180 cm
Weight: 79 Kg
Temperature: 37 °C
Pulse: 73 bmp
...
Lab Results
Glycaemia: *6.6*
LDL cholesterol: *0.0006*
Triglycerides: *0.005*
...

b. Sample EHR for patient 2

Figure 8. Sample EHRs for two migraine patients

3.2. Similarity Computation

After identifying the features of interest, the next step is to perform feature similarity computation to conduct hierarchical clustering. Similarity between atomic feature elements are computed according to their feature data-types (Table 2).

Table 2. Sample atomic element and feature vector similarity measures [4, 40, 60]

Scalar values similarity	Comparing two scalar values x_i and x_j: $\mathrm{Sim}(\mathrm{x_i}, \mathrm{x_j}) = 1 - \frac{\lvert \mathrm{x_i} - \mathrm{x_j} \rvert}{\mathrm{x_{max}}} \in [0,\ 1]$ where x_{max} is the maximum value from the reference dataset from which the values were sampled.	**(1)**
Date/Time stamps similarity	Comparing two date/time stamps x_i and x_j: $\mathrm{Sim}(\mathrm{x_i}, \mathrm{x_j}) = 1 - \frac{\lvert (\mathrm{x_i} + \mathrm{x_{min}}) - (\mathrm{x_j} + \mathrm{x_{min}}) \rvert}{\lvert \mathrm{x_{max}} + \mathrm{x_{min}} \rvert} \in [0,\ 1]$ where x_{max} and x_{min} are the maximum and minimum values from the reference dataset from which the date/time values were sampled.	**(2)**
Boolean values similarity	Comparing two Boolean values x_i and x_j: $\mathrm{Sim}(\mathrm{x_i}, \mathrm{x_j}) = \mathrm{x_i} \wedge \mathrm{x_j}$	**(3)**
String values similarity	Comparing two string values syntactically x_i and x_j: $\mathrm{Sim}(\mathrm{x_i}, \mathrm{x_j}) = 1 - \frac{\mathit{EditDistance}(\mathrm{x_i}, \mathrm{x_j})}{\lvert \mathrm{x_i} \rvert + \lvert \mathrm{x_j} \rvert} \in [0,\ 1]$	**(4)**
Feature vectors similarity	Comparing two feature vectors $\mathrm{V_i}$ and $\mathrm{V_j}$: $\mathrm{Sim}(\mathrm{V_i}, \mathrm{V_j}) = \frac{1}{n} \sum_{k=1}^{n} \mathit{Sim}(x_k^i, x_k^j)$	**(5)**

Similarity between aggregate feature elements is computed as the aggregation of the similarities of their constituent atomic elements. This can be computed in several ways, using for instance the *maximum*, *minimum*, *average*, or *weighted sum* functions [59, 63]. Here, we make use of the *weighted sum* function since it enables the users to choose the weight of each atomic feature in accordance with their notion of similarity. More formally, given two aggregate feature elements E_1 and E_2:

$$\mathrm{Sim}(\mathrm{E_1}, \mathrm{E_2}) = \underset{\mathrm{i=1..n}}{f_{agg}}\left(\mathrm{Sim_i}(\mathrm{e_i^1}, \mathrm{e_i^2})\right) = \sum_{\mathrm{i=1..n}} \mathrm{w_i} \times \mathrm{Sim_i}\ (\mathrm{e_i^1}, \mathrm{e_i^2}) \in [0,\ 1]$$

$$\text{given} \sum_{\mathrm{i=1..n}} \mathrm{w_i} = 1 \ \wedge\ (\mathrm{w_{i=1..n}}) \geq 0 \ \wedge\ \mathrm{Sim_{i=1..n}}(\mathrm{x}, \mathrm{y}) \in [0,\ 1] \qquad \textbf{(6)}$$

where e_i^j is an atomic element describing feature i within aggregate element E_j, w_i is the weight of feature i, and Sim_i is the similarity according to feature i. For instance, the similarity between two patient EHRs described in Figure 8, considering aggregate feature elements made of atomic features *gender*, *pulse*, and *glycaemia*, is computed as follows:

$$\begin{aligned}\mathrm{Sim}(\mathrm{E_1}, \mathrm{E_2}) &= \mathrm{w_{gender}} \times \mathrm{Sim_{gender}}\ (\mathrm{E_1}, \mathrm{E_2}) + \mathrm{w_{pulse}} \times \mathrm{Sim_{pulse}}\ (\mathrm{E_1}, \mathrm{E_2}) + \mathrm{w_{glycaemia}} \times \mathrm{Sim_{glycaemia}}\ (\mathrm{E_1}, \mathrm{E_2}) \\ &= \frac{1}{3} \times \mathrm{Sim_{gender}}(\mathit{Female},\ \mathit{Male}) + \frac{1}{3} \times \mathrm{Sim_{pulse}}(68,\ 75) + \frac{1}{3} \times \mathrm{Sim_{glycaemia}}(6.1,\ 6.6) \\ &= \frac{1}{3} \times 0 + \frac{1}{3} \times \frac{\lvert 68\ \text{-}\ 73 \rvert}{170} + \frac{1}{3} \times \frac{\lvert 6.1\ \text{-}\ 6.6 \rvert}{147} = 0.653\end{aligned}$$

We consider as reference $pulse_{max}$ = 170 bpm and $glycaemia_{max}$ =147 mmol/L for a middle aged human subject, in order to compute the corresponding atomic similarity functions accordingly[4] (cf. Table 2).

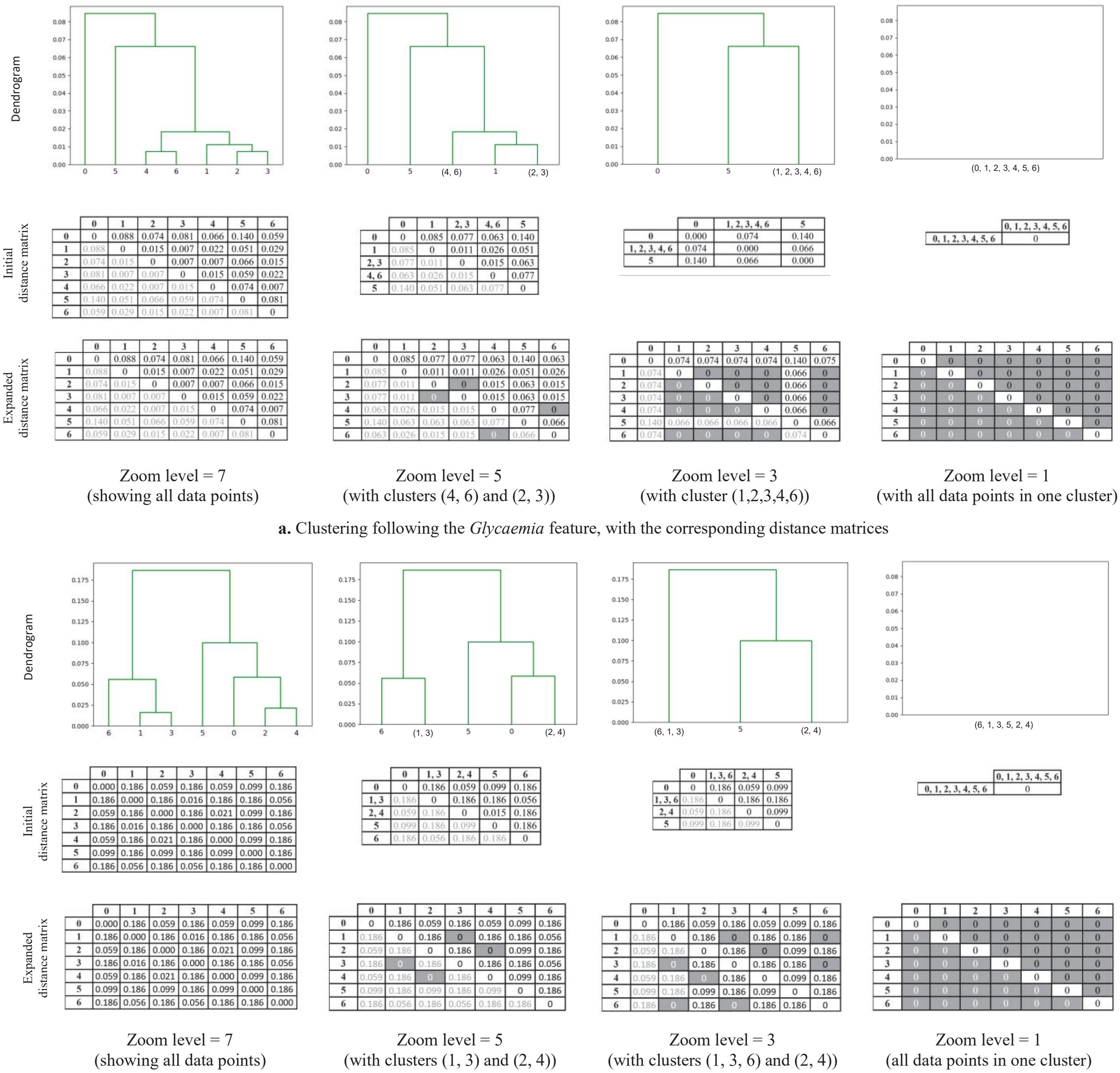
3.3. Data Clustering

In this study, we use the well-known Unweighted Pair-Group Method with Arithmetic mean (UPGMA) average link hierarchical clustering method [23, 30], although any form of hierarchical clustering can be utilized. Given n data points, we construct a fully connected graph G with n nodes and $\frac{n \times (n-1)}{2}$ weighted edges. The weight of an edge corresponds to the similarity (distance) between the connected nodes. We adopt an agglomerative clustering approach where each node in the connected graph initially represents an individual cluster. Consequently, the nearest two clusters (i.e., data points) are combined into a higher-level cluster. This is repeated iteratively at every step to combine the most similar clusters into higher-level clusters, where the similarity between

[4] *Gender* is modeled as a Boolean attribute, where *female* and *male* values are represented *true* (1) and *false* (0) respectively. We do not consider other gender types in our present use case scenario (e.g., transgender or gender neutral) since they do not exist within our patient data.

the clusters is computed as the average of all similarities between their constituent edges, i.e., the mean pair-wise similarity between all pairs of matching data points from both clusters.

Figure 9 shows the dendrograms and corresponding distance matrices produced for a sample dataset of 7 patient EHRs, clustered accordingly to the *Glycaemia* and *LDL* features[5] (cf. experiments in Section 4).



	0	1	2	3	4	5	6
0	0	0.088	0.074	0.081	0.066	0.140	0.059
1	0.088	0	0.015	0.007	0.022	0.051	0.029
2	0.074	0.015	0	0.007	0.007	0.066	0.015
3	0.081	0.007	0.007	0	0.015	0.059	0.022
4	0.066	0.022	0.007	0.015	0	0.074	0.007
5	0.140	0.051	0.066	0.059	0.074	0	0.081
6	0.059	0.029	0.015	0.022	0.007	0.081	0

	0	1	2	3	4	5	6
0	0	0.088	0.074	0.081	0.066	0.140	0.059
1	0.088	0	0.015	0.007	0.022	0.051	0.029
2	0.074	0.015	0	0.007	0.007	0.066	0.015
3	0.081	0.007	0.007	0	0.015	0.059	0.022
4	0.066	0.022	0.007	0.015	0	0.074	0.007
5	0.140	0.051	0.066	0.059	0.074	0	0.081
6	0.059	0.029	0.015	0.022	0.007	0.081	0

Zoom level = 7
(showing all data points)

	0	1	2, 3	4, 6	5
0	0	0.085	0.077	0.063	0.140
1	0.085	0	0.011	0.026	0.051
2, 3	0.077	0.011	0	0.015	0.063
4, 6	0.063	0.026	0.015	0	0.077
5	0.140	0.051	0.063	0.077	0

	0	1	2	3	4	5	6
0	0	0.085	0.077	0.077	0.063	0.140	0.063
1	0.085	0	0.011	0.011	0.026	0.051	0.026
2	0.077	0.011	0	0	0.015	0.063	0.015
3	0.077	0.011	0	0	0.015	0.063	0.015
4	0.063	0.026	0.015	0.015	0	0.077	0
5	0.140	0.063	0.063	0.063	0.077	0	0.066
6	0.063	0.026	0.015	0.015	0	0.066	0

Zoom level = 5
(with clusters (4, 6) and (2, 3))

	0	1, 2, 3, 4, 6	5
0	0.000	0.074	0.140
1, 2, 3, 4, 6	0.074	0.000	0.066
5	0.140	0.066	0.000

	0	1	2	3	4	5	6
0	0	0.074	0.074	0.074	0.074	0.140	0.075
1	0.074	0	0	0	0	0.066	0
2	0.074	0	0	0	0	0.066	0
3	0.074	0	0	0	0	0.066	0
4	0.074	0	0	0	0	0.066	0
5	0.140	0.066	0.066	0.066	0.066	0	0.066
6	0.074	0	0	0	0	0.074	0

Zoom level = 3
(with cluster (1,2,3,4,6))

	0, 1, 2, 3, 4, 5, 6
0, 1, 2, 3, 4, 5, 6	0

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

Zoom level = 1
(with all data points in one cluster)

a. Clustering following the *Glycaemia* feature, with the corresponding distance matrices

	0	1	2	3	4	5	6
0	0.000	0.186	0.059	0.186	0.059	0.099	0.186
1	0.186	0.000	0.186	0.016	0.186	0.186	0.056
2	0.059	0.186	0.000	0.186	0.021	0.099	0.186
3	0.186	0.016	0.186	0.000	0.186	0.186	0.056
4	0.059	0.186	0.021	0.186	0.000	0.099	0.186
5	0.099	0.186	0.099	0.186	0.099	0.000	0.186
6	0.186	0.056	0.186	0.056	0.186	0.186	0.000

	0	1	2	3	4	5	6
0	0.000	0.186	0.059	0.186	0.059	0.099	0.186
1	0.186	0.000	0.186	0.016	0.186	0.186	0.056
2	0.059	0.186	0.000	0.186	0.021	0.099	0.186
3	0.186	0.016	0.186	0.000	0.186	0.186	0.056
4	0.059	0.186	0.021	0.186	0.000	0.099	0.186
5	0.099	0.186	0.099	0.186	0.099	0.000	0.186
6	0.186	0.056	0.186	0.056	0.186	0.186	0.000

Zoom level = 7
(showing all data points)

	0	1, 3	2, 4	5	6
0	0	0.186	0.059	0.099	0.186
1, 3	0.186	0	0.186	0.186	0.056
2, 4	0.059	0.186	0	0.015	0.186
5	0.099	0.186	0.099	0	0.186
6	0.186	0.056	0.186	0.186	0

	0	1	2	3	4	5	6
0	0	0.186	0.059	0.186	0.059	0.099	0.186
1	0.186	0	0.186	0	0.186	0.186	0.056
2	0.059	0.186	0	0.186	0	0.099	0.186
3	0.186	0	0.186	0	0.186	0.186	0.056
4	0.059	0.186	0	0.186	0	0.099	0.186
5	0.099	0.186	0.099	0.186	0.099	0	0.186
6	0.186	0.056	0.186	0.056	0.186	0.186	0

Zoom level = 5
(with clusters (1, 3) and (2, 4))

	0	1, 3, 6	2, 4	5
0	0	0.186	0.059	0.099
1, 3, 6	0.186	0	0.186	0.186
2, 4	0.059	0.186	0	0.099
5	0.099	0.186	0.099	0

	0	1	2	3	4	5	6
0	0	0.186	0.059	0.186	0.059	0.099	0.186
1	0.186	0	0.186	0	0.186	0.186	0
2	0.059	0.186	0	0.186	0	0.099	0.186
3	0.186	0	0.186	0	0.186	0.186	0
4	0.059	0.186	0	0.186	0	0.099	0.186
5	0.099	0.186	0.099	0.186	0.099	0	0.186
6	0.186	0	0.186	0	0.186	0.186	0

Zoom level = 3
(with clusters (1, 3, 6) and (2, 4))

	0, 1, 2, 3, 4, 5, 6
0, 1, 2, 3, 4, 5, 6	0

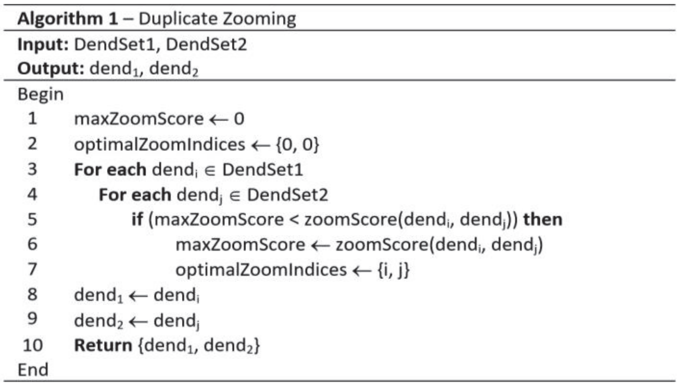
	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

Zoom level = 1
(all data points in one cluster)

b. Clustering following the *LDL* feature, with the corresponding distance matrices

Figure 9. Dendrograms produced for 7 patient EHRs clustered following the *Glycaemia* (a) and *LDL* (b) features

[5] *Glycaemia* refers to the level of glucose in the patient's blood. LDL is commonly referred to as the "bad" cholesterol since it collects in the blood vessel walls.



```
Algorithm 1 – Duplicate Zooming
Input: DendSet1, DendSet2
Output: dend_1, dend_2
Begin
1   maxZoomScore ← 0
2   optimalZoomIndices ← {0, 0}
3   For each dend_i ∈ DendSet1
4       For each dend_j ∈ DendSet2
5           if (maxZoomScore < zoomScore(dend_i, dend_j)) then
6               maxZoomScore ← zoomScore(dend_i, dend_j)
7               optimalZoomIndices ← {i, j}
8   dend_1 ← dend_i
9   dend_2 ← dend_j
10  Return {dend_1, dend_2}
End
```

Figure 10. Pseudo code of our dendrogram zooming algorithm

3.4. Data Zooming

After performing the clustering process on the selected features and producing the resulting dendrogram structures, the tool recommends the best zooming level to display the dendrograms. This is undertaken according to a combined zooming score highlighting: i) the maximum similarity between the dendrograms, and ii) the minimal granularity for both dendrograms. More formally, given two dendrograms *dend₁* and *dend₂*:

$$\text{zoomScore}(\text{dend}_1, \text{dend}_2) = \alpha \times \text{Sim}(\text{dend}_1, \text{dend}_2) + \beta \times (1 - \text{Gran}(\text{dend}_1, \text{dend}_2)) \quad \in [0, 1] \tag{7}$$

where $\alpha, \beta \in [0, 1]$, $\alpha + \beta = 1$, Sim($dend_1$, $dend_2$) $\in [0, 1]$, and Gran($dend_1$, $dend_2$) $\in [0, 1]$. Similarly to the element aggregation measure mentioned in Section 3.2, we make use of the *weighted sum* function since it allows users to emphasize dendrogram similarity versus granularity according to their needs.

The zooming algorithm is shown in Figure 10. It accepts as input two sets of dendrograms produced for both features being compared, including all zooming levels for each feature. It then computes the zooming score for each pair of dendrograms in both sets (lines 3-5) and identifies the pair which maximize the zooming score (lines 6-9).

