Kuznetsova, Irina; Lugmayr, Artur; Holzinger, Andreas

Visualization methods of hierarchical biological data

Published in:
7th International Workshop on Semantic Ambient Media Experiences, SAME 2014

Published: 01/01/2017

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Visualization Methods of Hierarchical Biological Data: A Survey and Review

Irina Kuznetsova  
Holzinger Group, HCI-KDD, Institute for Medical Informatics/Statistics, Medical University, Graz, Austria  
irina.kuznetsova@student.tugraz.at

Artur Lugmayr  
Visualisation and Interactive Media, Lab. (VisLab) Curtin University, Perth, AU & Aalto Univ., Helsinki, Finland  
lartur@acm.org

Andreas Holzinger  
Holzinger Group, HCI-KDD, Institute for Medical Informatics/Statistics, Medical University, Graz, Austria  
a.holzinger@hci-kdd.org

ABSTRACT
The sheer amount of high dimensional biomedical data requires machine learning, and advanced data visualization techniques to make the data understandable for human experts. Most biomedical data today is in arbitrary high dimensional spaces, and is not directly accessible to the human expert for a visual and interactive analysis process. To cope with this challenge, the application of machine learning and knowledge extraction methods is indispensable throughout the entire data analysis workflow. Nevertheless, human experts need to understand and interpret the data and experimental results. Appropriate understanding is typically supported by visualizing the results adequately, which is not a simple task. Consequently, data visualization is one of the most crucial steps in conveying biomedical results. It can and should be considered as a critical part of the analysis pipeline. Still as of today, 2D representations dominate, and human perception is limited to this lower dimension to understand the data. This makes the visualization of the results in an understandable and comprehensive manner a grand challenge.

This paper reviews the current state of visualization methods in a biomedical context. It focuses on hierarchical biological data as a source for visualization, and gives a comprehensive survey of visualization techniques for this particular type of data.

CCS CONCEPTS
• Human-centered computing→Visualization techniques

KEYWORDS
Visualization, hierarchical data, computer graphics, information visualization, big data, bioinformatics.

1 INTRODUCTION
The research domain of information visualization is broad, and involves a wide range of research fields, such as computer graphics (e.g. 2D and 3D graphics), information design to increase communication and sense making [24], creative aspects (e.g. design, layouts, colour use) [72], and methods from human-computer interaction. Making data understandable from a cognitive and machine learning point of view has emerged recently in the literature [40, 41], including the idea to render data throughout smart environments [37–39][49].

The biomedical domain is a complex field of biological processes. In addition, the advancements of biological technologies has led to a dramatic increase in data volumes [26], which has presented new challenges in knowledge extraction. Working with high-volume data requires the application of data mining that draws upon machine learning techniques. These methods help in extracting knowledge patterns and narrowing data to the smaller volumes. These include e.g. Support Vector Machines (SVM); Artificial Neural Networks (ANN); clustering [46][4]; statistical techniques (e.g. Bayesian statistics [51]; Hidden Markov Models (HMMs) [10]; Principle Component Analysis (PCA); classification methods [28][26][1].

The interpretation of extracted knowledge and the result of analysed data through visualisation is an essential step in the analysis pipeline, and becoming an important tool in bioinformatics. These not only includes simple visualizations (i.e. bar plots, pie charts, flow charts), but also advanced visualization techniques for representing final results in the biomedical domain (e.g. 3D). As also valid for other domains, visualisations should follow information design principles, which are defined in [60]:

1. Providing an overview of the data;
2. Zoom in/out options;
3. Filtering of unnecessary information;
4. Detailization of region of interests;
5. Relation between data points of interest;
6. History of actions; and
7. Possibility of extracting required parameters.

Another example for applying visualisation in biology is the application of new technologies such as the Next Generation Sequencing (NGS), which delivers enormous volumes of genomic data in a digital format. To visualise the data, genome browser applications are utilized to allow a real-time visualisation and exploration of genomic sequences; of any region of interest; in any required scale within a genome [33][31][56].

Within the scope of this paper, we firstly give an overview of existing visualization techniques, followed by description of the characteristics of hierarchical data. We focus mainly on traditional visualization techniques, as i.e. the visualization of hierarchically organized data in a 2D space at first place. We illustrate our approach based on a typical biological analysis workflow as previously discussed in [35]. The workflow narrows genetical information into a meaningful smaller subset called differentially expressed genes. These represent the active genes in the overall genetic information, and can be utilized to obtain Gene Ontologies.
(GO), which are categorizing the functions of various genes. Visualisation supports the understanding of results, as well as the obtained ontologies.