3.4.1. Dendrogram Similarity

We evaluate the similarity between two dendrograms using their expanded distance matrices. The distance between a data point x and a cluster Y in the initial matrix, is represented as a replication of the same distance value between x and every data point $y \in Y$ in the expanded matrix. We adopt the expanded distance matrices to maintain identical dimensionalities for both matrices being compared, regardless of hierarchical clustering (zooming) level (cf. Figure 9). This allows computing the similarity between any two matrices using typical vector (matrix) similarity measures. We adopt normalized Manhattan distance to compute the similarity between a pair of data points, yet other vector similarity measures can be used (e.g., Cosine, PCC, Dice, and Euclidian). Formally:

$$\text{Sim}(\text{dend}_1, \text{dend}_2) = 1 - \text{Dist}(\text{dend}_1, \text{dend}_2) \quad \in [0, 1]$$

$$/ \quad \text{Dist}(\text{dend}_1, \text{dend}_2) = \frac{\sum_{i,j} |m_{i,j} - n_{i,j}|}{\sum_{i,j} |m_{i,j} + n_{i,j}|} \quad \in [0, 1] \tag{8}$$

where $m_{i,j}$ is the distance entry in the distance matrix corresponding to *dend₁*, and $n_{i,j}$ is the distance entry in the distance matrix corresponding to *dend₂*. Table 3.a shows the pair-wise similarity scores between pairs of dendrograms produced following our *Glycemia* vs *LDL* running example. An entry at position (4, 5) in the similarity matrix represents the similarity score between the dendrogram of zooming level =4 for *Glycemia* and the dendrogram of zooming level =5 for *LDL*.

Table 3. Similarity (a), granularity (b), and zoomScore (c) matrices for *Glycemia* vs *LDL* dendrograms

a. Similarity matrix

Dend#	1	2	3	4	5	6	7
1	NaN	0	0	0	0	0	0
2	0	0.1851	0.2226	0.2885	0.2787	0.2768	0.2755
3	0	0.2944	0.3971	0.4509	0.4369	0.4343	0.4324
4	0	0.3189	0.4162	0.4678	0.4735	0.4806	0.4786
5	0	0.3235	0.4199	0.4709	0.4766	0.4837	0.4874
6	0	0.3273	0.4232	0.4741	0.4796	0.4866	0.4904
7	0	0.3312	0.4265	0.4771	0.4825	0.4896	**0.4933**

b. Granularity matrix

1	2	3	4	5	6	7
0	0.0833	0.1667	0.2500	0.3333	0.4167	0.5000
0.0833	0.1667	0.2500	0.3333	0.4167	0.5000	0.5833
0.1667	0.2500	0.3333	0.4167	0.5000	0.5833	0.6667
0.2500	0.3333	0.4167	0.5000	0.5833	0.6667	0.7500
0.3333	0.4167	0.5000	0.5833	0.6667	0.7500	0.8333
0.4167	0.5000	0.5833	0.6667	0.7500	0.8333	0.9167
0.5	0.5833	0.6667	0.7500	0.8333	0.9167	1

c. zoomScore matrix

1	2	3	4	5	6	7
NaN	0.1833	0.1667	0.15	0.1333	0.1167	0.1
0.1833	0.3147	0.3281	0.3641	0.3396	0.3214	0.3037
0.1667	0.3855	0.4510	**0.4774**	0.4495	0.4308	0.4126
0.1500	0.3885	0.4496	0.4742	0.4621	0.4511	0.4329
0.1333	0.3755	0.4359	0.4601	0.4479	0.4370	0.4233
0.1167	0.3618	0.4219	0.4459	0.4337	0.4226	0.4090
0.1	0.3483	0.4079	0.4317	0.4193	0.4083	0.3946

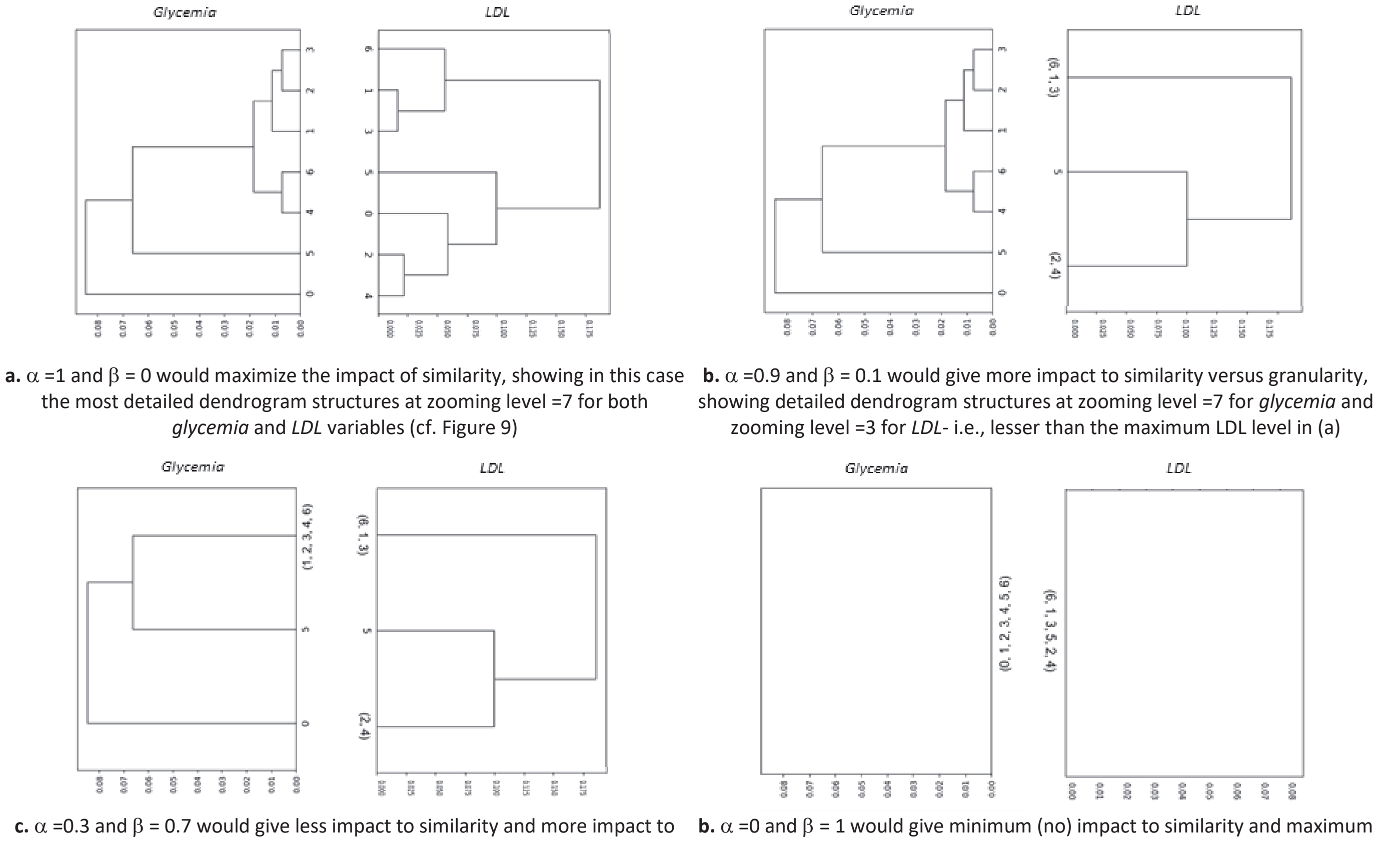
Note that the similarity between the dendrograms at the lowest level (=1) and all the remaining dendrograms is =0 because zoom level =1 is achieved when the number of leaf nodes in the dendrogram is lowest, i.e., =1. This represents the most zoomed-out level where only the root node is displayed in the dendrogram. In other words, a dendrogram at zoom level =1 does not provide any useful information about the data zooming since it shows all the data points grouped together in one big cluster.

3.4.2. Dendrogram Granularity

In addition to maximum dendrogram similarity, our solution recommends the best zooming level to display the dendrograms with the minimum granularity, i.e., minimum amount of information details presented to the user. More formally:

$$\text{Gran}(\text{dend}_1, \text{dend}_2) = \varphi \times \text{Gran}(\text{dend}_1) + \theta \times \text{Gran}(\text{dend}_2) \quad \in [0, 1]$$
$$/ \quad \text{Gran}(\text{dend}_i) = \frac{\#\text{ of leaf nodes}(\text{dend}_i)\text{ -1}}{\#\text{ of data points}(\text{dend}_i)\text{ -1}} \quad \in [0, 1] \qquad \textbf{(9)}$$

where φ, $\theta \in [0, 1]$, and $\varphi + \theta = 1$. A granularity score =1 means that the dendrogram is fully zoomed-in, showing the maximum number of nodes (i.e., the maximum amount of information details). A granularity score = 0 means that the dendrogram is fully zoomed-out, showing the minimum number of nodes =1 (i.e., the root node only, highlighting minimum information details). A lower (higher) granularity score highlights more (less) information details. Table 3.b shows the pair-wise granularity scores between all pairs of dendrograms from our *Glycemia* vs *LDL* running example, considering equal weights for individual granularity scores ($\varphi = \theta = 0.5$). The granularity score between the dendrograms at the lowest levels is =0. The granularity score between the dendrograms at the highest levels =1. The granularity score increases with the zoom level, and decreases accordingly. Table 3.c shows the combined zoomScore values between all pairs of dendrograms from our *Glycemia* vs *LDL* running example.



a. α =1 and β = 0 would maximize the impact of similarity, showing in this case the most detailed dendrogram structures at zooming level =7 for both *glycemia* and *LDL* variables (cf. Figure 9)

b. α =0.9 and β = 0.1 would give more impact to similarity versus granularity, showing detailed dendrogram structures at zooming level =7 for *glycemia* and zooming level =3 for *LDL*- i.e., lesser than the maximum LDL level in (a)

c. α =0.3 and β = 0.7 would give less impact to similarity and more impact to granularity, showing in this case lesser detailed dendrogram structures at zooming level =3 for both *glycemia* and *LDL* variables.

b. α =0 and β = 1 would give minimum (no) impact to similarity and maximum impact to granularity, showing in this case the minimum amount of details for both dendrogram structures at zooming level =0.

Figure 11. *Glycemia* vs *LDL* parallel dendrograms show at different similarity (α) and granularity (β) weight configurations

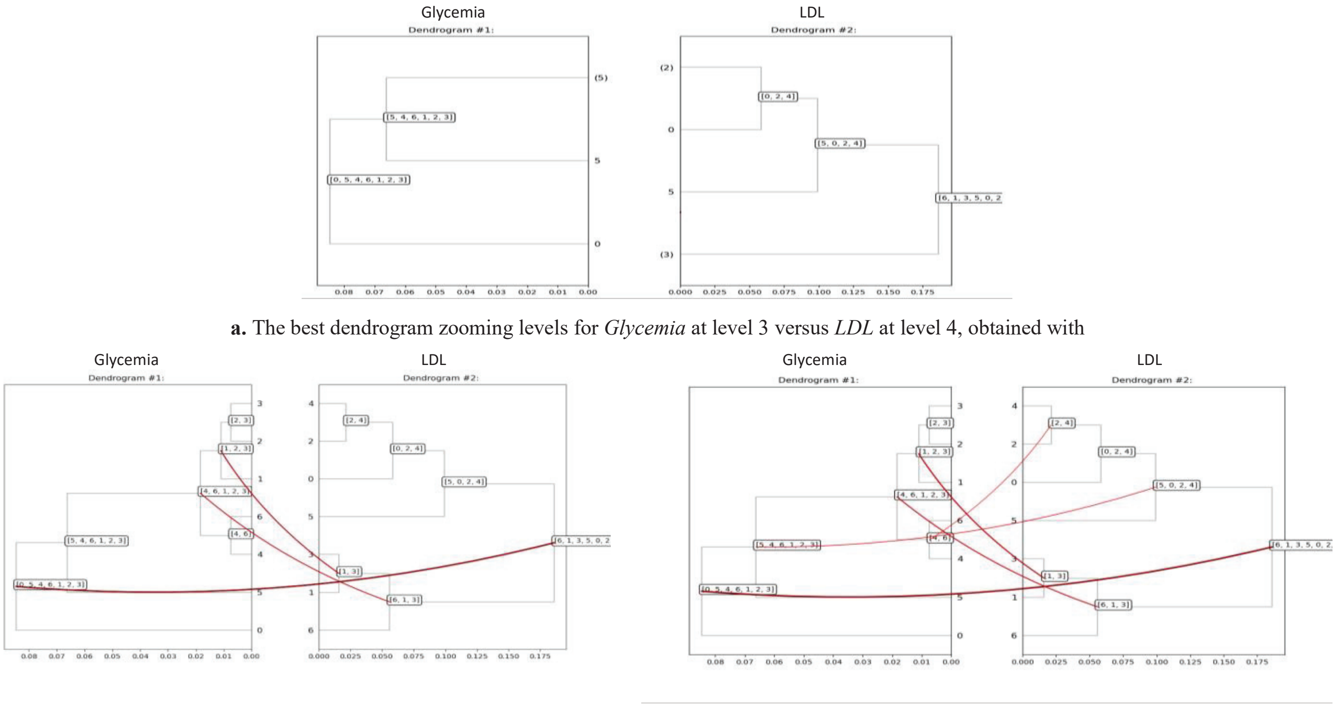
3.4.3. Tuning Similarity and Granularity weights

To produce the optimal zooming level between the mapped dendrograms, our combined zooming score seeks to i) maximize similarity between the dendrograms, and ii) minimize granularity for both dendrograms, while allowing the users to grant more weight to either similarity (by increasing the α weight) or granularity (by increasing the β weight) following their preferences (cf.

Formula 7). Note that weights are combined through a linear weighted sum where α, β ∈ [0, 1] such that α + β = 1. Hence, an increase in similarity weight α incurs a decrease in granularity weight β and vice versa, where α = β = 0.5 provide equal weights to both similarity and granularity scores in computing the dendrogram zooming level. Table 3.a shows the pair-wise similarity scores between pairs of dendrograms produced following our *Glycemia* vs *LDL* running example. In this example, the maximum similarity score is obtained at entry position (7, 7), which represents the similarity score between the dendrogram of zooming level =7 for *Glycemia* and the dendrogram of zooming level =7 for *LDL.* These dendrograms are visually represented in Figure 11.a. Table 3.b shows the pair-wise granularity scores between all pairs of dendrograms from our *Glycemia* vs *LDL* running example. Here the minimum granularity score is naturally obtained at entry position (0, 0), which represents the granularity score for both dendrograms of zooming level =0 (where only the root nodes of the dendrograms are presented, providing the minimum granularity/least amount of data accessible to the user, cf. Figure 11.d). If the user chooses to maximize the impact of dendrogram similarity (by setting α=1) and minimize the impact of granularity (by setting β=0), then the best zooming level would be the most detailed zooming shown in Figure 11.a. If the user wishes to minimize the impact of similarity (by setting α=0) and maximize the impact of granularity (by setting β=1), then the best zooming level would be the least detailed zooming shown in Figure 11.d. Figure 11 shows multiple other configurations of α and β, and how they impact the choice of the best zooming level. As α/β increases/decreases, more/less emphasis will be put on similarity/granularity, depending on the user preferences and the application scenario at hand. Practically, we would like our tool to show the zooming level providing simultaneously: maximum dendrogram similarity (i.e., maximum correlation) and minimal dendrogram granularity (i.e., minimum amount of data) presented to the user.

Note that following several experimental runs (cf. Section 4), we assign in our empirical evaluation a weight α = 0.8 for the dendrogram similarity score and β = 0.2 for the dendrogram granularity score, where the best zooming level dendrograms for our running example are shown in Figure 12.a. Results show that the *Glycemia* dendrogram of level =3 and the *LDL* dendrogrma of level =4 produce the maximum zoomScore value =0.4774, and thus will be returned by the system as the best zooming level to display the dendrograms.

Note that fine-tuning and optimizing the data zooming and the dendrogram granularity weight values can be handled automatically as a multi-objective optimization problem. This can be solved using a number of established solutions that apply machine learning and linear programming to identify the optimal weights for a given problem class, e.g., [27, 54, 73]. The main idea with this family of solutions is to assign a higher (lower) weight with higher (lower) weight, acting like contrast filters in image processing by increasing the contrast on input matrixes. Providing such a capability, in addition to manual tuning, would enable the users to adapt the zooming and granularity levels according to their needs. We do not further address weight value optimization here since it is out of the scope of this study, and we report it to a future study.



a. The best dendrogram zooming levels for *Glycemia* at level 3 versus *LDL* at level 4, obtained with

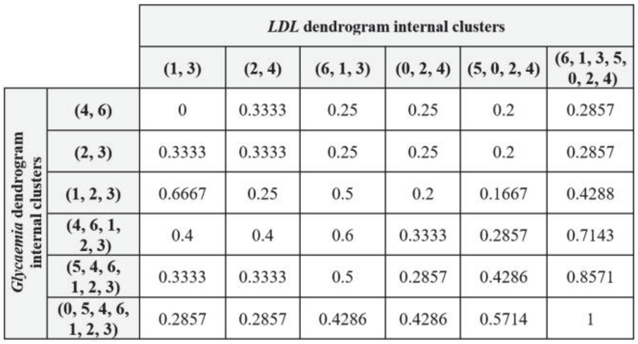
b. Node connections with a threshold value of 0.5

c. Node connections with a threshold value of 0.3

Figure 12. Best zooming (a) and fully zoomed-in visualizations (b, c) of *Glycemia* vs *LDL* parallel dendrograms

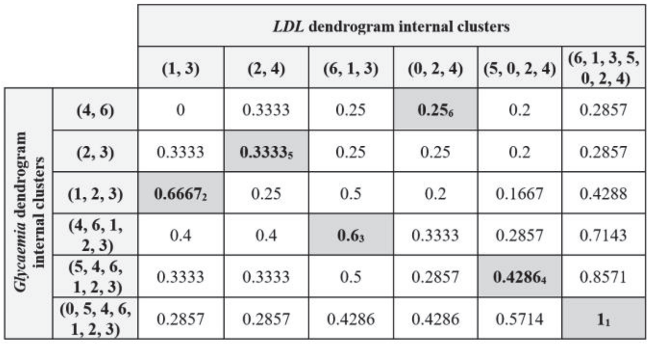
Table 4. Internal nodes similarity matrix for full zoomed-in visualization of *Glycemia vs LDL* parallel dendograms (cf. Figure 12.b, c)

a. Initial similarity matrix



		LDL dendrogram internal clusters					
		(1, 3)	(2, 4)	(6, 1, 3)	(0, 2, 4)	(5, 0, 2, 4)	(6, 1, 3, 5, 0, 2, 4)
Glycaemia dendrogram internal clusters	(4, 6)	0	0.3333	0.25	0.25	0.2	0.2857
	(2, 3)	0.3333	0.3333	0.25	0.25	0.2	0.2857
	(1, 2, 3)	0.6667	0.25	0.5	0.2	0.1667	0.4288
	(4, 6, 1, 2, 3)	0.4	0.4	0.6	0.3333	0.2857	0.7143
	(5, 4, 6, 1, 2, 3)	0.3333	0.3333	0.5	0.2857	0.4286	0.8571
	(0, 5, 4, 6, 1, 2, 3)	0.2857	0.2857	0.4286	0.4286	0.5714	1

b. Result of the transportation problem's minimum cost method, where the order of the iteratively selected cells is shown in subscript



		LDL dendrogram internal clusters					
		(1, 3)	(2, 4)	(6, 1, 3)	(0, 2, 4)	(5, 0, 2, 4)	(6, 1, 3, 5, 0, 2, 4)
Glycaemia dendrogram internal clusters	(4, 6)	0	0.3333	0.25	**0.25_6**	0.2	0.2857
	(2, 3)	0.3333	**0.3333_5**	0.25	0.25	0.2	0.2857
	(1, 2, 3)	**0.6667_2**	0.25	0.5	0.2	0.1667	0.4288
	(4, 6, 1, 2, 3)	0.4	0.4	**0.6_3**	0.3333	0.2857	0.7143
	(5, 4, 6, 1, 2, 3)	0.3333	0.3333	0.5	0.2857	**0.4286_4**	0.8571
	(0, 5, 4, 6, 1, 2, 3)	0.2857	0.2857	0.4286	0.4286	0.5714	**1_1**

3.5. Node Connections

Following the identification of the best zooming level among the paired dendrograms, the remaining step is to connect the internal nodes of the dendrograms in order to highlight their correlation. To achieve this, we compute dendrogram internal node similarity as the similarity between the corresponding clusters, represented as bags of data points. We utilize Jaccard similarity, yet other set similarity measures can be used (e.g., Intersection, Dice). More formally, considering two dendrograms $dend_1$ and $dend_2$, and two internal nodes $x_i \in dend_1$ and $y_j \in dend_j$ being compared:

$$\text{Sim}(x_i, y_j) = \frac{|\text{cluster}(x_i) \cap \text{cluster}(y_j)|}{|\text{cluster}(x_i) \cup \text{cluster}(y_j)|} \in [0, 1] \tag{10}$$

where *cluster*(x_i) and *cluster*($\underline{y_j}$) are the clusters represented by nodes x_i and y_j in their respective dendrograms.

Consequently, we utilize the transportation optimization problem, e.g., [53, 55], to match the related internal nodes from both dendrograms. The transportation problem seeks to associate a number of supply centers *m* (sources) with a number of demand centers *n* (destinations) to optimize supply delivery. In our case, we consider the internal nodes of the first dendrogram to be the supply centers, and the internal nodes of the second dendrogram to be the demand centers. Hence, considering two dendrograms with *m* and *n* internal nodes respectively, we construct an *m*×*n* matrix where the rows represent the internal nodes of the first dendrogram and the columns represent the internal nodes of the second dendrogram. Each entry (*i*, *j*) provides the similarity between internal node x_i from the first dendrogram, and internal node y_j from the second dendrogram. To elaborate the idea, we consider the fully zoomed-in visualization of *Glycemia* vs *LDL* mirrored dendrograms shown in Figure 12.b, c, with zoom level =7 for both dendrograms. We have m-1 = n-1 = 6, resulting in a 6×6 pairwise internal node similarity matrix shown in Table 4.

Once the internal node similarity matrix is produced, we start by matching the nodes together using the transportation problem's *minimum* (*least*) *cost method* widely adopted in the literature, e.g., [53, 55] (other approaches can be used to solve the transportation problem, such as *penalty-based* or *correction-based* methods [8]). In our case, we compute cost as the inverse of similarity, and hence we seek to minimize the cost (i.e., maximize the similarity) among the matching nodes. We briefly describe the process as follows: (i) assign the supply center (internal node from the first dendrogam) with the demand center (internal node from the second dendrogram) having the highest pair-wise similarity, (ii) cross-out the row where the supply center is located, (iii) cross-out the column where the demand center has been satisfied, (iv) repeat iteratively from (i) to assign the remaining units to the feasible allocations until no row or no column is left. Table 4.b shows the result of the transporation problem's minimum cost (maximum simialrity) method, where the order of the selected cells is shown in lowercase. Once all the internal node connections have been established, the system displays all the connections having a similarity score greater than or equal to a (user or system-defined) threshold. Figure 12.b shows the internal node connections having similarity scores above 0.5 (highlighting the nodes sharing more than 50% similarity). Figure 12.c shows more internal connections after lowering the similarity threshold to 0.3. In addition, the thickness of the node connections is defined proportionally to their similarity, where thicker connections highlight more similar nodes.

4. Experimental Evaluation

Our empirical evaluation section is organized as follows. Section 4.1. describes our prototype implementation. Section 4.2. describes our main EHR use case. Section 4.3 describes three semi-structured use cases from online data repositories. Section 4.4 presents our quantitative evaluation study. Section 4.5. presents our qualitative evaluation study, before concluding with a recap discussion of our contributions and results in Section 4.6.

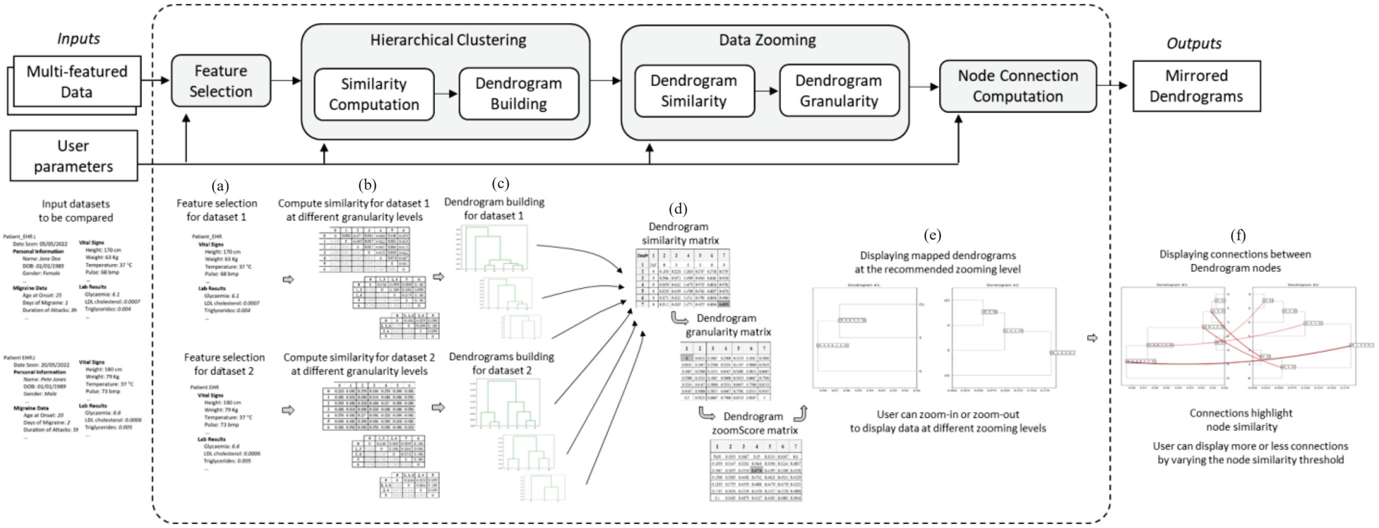

Inputs
Multi-featured Data
User parameters
Feature Selection
Hierarchical Clustering
Similarity Computation
Dendrogram Building
Data Zooming
Dendrogram Similarity
Dendrogram Granularity
Node Connection Computation
Outputs
Mirrored Dendrograms
Input datasets to be compared
(a)
Feature selection for dataset 1
(b)
Compute similarity for dataset 1 at different granularity levels
(c)
Dendrogram building for dataset 1
(d)
Dendrogram similarity matrix
Dendrogram granularity matrix
Dendrogram zoomScore matrix
Feature selection for dataset 2
Compute similarity for dataset 2 at different granularity levels
Dendrograms building for dataset 2
(e)
Displaying mapped dendrograms at the recommended zooming level
User can zoom-in or zoom-out to display data at different zooming levels
(f)
Displaying connections between Dendrogram nodes
Connections highlight node similarity
User can display more or less connections by varying the node similarity threshold

Figure 13. Data flow description of Mirrored Dendrograms tool w.r.t. its overall architecture from Figure 7

4.1. Prototype Implementation

We have implemented our tool using the Python programming language and libraries. Our implementation executes the data flow described in the previous sections (and summarized in Figure 13). Users start by selecting their features of interest to process the datasets being compared (Figure 13.a). The tool then performs similarity computation for each dataset separately according to the previously selected features (Figure 13.b), in order to produce the corresponding dendrogram structures (Figure 13.c). The dendrogram structures are then compared against each other (Figure 13.d), producing an integrated zoomScore matrix allowing to select the best zooming level to displayed the mirrored dendrograms (Figure 13.e). Users can choose to change the zooming level, zooming-in or out of the mirrored dendrograms to visualize more or less details according to their needs. Finally, the tool displays the connections between the dendrogram nodes, highlighting the mirrored dendrograms' structure correlation (Figure 13.f). Users can choose to show more or less connections highlighting more or less of the inner and outer node similarities according to their needs. We perform text preprocessing and feature extraction using *NLTK*, matrix computations using *NumPy*, clustering and dendrogram building using *SciPy*, dendrogram visualization using *MatplotLib*, and GUI functionalities using *Tkinter*. Our implementation is available online[6], along with its source code and test data described in the following sections.

4.1.1. Step-by-Step Walkthrough of the Tool Set-up

First, when the users initialize the tool, a splash screen appears to welcome them. The users are then prompted to choose the dataset to load and process (among the multiple available datasets set-up by the authors of this work) or to upload a new dataset. In the remainder of this paragraph, we will assume the users choose the EHR dataset utilized in our running example (cf. samples in Figure 1, and descried in more detail in the following subsections). Once the dataset is chosen and ready for processing, the users can then choose the EHR properties they wish to compare (cf. Figure 14). This is done through a dedicated interactive form which guides the users through the hierarchical structure of the data. After confirming, a pop-up window appears prompting the users to choose a value for α, which represents the weight that will be given for similarity (versus β for granularity, where $\alpha, \beta \in [0, 1]$ such that $\alpha + \beta = 1$). This will guide the tool in computing the optimum zooming level that would be displayed when the dendrograms are first produced, taking into account the users' similarity versus granularity preferences (cf. discussion in Section 3.4.3). The resulting dendrograms are then produced in a mirrored fashion (cf. Figure 14.c).

4.1.2. User Interactions and Result Visualization

Once the tool is set-up, the users can manipulate many of the visualization elements according to their needs:

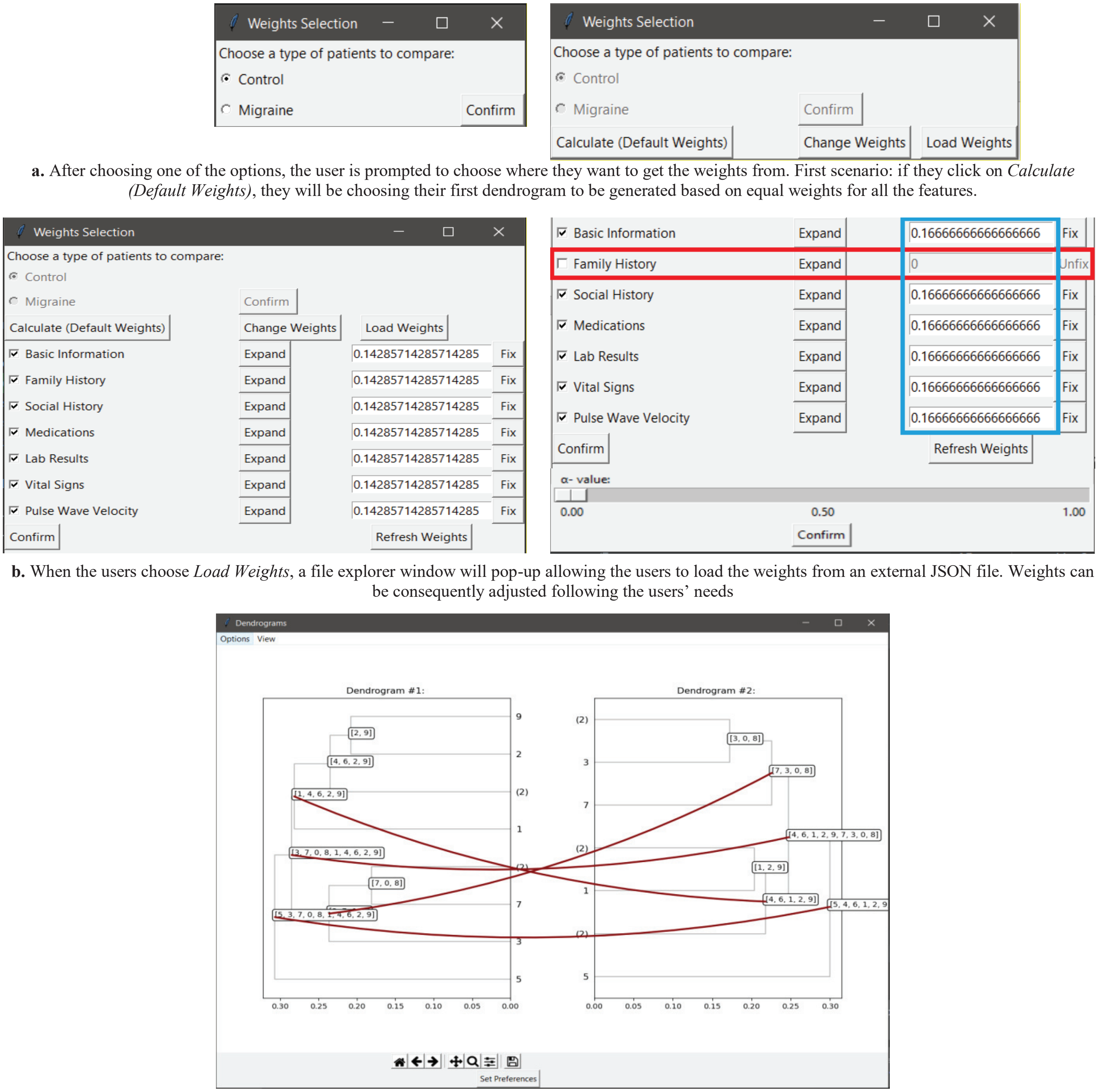
i. **Changing the zooming level:** zooming in/out of a dendrogram means that some of the dendrogram's leaf nodes are grouped together for simplicity. Zooming values reflect the number of total leaf nodes that the users wish to visualize on the *y* axis. In other words, a higher zooming value for a dendrogram means more details will be shown. For instance, Figure 15.a represents mirrored dendrograms with zooming levels (3, 3), while Figure 15.b represents zooming levels (15, 10) (annotations are not displayed for simplicity of presentation).
ii. **Connection type**: links between two similar nodes in the mirrored dendrograms can have one of the following three statuses: 1) *off*: links are hidden from display as shown in Figure 16.1, 2) *default*: a pair of nodes need to have a pair-wise similarity score above 0.5 (i.e., > 50% similarity) in order to draw a connection (Figure 16.b), 3) *threshold*: when selected,

[6] http://sigappfr.acm.org/Projects/MirroredDendrograms/

the user can change the minimum score that two nodes need to have in order to link them. For instance, in Figure 16.b, we choose the threshold value to be =1, which means that two nodes need to be identical to be linked. The user can also adjust the color of the node connections, to distinguish between groups of mapped nodes and their correlated features (cf. Figures 16 and 17).

iii. **Minimum Number of Nodes to Link**: when this button is clicked, a pop-up window will show-up, allowing the users to choose the minimum number of elements in a node to display a link. When links fall below the threshold, they are either hidden totally or replaced by randomly-colored dots, and the links are redrawn upon hovering over the dots.

iv. **Annotation Type:** users can display or hide the annotations for either dendrogram. Here, four options are available: 1) *on*: in both dendrograms, and over each node, an annotation box lists the elements that are grouped by the nodes in display, 2) *left*: annotations are only displayed on the left hand side dendrogram, 3) *right*: annotations are only displayed on the right hand side dendrogram, and 4) *off*: annotation are not displayed. Annotations can also include pictures when available in the reference dataset (cf. Figures 14.a.b and Figure 17).



Weights Selection

Choose a type of patients to compare:

● Control

○ Migraine

Confirm

Weights Selection

Choose a type of patients to compare:

● Control

○ Migraine

Confirm

Calculate (Default Weights) | Change Weights | Load Weights

a. After choosing one of the options, the user is prompted to choose where they want to get the weights from. First scenario: if they click on *Calculate (Default Weights)*, they will be choosing their first dendrogram to be generated based on equal weights for all the features.