2 RELATED WORKS

We would like to point to the following works in information visualisation and design for further reading: [60], [42], and the excellent introductory guide [64]. Visualization plays a key role in the biomedical domain. Various techniques aim to deliver the correct representation of results in visual format, and follow visualization design principles described in [60]. To state an example, the visualization of a protein structure in 3D space enables researchers to have an overview of the studied protein; to rotate a protein image in different dimensions; to see protein-protein interactions; to measure an atomic distance; and to zoom into the region of interests. Several visualisation tools have been developed and support the analysis process. Remaining in the domain of protein structures Web3DMol, UCSF Chimera and POLYVIEW-3D are just a few examples for visual protein structure investigation [59][48][50].

Table 1: Examples of visualization software

<table>
<thead>
<tr>
<th>Biological Task</th>
<th>Visualization Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview</td>
</tr>
<tr>
<td>1. A Protein Structure Visualization</td>
<td></td>
</tr>
<tr>
<td>Web3DMol</td>
<td>see [59]</td>
</tr>
<tr>
<td>UCSF Chimera</td>
<td>see [48]</td>
</tr>
<tr>
<td>POLYVIEW-3D</td>
<td>see [50]</td>
</tr>
<tr>
<td>2. NGS Data Visualization</td>
<td></td>
</tr>
<tr>
<td>IGV</td>
<td>see [53]</td>
</tr>
<tr>
<td>UCS</td>
<td>see [31]</td>
</tr>
<tr>
<td>ZEMBU</td>
<td>see [56]</td>
</tr>
<tr>
<td>3. Hierarchical Data Visualization (Phylogeny)</td>
<td></td>
</tr>
<tr>
<td>ETE Toolkit</td>
<td>see [21]</td>
</tr>
<tr>
<td>PhyD3</td>
<td>see [33]</td>
</tr>
<tr>
<td>EvolView</td>
<td>see [33, 78]</td>
</tr>
</tbody>
</table>

Another set of tools allowing real-time visualisation and exploration of genomic sequences are the IGV, UCS, or ZEMBU genome browsers, which have been mentioned in the introduction section of the publication [53][31][56]. These, and many others allow the exploration of various samples of sequenced genomic data.

Another example of information visualization in biology is phylogeny, where hierarchical data structure is considered for image development. The intuitive way of representing hierarchies is as a tree diagram. The ETE Toolkit, PhyD3, EvolView and other visualization programs enable phylogenetic trees to be studied in more detail [21][33][33, 78].

Several tools mentioned within this section are summarized in Table 1, which classifies them according data type, and biological task. Visualisation is a very important research tool, which enables researchers to explore and study biological structures, investigate biological data in various digital formats, and understand molecular processes in an intuitive and comprehensive way.

3 CHARACTERISTICS AND PROCESSING OF HIERARCHICAL DATA

The hierarchical pattern is observed in numerous aspects of our life and various biological fields are not an exception: phylogenetics, GO, microarray analysis, differential expression analysis (dendrograms), and protein similarities represent data as hierarchies. Hierarchically organized data facilitates results comprehension and interpretation and provides a global overview of the data. Hierarchical clustering belongs to an unsupervised machine learning technique used for building hierarchies. Data is usually presented as a parent-child relation, where a parent can have zero or more related children (see Fig. 1 A-B).

3.1 Processing Hierarchical Data Structures

To process hierarchical structures, numerous computational methods have been developed such as neighbour-joining [54], UPGMA [63], maximum parsimony [11][13], and maximum likelihood [57]. These utilize distance-based, character-based, or statistical method of approaching hierarchies respectively.

Distance-based clustering is the traditional method for hierarchical clustering. The input data is a matrix, where rows characterize a unique object, and columns show the object’s features. The distance matrix, also called a proximity matrix, is calculated with a linkage method (see Table 2) which enables the estimation of dissimilarity/similarity between objects. There are two types of algorithms for hierarchical clustering: (1) agglomerative, and (2) divisive [1]. The agglomerative or bottom-up is one of the popular hierarchical clustering algorithms that starts from grouping the two closest data points of the distance matrix into a cluster, updating the distance matrix for the just-generated cluster and the original matrix based on the selected linkage method, and continuing this process until only a single cluster remains [25]. In other words, it starts from grouping the closest data points of the input data ("bottom"), and ends when each data point is assigned to its related cluster ("up"). In contrast, the divisive method or top-down follows an opposite way of grouping data values. It considers an input data as one whole cluster and splits the data into smaller clusters by moving from the “top” – (one cluster) to the “down”- (many clusters) [1].