Weights Selection

Choose a type of patients to compare:

● Control

○ Migraine

Confirm

Calculate (Default Weights) | Change Weights | Load Weights

☑ Basic Information	Expand	0.14285714285714285	Fix
☑ Family History	Expand	0.14285714285714285	Fix
☑ Social History	Expand	0.14285714285714285	Fix
☑ Medications	Expand	0.14285714285714285	Fix
☑ Lab Results	Expand	0.14285714285714285	Fix
☑ Vital Signs	Expand	0.14285714285714285	Fix
☑ Pulse Wave Velocity	Expand	0.14285714285714285	Fix

Confirm | Refresh Weights

☑ Basic Information	Expand	0.16666666666666666	Fix
☐ Family History	Expand	0	Unfix
☑ Social History	Expand	0.16666666666666666	Fix
☑ Medications	Expand	0.16666666666666666	Fix
☑ Lab Results	Expand	0.16666666666666666	Fix
☑ Vital Signs	Expand	0.16666666666666666	Fix
☑ Pulse Wave Velocity	Expand	0.16666666666666666	Fix

Confirm | Refresh Weights

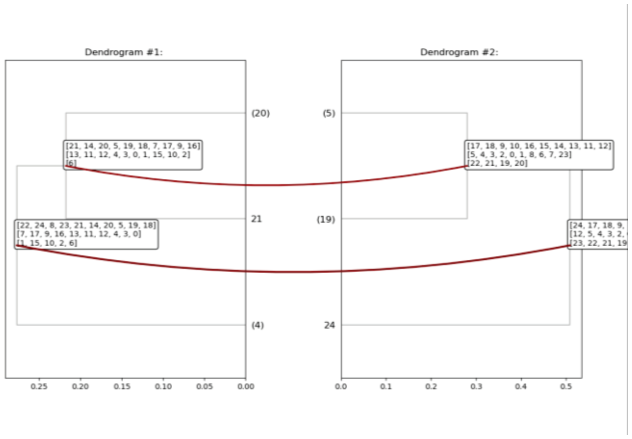
α- value: 0.00 0.50 1.00

Confirm

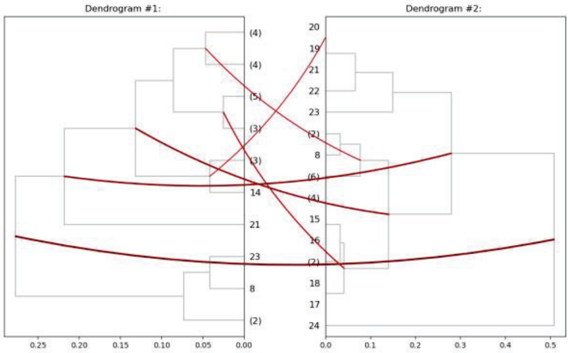
b. When the users choose *Load Weights*, a file explorer window will pop-up allowing the users to load the weights from an external JSON file. Weights can be consequently adjusted following the users' needs

c. Mirrored dendrograms generated following the users' parameter and weight choices

Figure 14. Snapshots of the mirrored dendrogram tool set-up

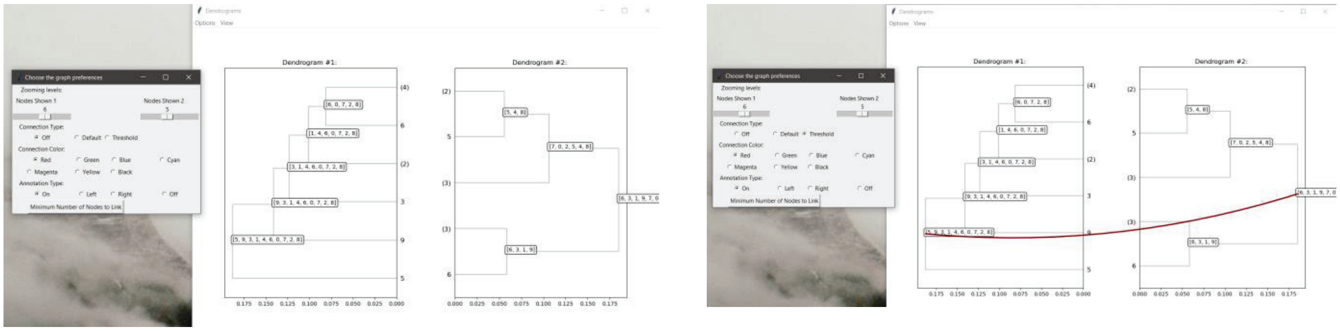


a. Mirrored dendrograms with zooming levels (3,3)



b. Mirrored dendrograms with zooming levels (10, 8)

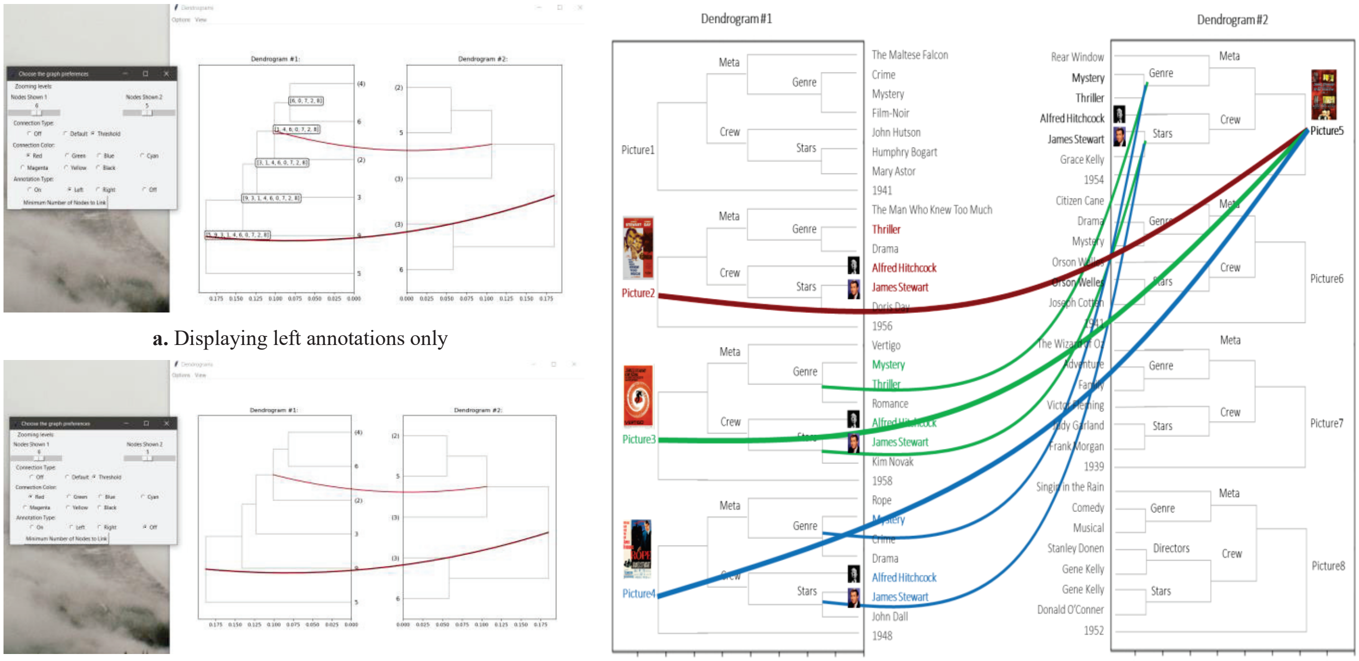
Figure 15. Sample mirrored dendrograms with different zooming levels



a. Node connections off

b. Node connections with threshold =1

Figure 16. Node connections display



a. Displaying left annotations only

b. Displaying no annotations

c. Displaying picture annotations with colored connections (distinguishing groups of node mappings and their correlated features)

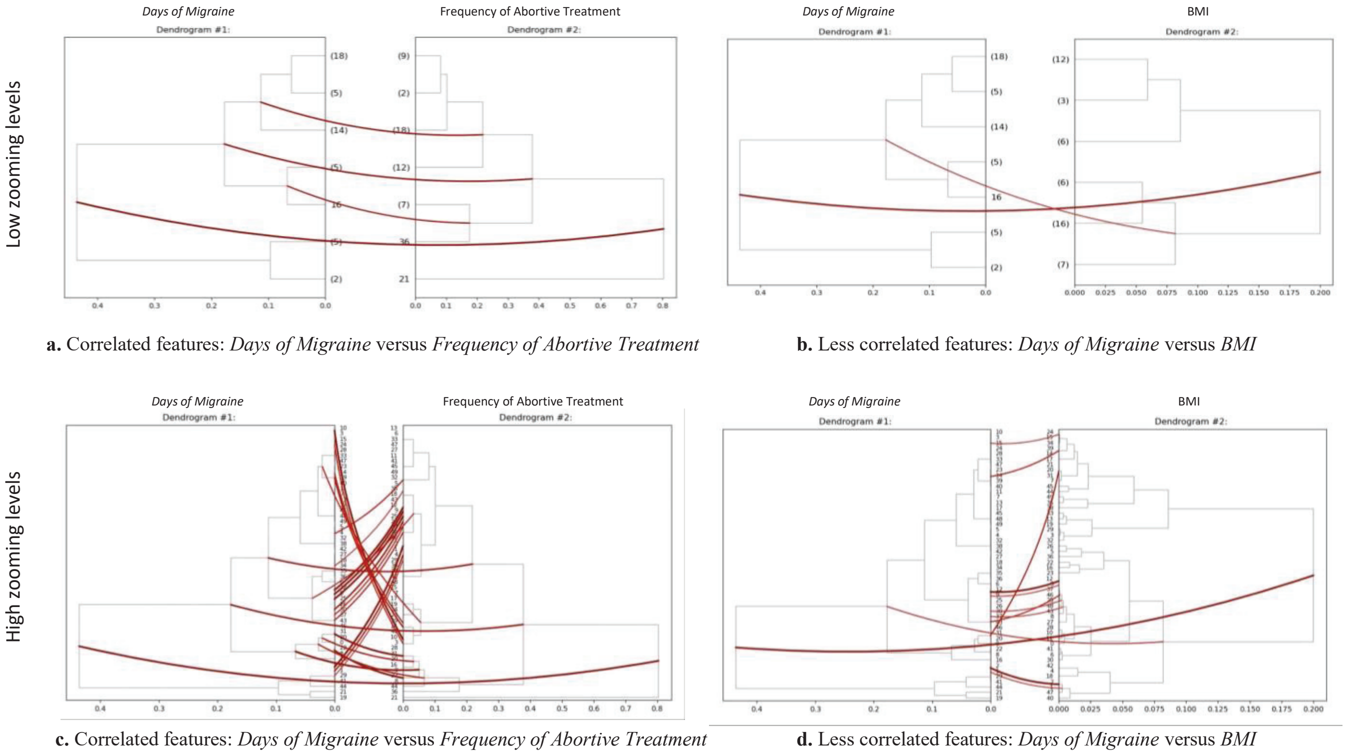
Figure 17. Node annotations display with varying link colors

4.2. EHR Case Study

We used a sample dataset of 114 EHRs of patients who suffer from migraine disorder, obtained from the private medical clinic of Dr. Sola Aoun Bahous, M.D. and professor in the department of internal medicine, division of nephrology, at LAU Rizk hospital. The EHRs were anonymized and vetted by Dr. Bahous. The test protocol was also vetted by Dr. Bahous before conducting the empirical evaluation, and the test results were reviewed and approved by her following the execution of the empirical evaluation. The authors were IRB (Institutional Review Board) exempt since the experiments were conducted through Dr. Bahous' private clinic for the purpose of education research, and following her strict guidance and explicit approval. Sample EHR extracts are shown in Figure 8. We conducted various tests to visualize correlated and uncorrelated features and compare the results with existing visualization tools.

4.2.1. Feature Correlation

In this test, we compare: i) a pair of correlated features: *days of migraine* and *frequency of abortive treatment* having average correlation pcc[7] = 0.5882, and ii) and a pair of less correlated features: *days of migraine* and *BMI*[8] having average pcc = 0.1556. A subset of the data is visualized in Figure 18 with varying zooming levels. Samples were presented to the testers without any manipulation or randomization, clustered following the internal structural properties of the data being compared. Based on the visualizations in Figure 18, we highlight the following observations: i) the mirrored dendrograms in Figure 18.a show similar structures with many connected nodes, reflecting high feature correlation, ii) the mirrored dendrograms in Figure 18.b show less similar structures with only four pairs of connected nodes, reflecting low feature correlation. We obtain similar observations using different zooming levels in Figure 18.c and d. The complete test results with a number of similar use cases are available online.



a. Correlated features: *Days of Migraine* versus *Frequency of Abortive Treatment*

b. Less correlated features: *Days of Migraine* versus *BMI*

c. Correlated features: *Days of Migraine* versus *Frequency of Abortive Treatment*

d. Less correlated features: *Days of Migraine* versus *BMI*

Figure 18. Mirrored dendrogram visualizations for two pairs of sample EHR features considering a subset of 50 patients, shown according to the best zooming levels identified by the tool, with node connection threshold = 0.5

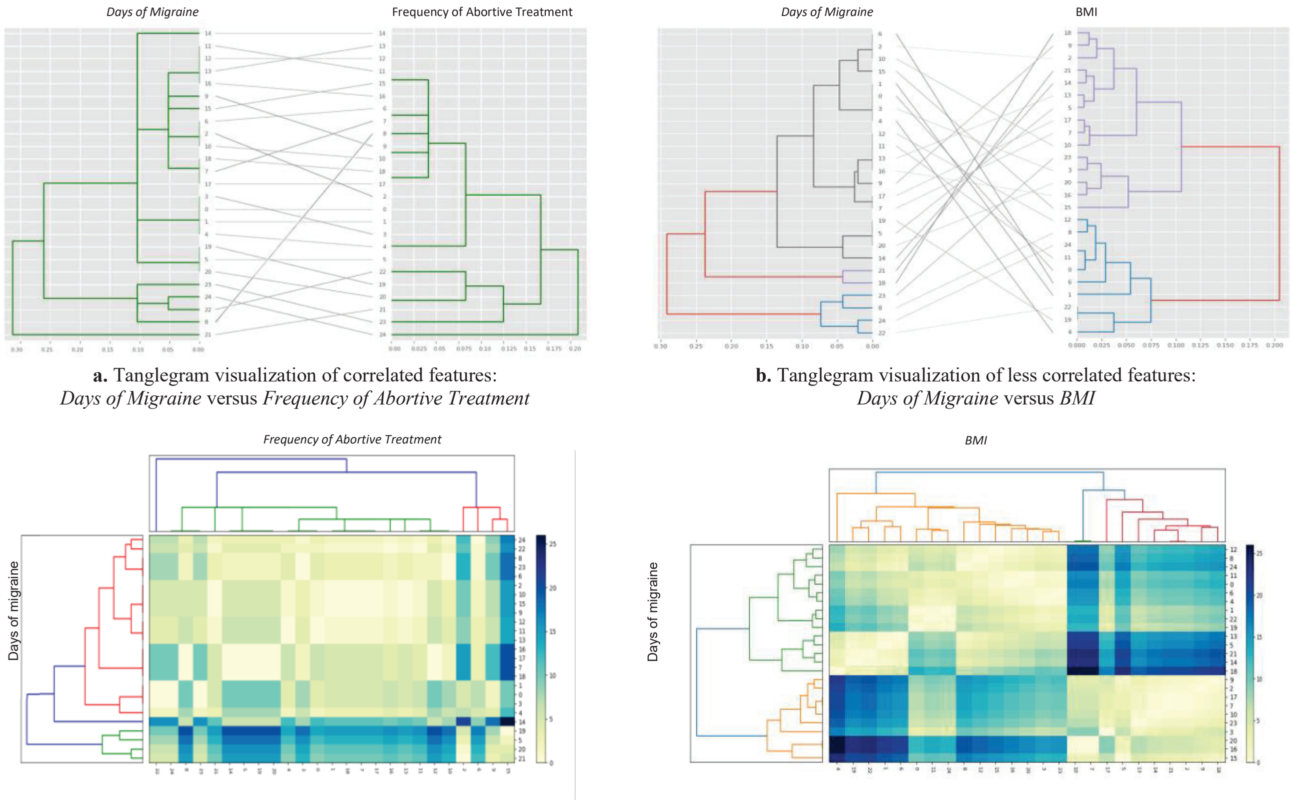
4.2.2. Comparison with Alternative Solutions

In addition, we compare our tool with two alternative visualizations: tanglegram and cluster heatmap. We use the sample dataset and pairs of EHR features from the previous example. Results are shown in Figure 19. While designed to describe the correlations between pairs of dendrograms, yet both tanglegram and cluster heatmap compare dendrograms according to their leaf node mapping, and do not visualize the similarities within the structures themselves. This can be misleading since two dendrograms can have different internal structures, while their leaf nodes are presented in a matching order, and vice versa. This is the case in Figure 19 where both the highly correlated features in Figures 19.a and c and the less correlated features in Figures 19.b and d produce

[7] Pearson Correlation Coefficient

[8] Body Mass Index

similar tanglegram and cluster heatmap visualizations respectively, making it difficult to judge the correlations between the compared features. Different from tanglegram and cluster heatmap, our tool i) computes the similarity between dendrogram structures and maps their internal nodes to describe their structure relationships, ii) allows to zoom-in and out of the data to show their relationships at different granularity levels (compared with existing static solutions), and iii) identifies the best zooming level between the two dendrograms, highlighting the maximum correlation with the minimal amount of details presented to the user.



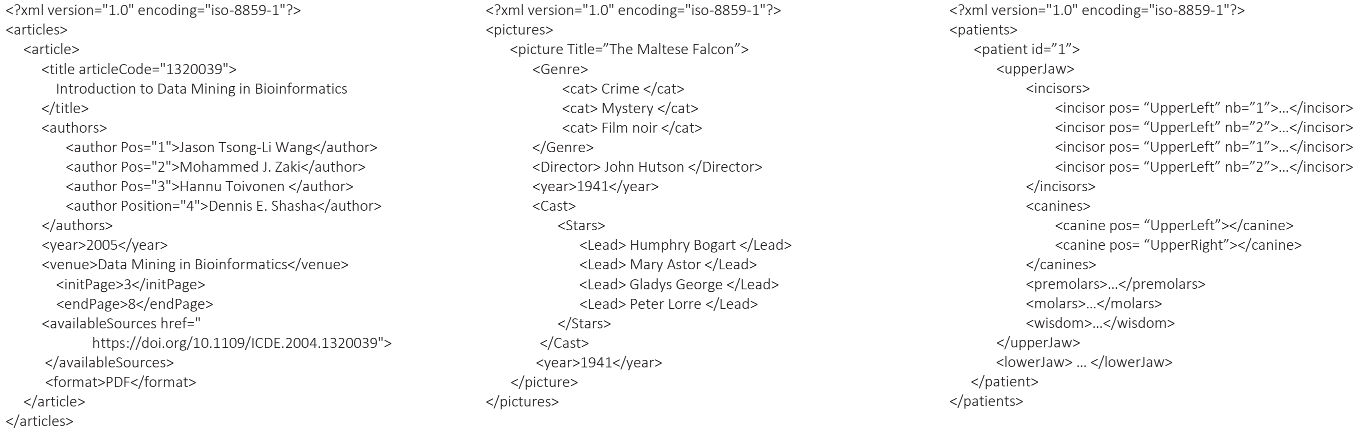
a. Tanglegram visualization of correlated features: *Days of Migraine* versus *Frequency of Abortive Treatment*

b. Tanglegram visualization of less correlated features: *Days of Migraine* versus *BMI*

c. Cluster heatmap visualization of correlated features: *Days of Migraine* versus *Frequency of Abortive Treatment*

d. Cluster visualization of less correlated features: *Days of Migraine* versus *BMI*

Figure 19. Tanglegram and cluster heatmap, visualizatons for two pairs of sample EHR features from Figure 8



```
<?xml version="1.0" encoding="iso-8859-1"?>
<articles>
   <article>
      <title articleCode="1320039">
         Introduction to Data Mining in Bioinformatics
      </title>
      <authors>
         <author Pos="1">Jason Tsong-Li Wang</author>
         <author Pos="2">Mohammed J. Zaki</author>
         <author Pos="3">Hannu Toivonen </author>
         <author Position="4">Dennis E. Shasha</author>
      </authors>
      <year>2005</year>
      <venue>Data Mining in Bioinformatics</venue>
         <initPage>3</initPage>
         <endPage>8</endPage>
      <availableSources href="
               https://doi.org/10.1109/ICDE.2004.1320039">
       </availableSources>
       <format>PDF</format>
   </article>
</articles>
```

a. Sample extract from DBLP

```
<?xml version="1.0" encoding="iso-8859-1"?>
<pictures>
   <picture Title=”The Maltese Falcon”>
      <Genre>
         <cat> Crime </cat>
         <cat> Mystery </cat>
         <cat> Film noir </cat>
      </Genre>
      <Director> John Hutson </Director>
      <year>1941</year>
      <Cast>
         <Stars>
            <Lead> Humphry Bogart </Lead>
            <Lead> Mary Astor </Lead>
            <Lead> Gladys George </Lead>
            <Lead> Peter Lorre </Lead>
         </Stars>
      </Cast>
      <year>1941</year>
   </picture>
</pictures>
```

b. Sample extract from IMDB

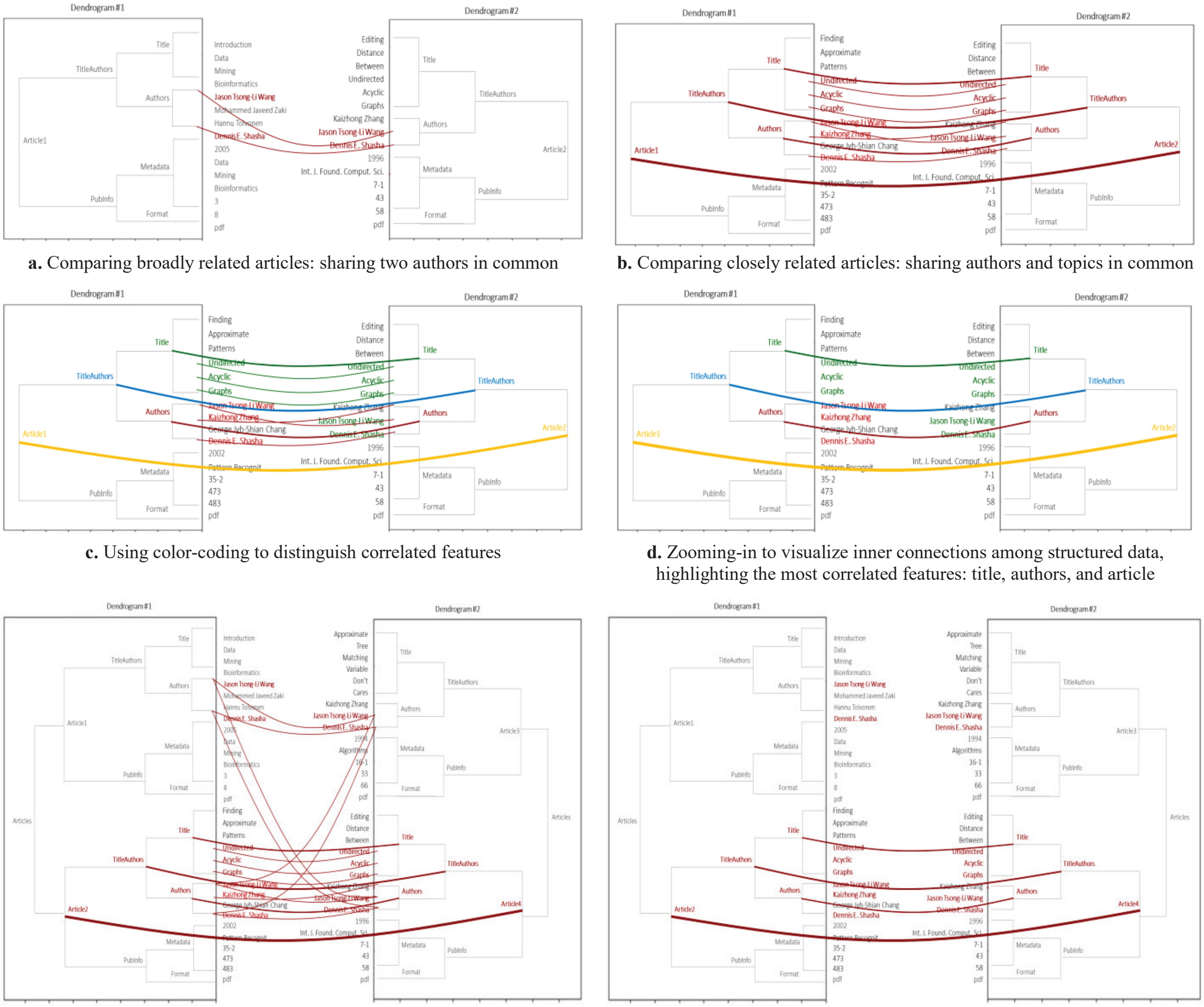
```
<?xml version="1.0" encoding="iso-8859-1"?>
<patients>
   <patient id=”1”>
      <upperJaw>
         <incisors>
            <incisor pos= “UpperLeft” nb=”1”>…</incisor>
            <incisor pos= “UpperLeft” nb=”2”>…</incisor>
            <incisor pos= “UpperLeft” nb=”1”>…</incisor>
            <incisor pos= “UpperLeft” nb=”2”>…</incisor>
         </incisors>
         <canines>
            <canine pos= “UpperLeft”></canine>
            <canine pos= “UpperRight”></canine>
         </canines>
         <premolars>…</premolars>
         <molars>…</molars>
         <wisdom>…</wisdom>
      </upperJaw>
      <lowerJaw> … </lowerJaw>
   </patient>
</patients>
```

c. Sample extract from SSG

Figure 20. Sample semi-structured data from DBMP (a), IMDB (b), and SSG (c)

4.3. Semi-structured Use Cases

Similarly to the EHR case study, different other applications scenarios that can benefit from our visualization tool. These mainly revolve around structured and semi-structured data analysis: our tool allows visualizing the internal connections between any two structured or semi-structured documents or datasets, highlighting their structural feature similarities and internal node mappings. In this context, we evaluate the feasibility and potential of our tool with three use cases, considering sample semi-structured data from: i) DBLP: the computer science bibliography database[9], ii) IMDB: the internet movie database[10], and iii) SSG: semantic SVG graph database[11]. Sample raw documents from each database are shown in Figure 21. We build 40 mirrored dendrogram visualizations from each database, producing a total of 120 visual iterations comparing sample documents against each other to highlight their feature correlations. In the following, Section 4.3.1-4.3.3 provide visual analyses of the different observations made for sample visualizations from each dataset. Then, Section 4.3.4 provides a quantitative evaluation highlighting the impact of our tool in identifying correlating features among the semi-structured data from each use case.



a. Comparing broadly related articles: sharing two authors in common

b. Comparing closely related articles: sharing authors and topics in common

c. Using color-coding to distinguish correlated features

d. Zooming-in to visualize inner connections among structured data, highlighting the most correlated features: title, authors, and article

e. Comparing multiple article entries

f. Zooming-in to highlight the inner feature correlations

Figure 21. Sample mirrored dendrogram visualizations for DBLP documents

4.3.1. DBLP Use Case

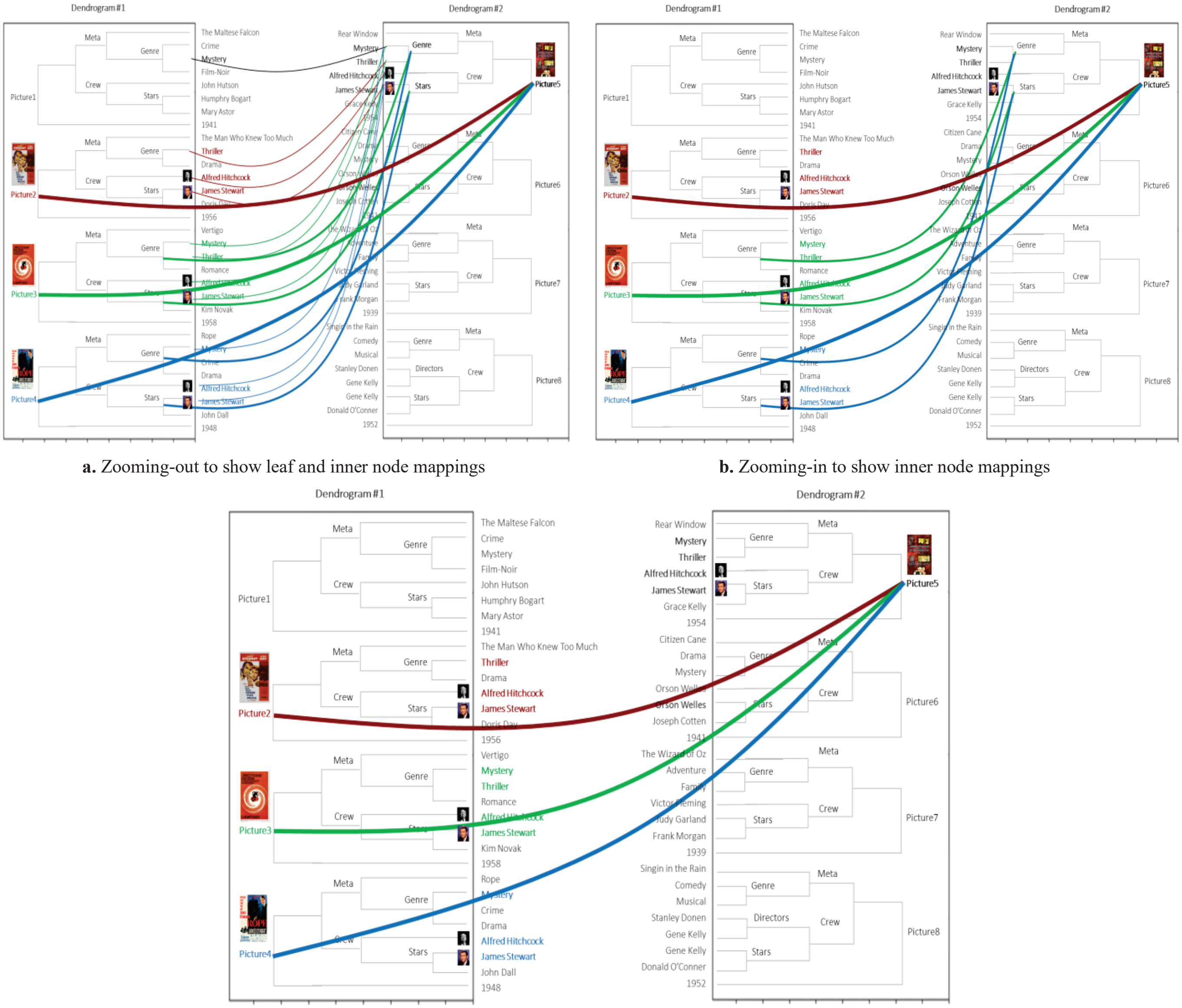
The DBLP database consists of data entries of scientific publications within the area of computer science. A sample DBLP document is shown in Figure 20.a. Sample mirrored dendrogram visualizations of DBLP data are shown in Figure 21. Comparing DBLP data at the leaf node level only allows to connect individual author names, individual keywords, and individual conference names together. While existing visualization tools like tanglegram and cluster heatmap are limited to leaf node mappings, our solution provides useful insights regarding the inner-connections among groups of authors collaborating among each other, as well as papers published by common groups of authors (e.g., research teams). Figure 21.a and b show two pairs of less similar and more similar articles respectively. Articles in Figure 21.a are less similar since they only share few author names in common, whereas

[9] https://www.dblp.org/

[10] https://www.imdb.com/

[11] http://sigappfr.acm.org/Projects/SSG/

articles in Figure 21.b are more similar since they share authors' names and title keywords in common. The difference in similarity is directly apparent by comparing both visualizations. By adding color-codings in Figure 21.c, our tool allows to better identify the structural features involved in the similarity mappings. Also, by zooming-in to the inner features in Figure 21.d, the matching articles become directly apparent through their root node mappings, as well as their upper-most inner node mappings. The importance of inner mappings becomes more inherent when comparing larger numbers of documents. For instance, while different mappings are visible among the leaf nodes in Figure 21.e, nonetheless, zooming-in to the inner nodes in Figure 21.f directly and easily shows the most important mappings highlighting the similarities between the matched documents. Recall that our tool aims at providing users with more useful mapping information among structured data, while spending less time and effort identifying the mappings. We further elaborate this point in our quantitative analysis in Section 4.3.4.



a. Zooming-out to show leaf and inner node mappings

b. Zooming-in to show inner node mappings

c. Zooming-in further to show the most similar movies

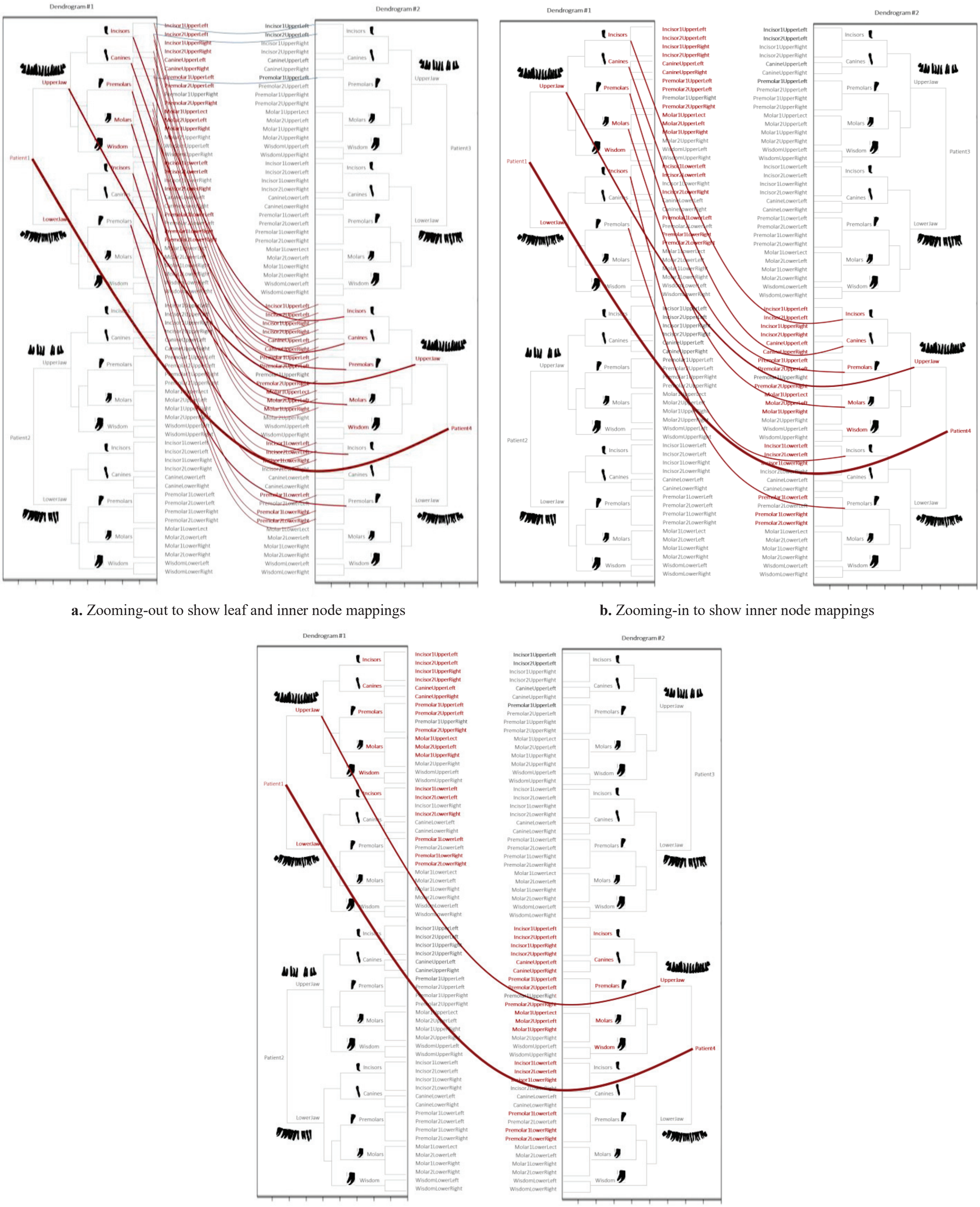
Figure 22. Sample mirrored dendrogram visualizations[12] for sets of four pairs of IMDB documents (comparing four movies)

4.3.2. IMDB Use Case

The IMDB database consists of data entries to describe cinema and TV movies and their crewmembers. A sample IMDB document is shown in Figure 20.b. Sample mirrored dendrogram visualizations of IMDB data are shown in Figure 22. Comparing IMDB data at the leaf node level only allows to connect individual actor and director names, individual movie titles, and individual movie genre types. Nonetheless, connecting inner nodes provides insights regarding the mappings between groups of actors and directors working together (Figure 22.b), and related movies executed by similar crewmembers and targeting similar genres (Figure 22.c). The examples in Figure 22 show that late director *Alfred Hitchcock* and late actor *James Stewart* collaborated on many movies

[12] The latest version of our tool supports visual snippets: showing all snippets (cf. Figure 23), or only snippets of connected data nodes (cf. Figure 22).

together targeting similar genres: *thriller*, *crime*, *mystery*. By zooming-in to focus on the inner node connections, we can clearly see the connections between *Hitchcock* and *Stewart* in Figure 22.b and the strong similarity between their movies in Figure 22.c.



a. Zooming-out to show leaf and inner node mappings

b. Zooming-in to show inner node mappings

c. Zooming-in further to show the most similar patient records

Figure 23. Sample mirrored dendrogram visualizations for sets 2 pairs of SSG documents (comparing two patients)

4.3.3. SSG Use Case

The SSG database consists of data entries to describe panoramic dental x-ray images, including information about patients' teeth positions (*juxtaposed*, *evenly spaced*, etc.), health state (*poorly developed*, *decaying*, etc.), and teeth shapes represented as SVG (Simple Vector Graphics[13]) images. A sample SSG document structure is shown in Figure 20.c. Sample mirrored dendrogram visualizations of SSG data are shown in Figure 23. Comparing SSG data at the leaf node level only allows connecting individual teeth together, according to their property and shape similarities. Connecting inner nodes provides insights regarding the mappings between groups of teeth (e.g., *incisors*, *molars*, *upper jaw*, etc., cf. Figure 23.b) and patients with similar teeth configurations (cf. Figure 23.c). For instance, the examples in Figure 23 show that *patient1* and *patient4* have *incisor*, *canine*, and *premolar* teeth with similar shapes and properties. By zooming-in on the inner node connections, we can see the mentioned patients share similarities among their upper jaw teeth configurations (cf. Figure 23.b) rather than their lower jaw teeth which seem relatively different (Figure 23.c). While the latter information could have been identified by looking at the leaf node mappings only (using existing tools like tanglegram and cluster heatmap), nonetheless, it would have taken more human time and effort to do so, versus looking at the inner connections and zooming-in and out of the inner connections, to easily and directly visualize the structure feature mappings. We further evaluate the time and quality gains through our visualization tool in the following section.

4.4. Quantitative Study

4.4.1. Test Metrics

In addition to the visual observations above, we conducted a quantitative evaluation to better assess the quality of our tool and its limitations. We evaluate visual quality using two metrics: i) the time needed by a user to identify the matching features between two data entries, and ii) the accuracy of the mapped features identified by the user. The time metric indicates how much time a user needs to spend assessing the visualization in order to understand and identify the mapped features: the more time spent, the lesser the quality of the visualization tool. The accuracy metric indicates the quality of the mapped features as identified by the user: the higher the number of accurate mappings detected, the better the quality of the visualization tool. More formally:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP+TN+FP+FN}} \quad \in [0, 1] \tag{11}$$

where *TP*, *TN*, *FP*, and *FN* designate true positives, true negatives, false positives, and false negatives [52, 63]. In this experiment, we evaluate accuracy w.r.t. the root node mappings, i.e., we target: i) DBLP's *article* root node mappings, ii) IMDB's *picture* root node mappings, and iii) SSG's *patient* root node mappings. For instance, considering DBLP:

- *TP* denotes the number of mappings between pairs of *article* root nodes that should be mapped together indeed,
- *TN* denotes the number of unmapped pairs of *article* root nodes that should not be mapped together indeed,
- *FP* denotes the number of mappings between pairs of *article* root nodes that should not have been mapped (miss-mapped),
- *FN* denotes the number of unmapped pairs of *article* root nodes that should have been mapped together (missed mappings).