In addition to existing clustering algorithms, new techniques have been developed with the aim of addressing the emerging issues associated with large data volumes produced by new technologies. Loewenstein and team proposed a memory-constrained UPGMA (MC-UPGMA) algorithm that enables clustering of the large data sets that was implemented in C++ [36]. Kannan and Wheeler extended the parsimony score to phylogenetic networks; the algorithm was implemented in OCAML [30].

### Table 2: Hierarchical clustering - types of linkage methods [26].

<table>
<thead>
<tr>
<th>Linkage methods</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-linkage</td>
<td>D(Ci,Cj) = min ( x_p \in Ci, x_q \in Cj ) d(x_p, x_q)</td>
</tr>
<tr>
<td>Complete-linkage</td>
<td>D(Ci,Cj) = max ( x_p \in Ci, x_q \in Cj ) d(x_p, x_q)</td>
</tr>
<tr>
<td>Average-linkage, WPGMA</td>
<td>D(Ci,Cj) = ( \frac{D(Ci,Cm)+D(Cj,Cm)}{2} )</td>
</tr>
<tr>
<td>Average-linkage, UPGMA</td>
<td>D(Ci,Cj) = ( \frac{D(Ci,Cm)+D(Cj,Cm)+D(Cm,Cn)}{3} )</td>
</tr>
<tr>
<td>Centroid-linkage</td>
<td>D(Ci,Cj) = d(c_i, c_j) for some centroid c_c_i, c_c_j</td>
</tr>
<tr>
<td>Median-linkage, WPGMC</td>
<td>D(Ci,Cj) = d(w_i, w_j)</td>
</tr>
<tr>
<td>Ward’s linkage</td>
<td>ESS = ( \sum_{x_p \in Ci} \hat{x}_p - \bar{x} |^2 )</td>
</tr>
</tbody>
</table>

### 3.2 Data Representation, Storage and Queries

Other essential aspects in dealing with hierarchical data is data representation, and data encoding for e.g. storage in a database or as in-memory representations for applying algorithms. Theoretical considerations can be found in database theory, formal languages, and query languages. An overview of these techniques can be found in an interesting online article [82], are listed in Table 3, and more details about the theoretical aspects can be found in [81][8][52]. Several techniques offer different ways of accessing the required information.

### Table 3: Methods for storing hierarchical data [82].

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjacency List</td>
<td>Recursive method. Each node of the tree has a pointer to a parent node. Intuitive and simple for implementation, but slow in performing queries.</td>
<td>[45]</td>
</tr>
<tr>
<td>Path Enumeration</td>
<td>Each entry is stored as a full path to the root.</td>
<td>[70]</td>
</tr>
<tr>
<td>Nested Set</td>
<td>Applies traversal method of numbering nodes. Each node is visited twice where each time the number of the visit is assigned (has two pointers) and stored. Fast for retrieving required information, but becomes slow for updating a tree.</td>
<td>[6]</td>
</tr>
</tbody>
</table>

Representation of hierarchical data as adjacency list model (see Fig. 1-A), where each element of the table has a pointer to its parent, or it can be visualized as a tree diagram (see Fig. 1-B).

### Figure 1: Hierarchical data representation as: A- the adjacency list model; B- a tree diagram.

#### 3.3 Challenges in Hierarchical Data Processing

### 3.3.1 Data Accessibility

The problem of modern health sciences is that generated data may not be accessible to a health science researcher directly [18], because certain patterns (“knowledge”) are hidden in arbitrarily high-dimensional spaces. Examples range from longitudinal rheumatology data sets, in which cohorts of patients are attributed with vectors in R\(^{100}\) [62], to the uncertainties of RNA sequence base pairing variants with a potentially arbitrary number of dimensions [27]. The results gained by machine learning and knowledge extraction techniques need to be mapped down into the lower dimensions to make them accessible to a human expert [73]. This calls for a closer cooperation between machine learning and visualization experts [74]. A crucial factor in clustering techniques is the curse of dimensionality [32]. With increasing
dimensionality, the volume of the space increases so quickly that the available data becomes sparse, hence becoming extremely difficult to find reliable clusters. A further significant problem is that distances become imprecise as the number of dimensions grows, since the distance between any two points in a given data set converges; moreover, different clusters might be found in totally different sub spaces. Consequently, a global filtering of attributes on its own is not sufficient.