Table 5. Summary description of test data properties

For each side of the visualization	DBLP				IMDB				SSG			
# of pairs of docs (root nodes)	1	3	6	10	1	3	6	10	1	3	6	10
avg. # of nodes (inner and leaf)	22	65	138	232	22	63	129	227	37	111	222	370
avg. # of leaf nodes	15	48	94	164	14	47	84	152	32	96	192	320

4.4.2. Test Data

We built 40 mirrored dendrogram visualizations from each database, consisting of 10 visualizations made of 1 pair of documents each (e.g., 1 pair or *articles* from DBLP, or 1 pair of movie *pictures* from IMDB, or 1 pair of *patients* from SSG, cf. sample visuals in Figure 21), 10 visualizations made of 3 pairs of documents each (cf. samples in Figure 22 with 4 pairs of documents each), 10 visualizations made of 6 pairs of documents each, and 10 visualizations made of 10 pairs of documents each; totaling a number of 120 visualizations. We also built the corresponding visualizations using tanglegram and cluster heatmap, in order to perform a comparative evaluation study. The test data properties are summarized in Table 5.

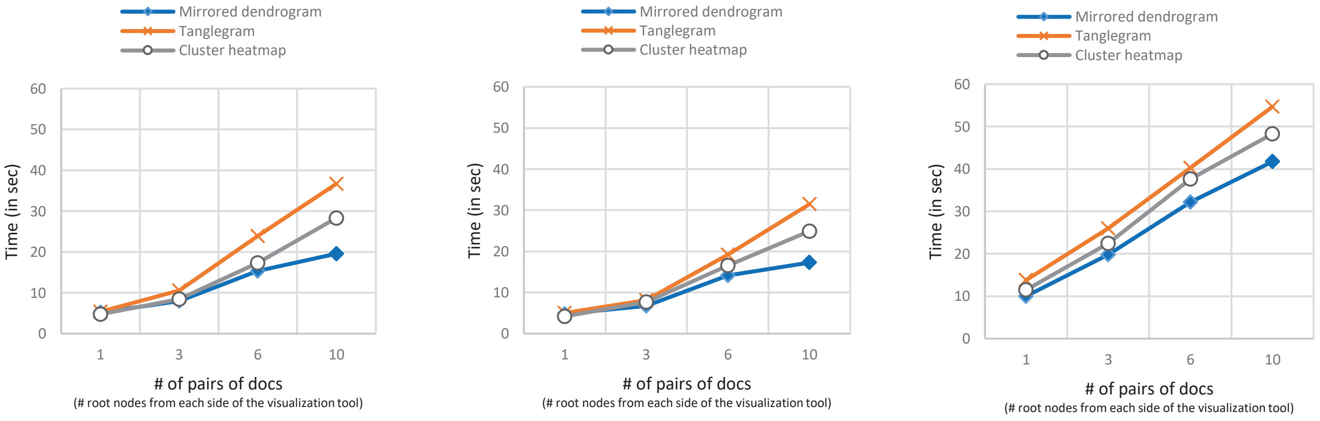
4.4.3. Test Subjects

A total of 40 human testers were invited to contribute to the experiment, where every tester independently processed every visualization in order to identify the root node mappings. Testers were senior computer engineering students following the senior author's technical elective course[14]. Testers were initially shown a demo of the mirrored dendrogram, tanglegram, and cluster

[13] https://www.w3.org/TR/SVG2/

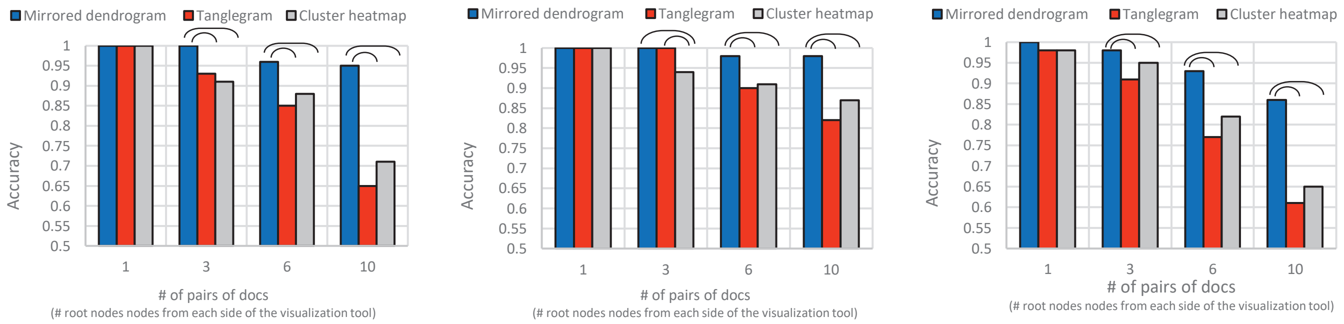
[14] Testers in this experiment consisted of author Joe Tekli's fifth year senior engineering students. They were motivated to conduct the empirical evaluation as part of their course classwork duties in the context of their technical elective course. This exercise was voluntary, with bonus points provided to the participants as part of their classwork.

heatmap tools, providing them with sample visualizations for every tool. Testers were then instructed to examine the leaf node connections (for each of the three compared tools) and the inner node connections (provided through the mirrored dendrograms, and using the zoom-in and zoom-out functionalities) for the purpose of identifying the root node mappings. Testers were asked to evaluate 10 random visualization tasks each, where every visualization task consisted of a mirrored dendrogram, a tanglegram, and a cluster heatmap describing the same pairs of data entries. The experiment was conducted live, during class time, where every tester could stop the evaluation and submit the results at any stage of the process, to ensure consistency and high confidence in the results. Visualizations were sequentially shuffled and shared with the testers in a round-robin fashion to ensure an even distribution of responses. In total, 400 responses were obtained, with every visualization task receiving over 3 responses, the average of which were reported in the test results. In addition, we utilize two-sample t-tests[15] to verify that the means of sample data groups being compared are statistically different from each other[16] [48]. Note that the limited size of our compared samples does not allow us to conduct a more detailed statistical analysis. Yet we believe that the results obtained in this work provide valuable insight and highlight interesting observations that need to be later statistically generalized using a larger number of tests.



a. DBLP time results **b.** IMDB time results **c.** SSG time results

Figure 24. Average user time in identifying the mapped root node features between two data entries



d. DBLP accuracy results **e.** IMDB accuracy results **f.** SSG accuracy results

Figure 25. Average user accuracy (d-f) in identifying the mapped root node features between two data entries, and their t-test indicators. Horizontal arc shaped indicators are added to the graphs to highlight the results of t-tests conducted on every pair of sample groups: they identify group averages (means) which are not statistically different from each other (cf. Table 6).

[15] We utilize *two-sample t-tests assuming unequal variances*, since our samples come from different groups of test cases (DBLP, IMDB, and SSG).

[16] The *t-test* evaluates whether the arithmetic means of two groups of data points are *statistically* different from each other. Two groups seem to be different or distinct from each other when their means are different and their standard deviations are low (i.e., low variability), meaning the groups share little (if any) overlap among their data curves. In contrast, the same difference between group averages becomes less sticking when comparing groups having high standard deviations (i.e., high variability), meaning the groups' data curves might heavily overlap [48].

Table 6. P-values computed following two-sample t-tests conducted on the compared data, considering the three test datasets combined. P-values < 0.05 (showcasing statistically significant *mean* differences) are highlighted in bold[17].

a. P-values - # pairs of docs (root nodes) = 1

	Mirrored Dendrogram	Tanglegram	Cluster heatmap
Mirrored dendrogram		0.3739	0.3739
Tanglegram	0.3739		1
Cluster heatmap	0.3739	1	

b. P-values - # pairs of docs (root nodes) = 3

	Mirrored Dendrogram	Tanglegram	Cluster heatmap
Mirrored dendrogram		0.1719	**0.0121**
Tanglegram	0.1719		0.6779
Cluster heatmap	**0.0121**	0.6779	

c. P-values - # pairs of docs (root nodes) = 6

	Mirrored Dendrogram	Tanglegram	Cluster heatmap
Mirrored dendrogram		0.1719	**0.0121**
Tanglegram	0.1719		0.6778
Cluster heatmap	**0.0121**	0.6778	

d. P-values overall – # pairs of docs (root nodes) = 1, 3, 6, and 10

	Mirrored Dendrogram	Tanglegram	Cluster heatmap
Mirrored dendrogram		**0.0199**	**0.0206**
Tanglegram	**0.0199**		0.7428
Cluster heatmap	**0.0206**	0.7428	

4.4.4. Test Results

Test results are compiled in Figure 24. We highlight the following observations regarding *mapping time* (cf. Figure 24.a-c):

- The three compared visualization tools required almost equal time when mapping a reduced number of data nodes (i.e., with # of root nodes from each size of the visualization tool <=3).
- Mirrored dendrogram consistently produced minimum mapping time with the increase in the number of data nodes being compared (i.e., # of root nodes from each side of the visualization tool >3), compared with its two counterparts. Based on our observations and discussions with the human testers, this is mainly due to i) the existence of mappings between inner nodes, and ii) the easiness of zooming-in and zooming-out of the inner node mappings. Most human testers found the inner node mappings and zooming functionality very useful in easily identifying the root node mappings among the compared documents.
- Clustered heatmap consistently produced lesser mapping time, compared with its tangelegram counterpart. Based on our discussions with the testers, the clustered heatmap color-coding helped them identify correlated regions faster than the leaf node connections of the tanglegram.

We also highlight the following observations regarding *mapping accuracy* (cf. Figure 24.d-f):

- The three visualization tools produced almost equal and comparable accuracy levels when mapping a reduced number of data nodes (i.e., # of root nodes from each size of the visualization tool <=3). This is also highlighted through the corresponding two sample t-tests where the obtained p-values are > 0.05, meaning that the measured pair-wise group averages between mirrored dendrogram/tanglegram, mirrored dendrogram/cluster heatmap, and tanglegram/cluster heatmap are not statistically different from each other (cf. Table 6.a).
- Mirrored dendrogram consistently produced the best accuracy levels with the increase in the number of data nodes being compared (i.e., with # of root nodes from each side of the visualization tool >3), compared with its two counterparts. This is also highlighted through the corresponding two sample t-tests, where the obtained p-values between mirrored dendrogam/cluster heatmap (for # of pairs of documents = 3 and 6) and mirrored dendrogram/tanglegram (for the overall combined # of pairs of documents = 1, 3, 6, and 10) become < 0.05, meaning that the measured pair-wise group averages between mirrored dendrogram and its two alternatives become statistically different from each other with the increase of the # of pairs of documents being processed (cf. Table 6. b, c, and d).
- Similarly to mapping time, most testers concur that inner node mapping and zooming functionalities available through mirrored dendrogram allowed them to better identify the root node mappings, compared with the other two visualization tools which only provide leaf node mappings.
- Clustered heatmap consistently performed better than tanglergram in terms of accuracy. Discussions with testers revealed that color-coding the leaf node connections was helpful in identifying the correlating areas within the two structured being compared, and consequently identifying the mapped root nodes.

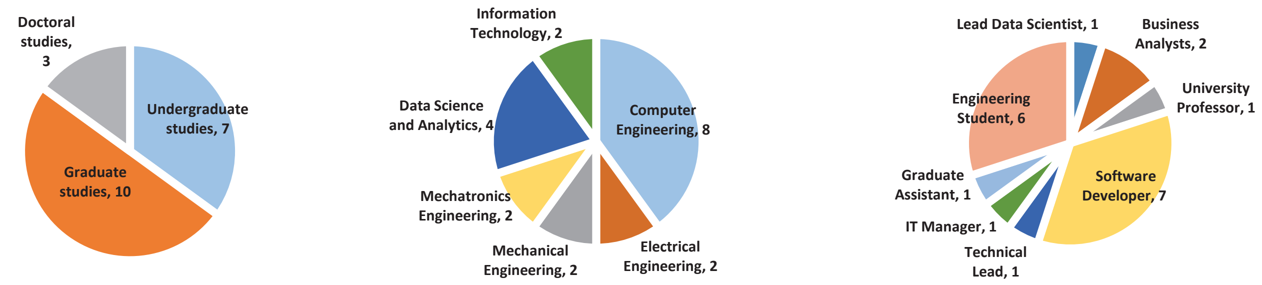
Discussion: The above results highlight the quality of our tool in identifying mappings among structured data, and reducing mapping time. Also, we note that average accuracy levels between mirrored dendrogam and its alternatives become more statistically different with the increased size of the compared datasets (i.e., with the increased # of pairs of documents being processed), as underlined by the t-test p-values in Figure 25 and Table 6. Results also highlight the potential of cluster heatmap as a consistent first runner-up across both experiments. On the one hand, testers mostly agree on the usefulness of inner node mapping and zooming functionalities to better understand the structural mappings and correlations between structured data. On the other hand, testers also agree on the usefulness of cluster heatmap's color-coding of the leaf node mappings, which allowed them to

[17] On the one hand, p-values < alpha (=0.05) imply that the t-test hypothesis is rejected, i.e., the compared *means* are different. On the other hand, p-values > alpha (=0.05) imply that the t-test hypothesis stands, i.e., that the compared *means* are the same, and their difference is not statistically significant.

identify the root node mappings faster and more effectively. Hence, our upcoming work will focus on leveraging the color-coding of cluster heatmap, to enhance the visualization of data mappings within mirrored dendrogram.

4.5. Qualitative Study

Since our work involves visualizations perceived by users, we acquired and evaluated the feedback from human testers to assess both i) the visualization quality and ii) the ease of use of our visualization tool. A total of 20 testers were invited to contribute to the experiment, where they independently filled two surveys: one targeting visualization quality, and another targeting ease of use. Testers were undergraduate and graduate engineering students, as well as junior and senior engineers with background in data science, business analytics, computer science, or computer engineering (cf. Figure 26). Invitation emails were shared by the authors and broadcast to their undergraduate and graduate engineering students and alumni. The first 20 testers who accepted the invitation volunteered to conduct the surveys and did not receive any compensation. Testers were initially shown a demo of the mirrored dendrogram, tanglegram, and cluster heatmap tools, providing them with sample visualizations for every tool. Testers were also invited to use the tools on three small data samples provided by the authors, to familiarize with their visualizations and functionality, including the inner node connections and zooming functionalities provided by mirrored dendrograms.



a. Educational level

b. Field of study

c. Profession

Figure 26. Testers' education level, and field of study, and professions

4.5.1. Visualization Evaluation

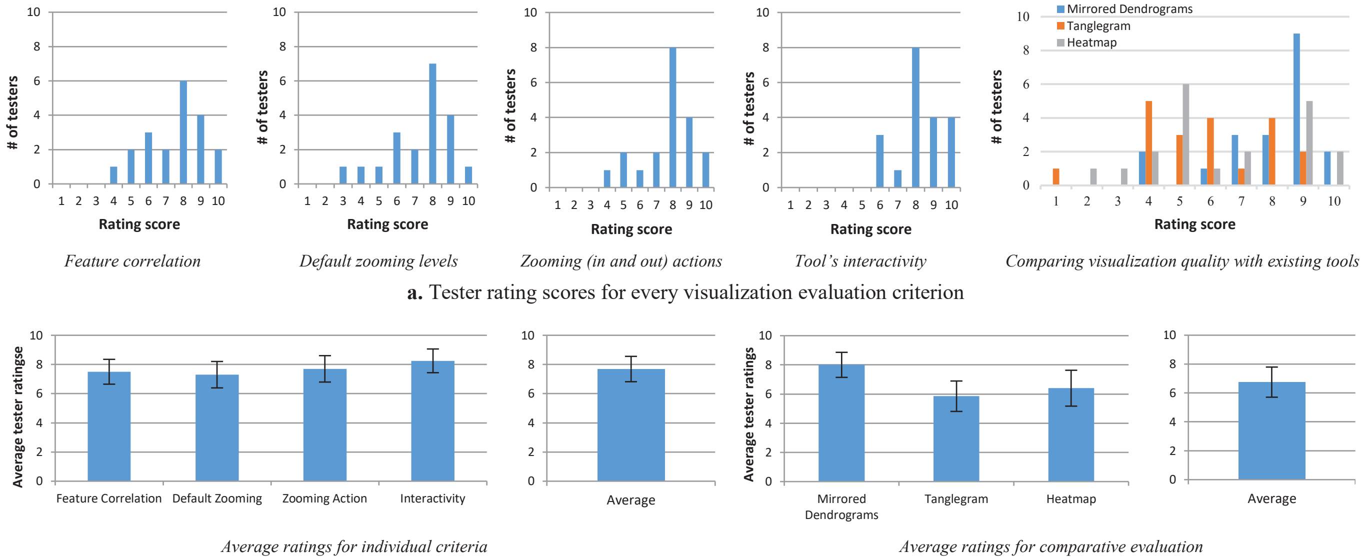
For the purpose of this experiment, we created an online survey[18] considering five evaluation criteria: i) feature correlation visualization, ii) default zooming levels, iii) zooming in and out actions, iv) tool's interactive functionalities, v) comparison with existing solutions (cf. Table 7). Every evaluation criterion was rated on a on a Likert scale from 1 to 10 (i.e., from *highly dissatisfied* to *highly satisfied*). Results in Figure 27 show the compiled tester ratings, as well as the average rating scores and their standard deviations aggregated for every criterion. The top scoring criterion is *tools' interactivity*, where 65.8% of the testers gave it scores ≥ 7/10, achieving an average overall score of 8.3/10 (stdev = 1.3). The bottom scoring criterion is *default zooming*, where 68.4% of the testers gave this criterion scores ≥ 7/10, achieving an average overall score of 7.3/10 (stdev = 1.8). *Comparative evaluation* results show that 84.2% of the testers gave the mirrored dendrograms rating scores ≥ 7/10, compared with 36.8% and 47.4% for tanglegram and cluster heatmaps respectively. The mirrored dendrograms achieved an average rating of 8 (stdev = 1.7), compared with 5.85 (stdev = 2) and 6.4 (stdev = 2.5) for tanglegram and cluster heatmaps respectively. Considering all criteria combined, results produce an average overall rating score of 7.75/10 (stdev = 1.73), highlighting the overall visualization quality of the tool according to most testers. In addition, we compute Cronbach's alpha ($\in [0, 1]$) as a measure of internal consistency, evaluating how closely related the ratings of the different criteria are as a group (higher values indicate higher agreement between the criteria) [16]. The purpose of this measure is to study both the relatedness and the distinctiveness of the evaluation criteria's results [16], where very low scores (close to 0) indicate that the criteria are completely unrelated and thus inconsistent together, while extremely high scores (close to 1) indicate that the criteria are almost identical and can be substituted by each other. The goldilocks zone for Cronbach's alpha is usually $\in [0.7, 0.95]$, meaning the criteria are correlated enough to be consistent together, without overlapping each other [32]. Results for the above five visualization criteria combined produce Cronbach's alpha = 0.84, highlighting high consistency, i.e., high correlation while maintaining the distinctiveness of the criteria.

In summary, results show that most testers are satisfied with the visualization tool: i) underlining its quality in describing feature correlations, ii) suggesting a default zooming level to compromise between maximum correlation and minimal amount of details presented to the user, iii) zooming-in and out the data to visualize data samples and their cluster hierarchies at different levels of details, iv) providing several interactive capabilities allowing users to set their preferences according to their needs, and v) providing improved visualizations compared with existing solutions.

[18] Available at: https://github.com/akf98/mirrored-dendrogram-tool

Table 7. Tool's visualization evaluation criteria

Criterion	Description	Evaluation question
1. Feature correlation visualization	It is the ability of the tool to allow users to visually distinguish between highly correlated features and loosely correlated features, when mirrored against each other.	Given the criterion's description, how satisfied are you with the feature correlation visualization of the tool?
2. Default zooming levels	It refers to the quality of the default zooming levels suggested by the tool, highlighting the maximum correlation with the minimal amount of details presented to the user.	Given the criterion's description, how satisfied are you with the tool's default zooming levels?
3. Zooming in and out actions	It describes how efficient it is to zoom in and out of the data, and navigate up and down the dendrogram hierarchies.	Given the criterion's description, how satisfied are you with the zooming actions of the tool?
4. Tool's interactivity	It refers to the capacity of tool to provide interactive functionalities, including parameter settings, similarity thresholds, node and edge visualizations and coloring, among others.	Given the criterion's description, how satisfied are you with the tool's interactive functionalities?
5. Comparing visualization quality with tools	It refers to the quality of the tool's visualization compared with existing solutions: namely tanglegram and cluster heatmap.	Given the criterion's description, how satisfied are you with the tool's visualization quality compared with existing solutions?



Feature correlation *Default zooming levels* *Zooming (in and out) actions* *Tool's interactivity* *Comparing visualization quality with existing tools*

a. Tester rating scores for every visualization evaluation criterion

Average ratings for individual criteria *Average ratings for comparative evaluation*

b. Average tester ratings

Figure 27. Tester ratings for the visualization evaluation criteria

4.5.2. Usability Evaluation

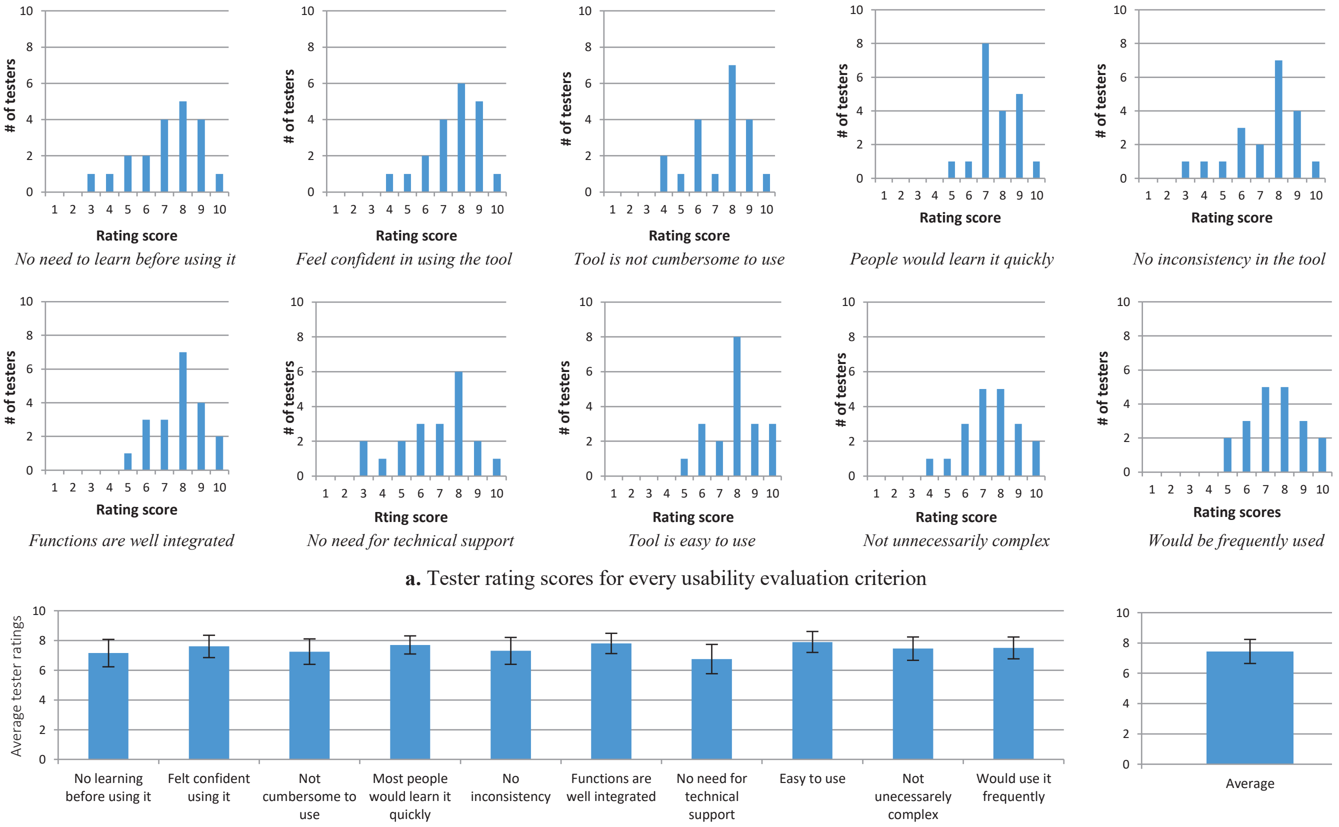
In addition to the visualization evaluation study described in the previous sub-section, we acquired feedback from the human testers to assess the usability of our visualization tool. To do so, we created an online survey[19] considering ten questions included in a typical SUS (System Usability Scale) template [67]: i) no (minimal) learning is required before using the tool, ii) feeling confident in using the tool, iii) tool is not cumbersome to use, iv) most people would learn it quickly, v) no inconsistency in the tool, vi) functions are well integrated, vii) no need for technical support while using the tool, viii) tool is easy to use, ix) tool is not unnecessarily complex, and x) tool would be frequently used (cf. Table 8). Every evaluation criterion was rated on a on a Likert scale from 1 to 10 (i.e., from *strongly disagree* to *strongly agree*). Results in Figure 28 show the compiled tester ratings, as well as the average rating scores and their standard deviations aggregated for every criterion.

The top scoring criterion is *ease of use*, where 80% of testers gave it scores ≥ 7/10, achieving an average of 7.9/10 (stdev = 1.41). The bottom scoring criterion is *technical support is not needed*, where 65% of testers gave this criterion scores ≥ 7/10, with an average of 6.75/10 (stdev = 1.97). Considering all criteria combined, results produce an average overall rating score of 7.44/10 (stdev = 1.59), highlighting the overall usability of the tool according to most testers. In addition, we compute Cronbach's alpha ($\in [0, 1]$) as a measure of internal consistency, evaluating how closely related the ratings of the different criteria are as a group. Results for the above ten usability criteria combined produce Cronbach's alpha = 0.93, highlighting high correlation while maintaining some distinctiveness among the criteria. Note that Cronbach's alpha for the usability criteria is clearly higher than its visualization counterpart (i.e., 0.84, cf. Section 4.4.1), due to the closer relationship between the usability criteria themselves, which is evident in their definitions (e.g., criterion 1 - *no* (*minimal*) *learning is required before using the tool* versus criterion 4 - *people would learn it quickly*, and criterion 8 - *tool is easy to use* versus criterion 9 - *tool is not unnecessarily complex*, cf. Table 8), where correlated ratings are expected.

[19] Available at: https://github.com/akf98/mirrored-dendrogram-tool

Table 8. Tool's usability evaluation criteria

Criterion	Description	Evaluation question
1. No (minimal) learning is required	I did not need to learn many things before I properly used the tool	Do you need to learn many things before properly using the tool?
2. Feel confident in using the tool	I felt very confident using the tool	Do you felt confident using the tool?
3.Tool is not cumbersome to use	I did not find the tool cumbersome to use	Do you find the tool cumbersome to use?
4. People would learn it quickly	I would imagine that most people would learn to use this tool quickly	Do you think that most people would learn to use this tool quickly?
5. No inconsistency in the tool	I did not think there was much inconsistency in the tool	Do you think there is much inconsistency in the tool?
6. Functions are well integrated	I found the various functions in the tool were well integrated	Do you find the various functions in the tool are well integrated?
7. No need for technical support	I did not need the support of a technical person to use the tool	Do you need the support of a technical person to use the tool?
8. Tool is easy to use	I thought the tool was easy to use	Do you think the tool is easy to use?
9. Tool is not complex	I did not find the tool unnecessarily complex	Do you find the tool unnecessarily complex?
10. Tool would be frequently used	I think I can use this tool frequently	Do you think you would use this tool frequently?



No need to learn before using it *Feel confident in using the tool* *Tool is not cumbersome to use* *People would learn it quickly* *No inconsistency in the tool*

Functions are well integrated *No need for technical support* *Tool is easy to use* *Not unecessarily complex* *Would be frequently used*

a. Tester rating scores for every usability evaluation criterion

b. Average tester ratings

Figure 28. Tester ratings for the usability evaluation criteria

4.5.3. Qualitative Result Review Session

Following the collection and analyses of the surveys' results, an online session was organized with the testers to share and discuss the obtained results. The session started by reminding the testers of the two surveys conducted to evaluate the tool's visualization quality and usability. The survey questionnaires were first projected on-screen, followed by the obtained result graphs, and the corresponding observations and analyses conducted by the authors. Testers were requested to confirm or object to the observations and analyses resulting from the test data. Subsequently, the session concluded with an open discussion where the testers were invited to provide their opinions concerning the limitations of and the possible improvements to the tool. Concerning the empirical results review: testers unanimously concurred with the produced observations and analyses described in Sections 4.4.1 and 4.4.2. No reservations were recorded. We discuss the limitations and improvement recommendations in the following discussion section.

4.6. Discussion

To wrap up, this study is based on our intuition that users wish to acquire the most value out of the data, while spending the least amount of time and effort analyzing the data, i.e., while viewing the least amount of data. As a result, we developed the mirrored dendrogram tool to cater to the above intuition. In this discussion section, we first recap and compare the contributions of our solution with related works in the literature. Second, we recap and summarize the results of our empirical evaluation study. Third, se discuss the limitations of our solution and highlight future improvements and directions.

4.6.1. Comparative Analysis

Table 9 summarizes the main differences between our method and its related approaches based on data clustering techniques. In short, our approach: i) processes structured data (in contrast with parallel coordinates which describe the relationships between sets of flat data, and are not designed to compare structured data), ii) builds cluster dendrograms to describe the structural relationships between data items (in contrast with graph-based techniques which focus on improving the visualization of entities and connections within an individual graph, and do not specifically address the comparison of pairs of datasets), iii) computes the structural similarity between two dendrograms (this is partially achieved with tanglegram and cluster heatmap, which only compare structured data according to their leaf node ordering, disregarding their inner node structural similarities), iv) matches both inner nodes and leaf nodes to visualize the dendrogram structural similarities (in contrast with tanglegram and cluster heatmap which do not visualize the similarities within the dendrogram structures themselves, but rather only visualize their leaf node mappings – this is often misleading when evaluating the correlation between tree structures, since two trees can have different internal structures, while their leaf nodes are presented in a matching order, and vice versa), and iv) computes the best structural zooming level to visualize the mappings between the mirrored dendrograms (which is not achieved with any existing tool to our knowledge).

Table 9. Comparing our method with related data visualization tools based on clustering techniques

Visualization tools	Processes structured data	Builds cluster dendrograms	Computes structure similarity between dendrograms	Matches leaf nodes	Matches inner nodes	Computes best structural matching level
Parallel Coordinates (e.g., [9, 39])	×	×	×	×	×	×
Dendrogram (e.g., [3, 13])	✓	✓	×	×	×	×
Tanglegram (e.g., [13, 18])	✓	✓	Partial	✓	×	×
Cluster Heatmap (e.g., [24, 28])	✓	✓	Partial	✓	×	×
Graph-based (e.g., [2, 5])	✓	×	×	✓	✓	×
Mirrored Dendrograms (our approach)	✓	✓	✓	✓	✓	✓

4.6.2. Empirical Analysis

We have conducted quantitative and qualitative evaluations to assess our visualization tool. For the quantitative study, we considered sample data from three different sources: i) DBLP: the computer science bibliography database, ii) IMDB: the internet movie database, and iii) SSG: semantic SVG graph database. Sample documents were selected to build 40 mirrored dendrogram visualizations from each database, producing a total of 120 visual iterations comparing sample documents against each other to highlight their feature correlations. We measured two evaluation metrics: i) the time needed by a user to identify the matching features, and ii) the accuracy of the mapped features. The time metric indicates how much time a user needs to spend assessing the visualization to understand and identify the mapped features: the more time spent, the lesser the quality of the visualization tool. The accuracy metric indicates the quality of the mapped features as identified by the user: the higher the number of accurate mappings detected, the better the quality of the visualization tool. A total of 40 human testers (senior engineering students) participated in this study, where every tester independently processed 10 sample visualizations, each consisting of a mirrored dendrogram, a tanglegram, and a cluster heatmap describing the same pairs of data entries, to identify the root node mappings. In short, mirrored dendrogram consistently produced minimum mapping time and maximum accuracy levels with the increase in the number of data nodes being processed, compared with tanglegram and cluster heatmap. Based on discussions with the testers, this is mainly due to i) the existence of mappings between inner nodes, and ii) the easiness of zooming-in and zooming-out of the inner node mappings. Most human testers found the inner node mappings and zooming functionalities available through mirrored dendrogram very useful to better identify the root node mappings, compared with the other two visualization tools which only provide leaf node mappings. Also, clustered heatmap consistently ranked second, performed better than tanglergram in terms of both time and accuracy. Discussions with testers revealed that color-coding the leaf node connections was helpful in identifying the correlating areas within the two structured being compared.

For the qualitative study, we assessed both i) the visualization quality and ii) the usability of our tool. A total of 20 testers (senior students, graduates, and professionals) were invited to independently fill two surveys: one targeting visualization quality, and another targeting ease of use. For visual quality, we considered five evaluation criteria: i) feature correlation visualization, ii) default zooming levels, iii) zooming in and out actions, iv) tool's interactive functionalities, v) comparison with existing solutions. Considering all visualization criteria combined, results produce an average overall rating score of 7.75/10 (with stdev = 1.73 and Cronbach's alpha = 0.84), highlighting the overall visualization quality of the tool and the consistency of the results according to most testers. For usability, we considered a typical SUS (System Usability Scale) template: i) no learning is required, ii) feeling confident in using the tool, iii) not cumbersome to use, iv) most people would learn it quickly, v) no inconsistency in the tool, vi) functions are well integrated, vii) no need for technical support, viii) tool is easy to use, ix) not unnecessarily complex, and x) tool would be frequently used. Considering all usability criteria combined, results produce an average overall rating score of 7.44/10 (with stdev = 1.59 and Cronbach's alpha = 0.93), highlighting the overall usability of the tool according to most testers. Note that Cronbach's alpha for the usability criteria is clearly higher than its visualization counterpart (i.e., 0.84, cf. Section 4.4.1), due to the closer relationship between the usability criteria themselves, which is evident in their definitions (e.g., criterion 1 - *no* (*minimal*) *learning is required before using the tool* versus criterion 4 - *people would learn it quickly*, and criterion 8 - *tool is easy to use* versus criterion 9 - *tool is not unnecessarily complex*, cf. Table 8), where correlated ratings are expected.

4.6.3. Limitations and Future Improvements

Following the collection and analyses of the empirical results, an online session was organized with the testers to acquire and discuss their feedback about the tool, highlighting their concerns and limitations that require improvement. In short, most testers concurred about the usefulness of inner node mapping and zooming functionalities to better understand the structural mappings and correlations between structured data. Many testers also highlighted the usefulness of color-coding the node connections to identify mapping features in a faster and more effective way.

Concerning the limitations of the tool, two main points were highlighted: i) the visualization might become cumbersome with an increased number of nodes in the dendrograms (testers suggested proposing an upper limit on the number of nodes that can be processed by the tool, in order to maintain a certain level of understanding of the visualization results), ii) node display ordering might mislead the viewer into believing the data is more or less correlated than they really are, regardless of the internal dendrogram structure (this is similar to the problem faced with tanglegrams, where the trees being compared can have different internal structures or topologies, while their leaf nodes are presented in a matching order, cf. Section 2.3. The latter mentioned limitations are similar to the problems encountered with tanglegrams, e.g., [13, 18], where there's a need to reduce the number of entanglements between node connections to make the visualization easier to understand.

Various possible solutions to handle the edge entanglements problem can be investigated, including leaf node ordering, edge bundling, tree zooming, and tree filters. Many leaf node ordering solutions have been proposed especially in the field of bioinformatics to visualize multiple phylogenetic trees in order to identify common patterns in their subtree structures [14, 19]. For instance, the authors in [11, 14] propose multiple dynamic programming solutions aiming: i) to minimize the number of leaves for deletion from one tree in order to correctly match the input order of the remaining leaves of the other tree, ii) to delete the minimum number of leaves in one tree such that the remaining leaves of both trees can be ordered with the same order, and iii) to utilize external data about some expected order on the tree leaves (such as chronological order when the time dimension is available, or some semantic order if a reference taxonomy or ontology is available) [29, 41]. Edge bundling is another group of techniques which aim to reduce the clutter in hierarchical structures by grouping similar edges into bundles and ordering leaf nodes to minimize crossing within bundles [25, 46]. An edge routing function is usually minimized to decide on the edges which need to be bundled together, while keeping the paths relatively short by penalizing the routing of too many edges through narrow gaps between the nodes. As a result, paths belonging to the same bundle are nudged away from each other, making them more visible with the bundles acquiring more thickness. Subsequently, an order of the edge segments within each bundle is computed to minimize the number of crossings between the edges of the same bundle [6, 46]. Hierarchical tree zooming is another group of techniques which aim to re-organize the structural properties of a tree in order to enhance the understandability of its underlying hierarchical data [17, 42, 49]. The main premise with this family of techniques is that hierarchical structure represents information at different levels of details and every level of detail can show a different set of nodes and paths from the structure. Various node clustering and hierarchical zooming solutions have been investigated to provide a seamless zooming in and out of the tree hierarchies, to optimize certain criteria (e.g., improving accessibility to the data, providing more/less visibility to certain details, and highlighting the users' data preferences) [17, 49]. Tree filtering can also be used to spatially smoothen the dendrogram visualization by optimizing edge filters or spatial filters [33, 70]. Local or regional cost aggregation functions can be optimized by processing the nodes of the tree as image pixels, and the edges as the connections between the nearest neighboring pixels. The similarity between any two pixels is decided by their shortest distance on the tree, which can be subsequently used to apply different local or regional filter functions to optimize the spatial representation of the trees in question [33, 70].

Concerning other future improvements to the tool, the following points were suggested: i) increase the usage of color coding to further improve the visualization, ii) include a cluster heatmap–like visualization into the mirrored dendrogram tool to improve the visualization, and iii) allow the mapping of more than two mirrored dendrograms (e.g., mapping three or four dendrograms together) using regular edges or hyper-edge structures (where a single hyper-edge would connect three or four nodes at a time). The latter is not a trivial task, and would transform the tool from a two dimensional visualization into a multi-dimensional visualization (similar to multi-dimensional parallel coordinate visualizations discussed in Section 2.1). This can be particularly useful when considering the time dimension to describe temporal data (where data belonging to the same timestamp can be clustered together and presented on a separate plane related to the specific timestamp).

In light of the above feedback, we believe many improvements and revisions can be done to enhance mirrored dendrograms. First, we need to conduct a thorough user study through surveys and exploratory interviews, in order to confirm our intuition and motivations behind the mirrored dendrogram design. Second, in upcoming studies, we aim to contemplate and answer the following questions. While the zooming functionality was proven useful in our experiments, could zooming mislead the user or hide certain useful information when focusing on a specific part of the structure? Are there specific situations where the user would need to see more or less of the data to get a better idea of larger or smaller patterns? The presence of many mapping lines between closely mapped nodes would result in an occlusion problem that would make it difficult to compare between the structured, especially when comparing structured with interleaving similar and dissimilar parts, as well as structures with relatively large numbers of nodes. Is there a way to combine mirrored dendrograms with cluster heatmaps, or leverage the color-coding of cluster heatmaps within mirrored dendrograms to improve the mapping presentation of the tool: maybe replacing line connections with some other form of color-coded representation? We aim to address the above questions and improvements in our upcoming studies.

5. Conclusion

This paper introduces a new unsupervised feature-based tool for interactive data visualization titled "mirrored dendrograms". It accepts as input semi-structured and multi-featured data, and allows the user to select the target features to be visualized and mapped against each other. Different from existing solutions like tanglegram and cluster heatmap, mirrored dendrogram offers three main contributions: (i) it produces a dendrogram structure for each combination of target features, connecting the data's internal nodes to describe their structure relationships (instead of connecting their leaf nodes only), (ii) the user can zoom-in and out of the data to show their relationships at different granularity (compared with existing static solutions which do not allow any zooming functionality), and (iii) the tool identifies the best zooming level which highlights the maximum correlation between the mapped data albeit with the minimal amount of details presented to the user (acquiring the most value out of the data, while viewing the least amount of data). We have evaluated our solution using multiple use case scenarios, where 60 testers participated in quantitative and qualitative evaluations to assess the data visualization tool, compared with existing solutions. Testers evaluated visual quality by measuring i) the time needed to identify the matching features between data, and ii) the accuracy of the mapped features. A qualitative survey was also conducted to evaluate the tools usability, interactivity, and data zooming quality. Empirical results are promising and highlight the quality and potential of the tool.

We are currently extending the tool to consider the time dimension in order to describe temporal data. This requires producing a three dimensional visualization by considering the time when forming dendrograms, where data belonging to the same timestamp will be clustered together and presented on a separate plane related to the specific timestamp. We are also working on a solution to reduce the number of line crossings (entanglements) by manipulating leaf node order [13, 18], aiming to address the mapping lines occlusion problem in order to improve user experience [36]. We also plan to extend the tool toward describing additional spatial and semantic dimensions, describing the location of the data and their meaning according to a reference dictionary [62, 69]. Users will be able to choose the dimensions to be visualized, according to their needs. This is specifically useful when correlating social media data (e.g., describing social event correlations [1, 58]) and sensor network data (e.g., describing network event correlations [21, 22]).

Acknowledgements

We would like to thank Dr. Sola Bahous Aoun, MD, PhD, MHPE, Dean of the Gilbert and Rose-Marie Chaghoury School of Medicine at the Lebanese American University (LAU), and Nephrology Specialist at LAU Medical Center-Rizk Hospital, for her assistance in vetting and anonymizing the EHRs used in our empirical evaluation. We are also very grateful to all the testers who helped evaluate our software tool.

Declarations

The initial design of mirrored dendrograms and its primary results are described in [43]. This paper extends the mirrored dendrograms tool and functionalities, and adds a substantial experimental evaluation to evaluate its performance in different use cases and with different datasets. The new tool recommends the best zooming level to display the dendrograms, by introducing a new dedicated measure computing the maximum correlation (similarity) and the minimal amount of details (granularity) presented to the user. The tool also provides new interactive visualization capabilities, allowing the user to adjust the zooming level and the number and weight of the connections between the mirrored dendrograms. Connection width are automatically adjusted to reflect the mapped nodes' similarity scores. Connection colors can be automatically adjusted to reflect different sub-clusters within the connected dendrograms. Visual snippets can be automatically added from the source datasets to provide a visual description of the connected sub-clusters through their root nodes. The tool has undergone an extended empirical evaluation, using multiple use case scenarios including 120 mirrored dendrogram visualizations from the DBLP, IMDB, and SSG databases, along with their corresponding tanglegram and cluster heatmap visualizations in order to perform an extended comparative evaluation study.

We confirm the authors' contributions in this work and their distribution of the tasks as follows: Angela Moufarrej (M.Sc.): Conceptualization, Software, Data curation, Validation, Writing-Original Draft. Abdulkader Fatouh (B.Eng.): Software, Implementation, Data curation, Validation, Writing-Original Draft. Joe Tekli (M.S. supervisor): Conceptualization, Methodology, Formal analysis, Writing-Original Draft, Writing-Review and Editing, Visualization, and Supervision.

Conflicts of interest/Competing interests

Not Applicable.

Compliance with Ethical Standards

- Research involving human participants and/or animals
  1. Statement of human rights: *Ethical approval: For this type of study formal consent is not required.*
  2. Statement on the Welfare of Animals: *Ethical approval: This article does not contain any studies with animals performed by any of the authors.*

- Informed consent: *Additional informed consent was obtained from all individual participants for whom identifying information is included in this article.*

Availability of Data

The datasets generated and analyzed during the current study are accessible online at the following link: http://sigappfr.acm.org/Projects/MirroredDendrograms/. They are also available from the authors on request.

Availability of Code

A software demo and an executable version of the prototype are available online at the following link: http://sigappfr.acm.org/Projects/MirroredDendrograms/.

References

[1] Abebe M., et al., *Generic Metadata Representation Framework for Social-based Event Detection, Description, and Linkage.* Knowledge Based Systems 2020. 188.

[2] Abou Akar C., et al., *Generative Adversarial Network Applications in Industry 4.0: A Review.* Int. J. Comput. Vis. , 2024. 132(6): 2195-2254.

[3] Ahmad A. and Khan S., *Survey of State-of-the-Art Mixed Data Clustering Algorithms.* IEEE Access 2019. 7: 31883-31902.

[4] Algergawy A., Nayak R., and Saake G., *Element Similarity Measures in XML Schema Matching.* Elsevier Information Sciences, 2010. 180(24): 4975-4998

[5] Attieh J. and Tekli J., *Fast Text Classification using Lean Gradient Descent Feed Forward Neural Network for Category Feature Augmentation.* International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'23), 2023. 2341-2348.

[6] Bekos M., et al., *Line Crossing Minimization on Metro Maps.* International Symposium on Graph Drawing and Network Visualization (GD'07), 2007. 231-242.

[7] Bengesi S., et al., *Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers.* IEEE Access, 2024. 12: 69812-69837.

[8] Biswas A., et al., *A Study of Multi-Objective Restricted Multi-Item Fixed Charge Transportation Problem considering Different Types of Demands.* Applied Soft Computing, 2022. 118:108501.

[9] Bok J., Kim B., and Seo J., *Augmenting Parallel Coordinates Plots With Color-Coded Stacked Histograms.* IEEE Trans. Vis. Comput. Graph, 2022. 28(7): 2563-2576.

[10] Bokhan D., et al., *Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning.* Frontiers in Physics, 2022. 10:1069985.

[11] Brandes U., *Optimal leaf ordering of complete binary trees.* Journal of Discrete Algorithms, 2007. 5(3): 546-552.

[12] Britzolakis A., Kondylakis H., and Papadakis N., *AthPPA: A Data Visualization Tool for Identifying Political Popularity over Twitter.* Information journal, 2021. 12(8): 312.

[13] Buchin K., et al., *Drawing (Complete) Binary Tanglegrams - Hardness, Approximation, Fixed-Parameter Tractability.* Algorithmica, 2012. 62(1-2): 309-332.

[14] Bulteau L., Gambette P., and Seminck O., *Reordering a Tree According to an Order on Its Leaves.* 33rd Annual Symposium on Combinatorial Pattern Matching (CPM'22), 2022. 24:1–24:15.

[15] Chen C., Huang E., and Yan H., *Detecting the Association of Health Problems in Consumer-level Medical Text.* Journal of Information science, 2018. 44(1): 3-14.

[16] Cong I., Choi S., and Lukin M., *Quantum convolutional neural networks.* Nature Physics, 2019. 15(12):1273–1278.

[17] De Luca F., et al., *Multi-level tree based approach for interactive graph visualization with semantic zoom.* CoRR abs/1906.05996, 2019.

[18] De Vienne D., *Tanglegrams are Misleading for Visual Evaluation of Tree Congruence.* Molecular Biology and Evolution, 2019. 36(1): 174-176, doi:10.1093/molbev/msy196.

[19] Dwyer T. and Schreiber F., *Optimal leaf ordering for two and a half dimensional phylogenetic tree visualisation.* In APVis '04: Proceedings of the 2004 Australasian symposium on Information Visualisation, 2004. 35: 109–115.

[20] Dwyer T., *Scalable, Versatile and Simple Constrained Graph Layout.* Comput Graph Forum, 2009. 28(3):991–8. doi:http://dx.doi.org/10.1111/j.1467-8659.2009.01449.x.

[21] Ebrahimi D., Sharafeddine S., and A.C. Ho P., *Data Collection in Wireless Sensor Networks Using UAV and Compressive Data Gathering.* GLOBECOM, 2018. pp. 1-7.

[22] Ebrahimi D., et al., *UAV-Aided Projection-based Compressive Data Gathering in Wireless Sensor Networks.* IEEE Internet Things journal, 2019. 6(2): 1893-1905.

[23] Edwards R., *UPGMA Worked Example.* Edwards Lab, University of New South Whales, Australia, 2016. http://www.slimsuite.unsw.edu.au/teaching/upgma/.

[24] Engle S., et al., *Unboxing Cluster Heatmaps.* Proceedings of the Symposium on Biological Data Visualization (VIS'17), 2017. 18(S-2):63:1-63:15.

[25] Erves R. and Zerovnik J., *Improved upper bounds for vertex and edge fault diameters of Cartesian graph bundles.* Discrete Applied Mathematics, 2015. 181: 90-97.

[26] Fua Y., Ward M., and Rundensteiner E., *Hierarchical Parallel Coordinates for Exploration of Large Datasets.* IEEE Visualization'99, 1999. pp. 43-50.

[27] Gal A., Roitman H., and Sagi T., *From Diversity-based Prediction to Better Ontology & Schema Matching.* Inter. WWW Conference, 2016. pp. 1145-1155.

[28] Galili T., et al., *Heatmaply: an R Package for Creating Interactive Cluster Heatmaps for Online Publishing.* Bioinformatics, 2018. 34(9):1600-1602.

[29] Gambette P., et al., *Evaluating hierarchical clustering methods for corpora with chronological order.* Second International Conference of the European Association for Digital Humanities (EADH'21), 2021. https://hal.archives-ouvertes.fr/hal-03341803.

[30] Halkidi M.;Batistakis Y. and Vazirgiannis M., *Clustering Algorithms and Validity Measures.* Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM), 2001. pp 3-22.

[31] Henry N. and Fekete J., *Matrixexplorer: a dual-representation system to explore social networks.* IEEE Transactions on Visualization and Computer Graphics, 2006. 12:677–684.

[32] Herrmann J. et al., *Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases.* Nature Communications 2022. 13(1):4144.

[33] Jin Y., Zhao H., and Bu P., *Spatial-tree filter for cost aggregation in stereo matching.* IET Image Process, 2021. 15(10): 2135-2145.

[34] Johansson J., Treloar R., and Jern M., *Integration of Unsupervised Clustering, Interaction and Parallel Coordinates for the Exploration of Large Multivariate Data.* 8th IEEE International Conference on Information Visualisation, 2004, 2004. pp. 52-57.

[35] Johansson J., Cooper M., and Jern M., *3-Dimensional Display for Clustered Multi-Relational Parallel Coordinates.* 9th International Conference on Information Visualisation, 2005. pp. 188-193.

[36] Karim R., et al., *Improving user experience of color palette extraction by using interactive visualization based on hierarchical color model.* International Journal of Human-Computer Studies, 2023. 169: 102924.

[37] Koren Y., *On Spectral Graph Drawing.* 9th Annual International Conference on Computing and Combinatorics (COCOON'03), 2003. Springer-Verlag LNCS, pp. 496-508.

[38] Li Y., et al., *MR-BIRCH: A Scalable MapReduce-based BIRCH Clustering Algorithm.* Journal of Intelligent and Fuzzy Systems, 2021. 40(3): 5295-5305.

[39] Lou J., Dong K., and Wang M., *A Parallel Coordinates Plot Method Based on Unsupervised Feature Selection for High-Dimensional Data Visualization.* . International Conference on Wireless Communications and Mobile Computing (IWCMC'21), 2021. pp. 532-536.

[40] Ma Y. and Chbeir R., *Content and Structure Based Approach for XML Similarity.* Proceedings of the IEEE International Conference on Computer and Information Technology (CIT), 2005. pp. 136-140.

[41] Moisl H., *How to visualize high-dimensional data: a roadmap.* Journal of Data Mining and Digital Humanities, 2020, 2020. doi:10.46298/jdmdh.5594.

[42] Moncada D., Reich J., and Tchangou M., *Interactive information zoom on Component Fault Trees.* Modellierung, 2018. 311-314.

[43] Moufarrej A., Fatouh A., and Tekli J., *Unsupervised and Dynamic Dendrogram-based Visualization of Medical Data.* International Web Information Systems Engineering conference (WISE'24), 2024. Doha, Qatar.

[44] NCSS Statistical Software, *Clustered Heatmaps.* 2022. Ch. 450, pp. 1-12, http://ncss.com.

[45] Nohno K., et al., *Spectral-Based Contractible Parallel Coordinates* 18th International Conference on Information Visualization, Paris, France. , 2014. pp. 7-12, doi:10.1109/IV.2014.60.

[46] Pupyrev S., et al., *Edge routing with ordered bundles.* Computational Geometry: Theory and Applications, 2016. 52: 18-33.

[47] Raj. J., *7 Ways Data Visualization Can Improve Sales and Marketing Alignment.* In intellectyx, 2019. https://www.intellectyx.com/blog/ways-data-visualization-can-improve-sales-and-marketing-alignment/.

[48] Rice J.A., *Mathematical Statistics and Data Analysis.* Duxbury Press, 3rd Edition, 2006. 688 pages.

[49] Saadeh H. and Tekli J., *Hierarchical Indexing for Interactive Zooming of Document Clusters.* International Conference on Web Information Systems Engineering (WISE'24), 2024. 1: 286-303.

[50] Sakai R., et al., *Modular Leaf Ordering Methods for Dendrogram Representations in R.* F1000Research, 2014. 3(177). doi:http://dx.doi.org/10.12688/f1000research.4784.1.

[51] Sakai R., *Gapmap: Functions for Drawing Gapped Cluster Heatmap with ggplot2.* R Package, Version 0.0.4. , 2015. https://CRAN.Rproject.org/package=gapmap.

[52] Salameh K., El Akoum F., and Tekli J., *Unsupervised Knowledge Representation of Panoramic Dental X-ray Images using SVG Image-and-Object Clustering.* Multimedia Systems, 2023. 29(4): 2293-2322.

[53] Salazar R., *Operations Research with R - Transportation Problem.* Towards Data Science, 2019. https://towardsdatascience.com/operations-research-in-r-transportation-problem-1df59961b2ad.

[54] Salloum G. and Tekli J., *Automated and Personalized Nutrition Health Assessment, Recommendation, and Progress Evaluation using Fuzzy Reasoning.* International Journal of Human-Computer Studies (IJHCS), 2021. Volume 151, 102610.

[55] Salloum G. and Tekli T., *Automated and Personalized Meal Plan Generation and Relevance Scoring using a Multi-Factor Adaptation of the Transportation Problem.* Soft Computing, 2022. 26(5): 2561-2585.

[56] Simpao A., et al., *A Review of Analytics and Clinical Informatics in Health Care.* Journal of Medical Systems, 2014. 38(4), 1-7.

[57] Stasko J. and Zhang E., *Focus+ Context Display and Navigation Techniques for Enhancing Radial, Space-filling Hierarchy Visualizations.* IEEE Symposium on Information Visualization, 2000. p. 57–65. doi:http://dx.doi.org/10.1109/INFVIS.2000.885091.

[58] Taddesse F.G., et al., *Semantic-based Merging of RSS Items.* World Wide Web Journal: Internet and Web Information Systems Journal Special Issue: Human-Centered Web Science., 2010. 13(1-2): 169-207, Springer Netherlands.

[59] Tekli J., Chbeir R., and Yétongnon K., *Minimizing User Effort in XML Grammar Matching.* Elsevier Information Sciences Journal, 2012. 210:1-40.

[60] Tekli J., et al., *Approximate XML Structure Validation based on Document-Grammar Tree Similarity.* Elsevier Information Sciences, 2015. 295:258-302.

[61] Tekli J., et al., *(k, l)-Clustering for Transactional Data Streams Anonymization.* Information Security Practice and Experience, 2018. pp. 544-556.

[62] Tekli J., et al., *Full-fledged Semantic Indexing and Querying Model Designed for Seamless Integration in Legacy RDBMS.* Data and Knowledge Engineering, 2018. 117: 133-173.

[63] Tekli J., *An Overview of Cluster-based Image Search Result Organization: Background, Techniques, and Ongoing Challenges.* Knowl. Inf. Syst., 2022. 64(3): 589-642.

[64] Walz E., *BMW aims to deploy humanoid robots at its Spartanburg factory.* Automotive Drive, 2024. https://www.automotivedive.com/news/bmw-autonomous-humanoid-robots-spartanburg-factory-figure-bots/705680/.

[65] Wang W., et al., *Visualization of Large Hierarchical Data by Circle Packing.* Conference on Human Factors in Computing Systems, 2006. p. 517-20. doi:http://dx.doi.org/10.1145/1124772.1124851.

[66] Weinstein J., *A Postgenomic Visual Icon.* Science jounal 2008. 319(5871):1772–3. doi:http://dx.doi.org/10.1126/science.1151888.

[67] Wiebe N., Kapoor A., and Svore K., *Quantum Deep Learning.* arXiv:1412.3489, 2014.

[68] Xiao A.. et al., *Synlidar: Learning from synthetic lidar sequential point cloud for semantic segmentation.* CoRR abs/2107.05399, 2021. https://arxiv.org/abs/2107.05399.

[69] Yakhni S., et al., *Using Fuzzy Reasoning to Improve Redundancy Elimination for Data Deduplication in Connected Environments.* Soft Computing, 2023. https://doi.org/10.1007/s00500-023-07880-z.

[70] Yang Q., *Stereo matching using tree filtering.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015. 37: 834-846.

[71] Zhao W., et al., *Dense text retrieval based on pretrained language models: A survey.* ACM Transactions on Information Systems (TOIS), 2024. 42(4):1–60.

[72] Zhonghua Y. and Lingda W., *3D-Parallel Coordinates: Visualization for Time Varying Multidimensional Data.* 7th IEEE International Conference on Software Engineering and Service Science (ICSESS'16), 2016. doi:10.1109/ICSESS.2016.7883153.

[73] Zou F., et al., *A Reinforcement Learning Approach for Dynamic Multi-objective Optimization.* Information Sciences, 2021. 546: 815-834.