3.3.2 Subspace Clustering. The subspace clustering problem is difficult, as very different characteristics for grouping can be used: this can be highly subjective and context-specific and requires an expert-in-the-loop [19][20]. What is recognized as comfort for end-users of individual systems? It is interesting to note that human experts are quite capable in determining similarities and dissimilarities, which has been described by nonlinear Multidimensional Scaling (MDS) [58][68].

We can represent similarity relations between entities as a geometric model consisting of a set of points within a metric space. The output of an MDS routine is a geometric model of the data, with each object of the data set represented as a point in n-dimensional space. Consequently, there is urgent need to map very high-dimensional data into a small number of relevant dimensions to make it accessible for human expert analysis. For example, the similarity between patients may change by considering different combinations of relevant dimensions [22]. This is called subspace analysis and is a very interesting and relevant field of current research [12]. For example, with the goal of finding a k-dimensional subspace of \( \mathbb{R}^d \) in a way that the expected squared distance between instance vectors and the subspace is a minimum. This so-called subspace learning can also be used as a dimensionality reduction technique [15]. Common tools include the stationary subspace analysis toolbox [44], SubVIS [23] and Morpheus [43] – just to mention three.

4 VISUALISATION OF HIERARCHICAL DATA

Tree-like structured graphs are a common way of representing hierarchical data. In general, a tree-structured graph is defined as a root node, which is connected through links or edges to the parent and children nodes [77]. The traditional tree view is visualized in upside-down way, where the root is on the top and the child-child relation is shown towards the bottom. However, a tree graph can be also represented as a left-to-right diagram [76].

According to [55], visualization of hierarchical organized data can be represented as (see Table 4) [55]:

1. explicit vs implicit;
2. axes-oriented vs radial (see Table 4).

The implicit method belongs to the space-filling technique that fits provided data into a defined space for example, rectangular, triangle, circular etc. The explicit method utilizes a traditional tree-like structure.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Axes-Oriented Layout</th>
<th>Radial Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Visualization method representing hierarchy as a node-link diagram [55]</td>
<td>Dendrogram, Intended tree (see Fig. 2, Fig. 4)</td>
<td>Circular tree (see Fig. 3)</td>
</tr>
<tr>
<td>Explicit</td>
<td>Visualization method representing hierarchy in a space-filling way [55]</td>
<td>Tree-maps (see Fig. 5)</td>
<td>Sunburst (see Fig. 6)</td>
</tr>
</tbody>
</table>

Table 4: Types of visualization methods of hierarchical data.

There are a range of visualization graphs that enables hierarchies to be shown in 2D format. Dendrograms and intended layouts (see Fig. 2, Fig. 4) are examples of the explicit method in an axes-oriented layout, whereas a circular tree (see Fig. 3) is an explicit method in a radial layout. Space-filling techniques can also use axes-oriented layouts, such as tree-maps (see Fig. 5); or in a radial layout such as Sunburst (see Fig. 6).

4.1 Explicit Visualization

This part of the paper describes 2D visualization techniques of hierarchical data on GO data. The example subset data is taken from the REVIGO Web server at (http://revigo.irb.hr/), which applies a neighbor-joining hierarchical clustering algorithm to achieve hierarchies [66].

The output data of REVIGIO tool may be used as input data for tree-maps, or Sunburst visualization methods, whereas a distance-based clustering used for a tree structured diagrams. The second and the third columns of Table 5 named “representative” and “description” show the parent-child relationships respectively. For example, “nucleoside triphosphate metabolism” is a parent node of five related children annotations such as “nucleoside triphosphate metabolic process”, “alanine biosynthetic process”, “inositol biosynthetic process”, “isocitrate metabolic process” and “regulation of translation”. The “response to herbicide” and “ion transmembrane transport” have two and four related children correspondingly, whereas “protein refolding” has no children at all.

Although a tree diagram is a traditional way of representing hierarchies, REVIGO can visualize hierarchies as scatter plots, interactive graphs, tree-maps, tag clouds and intended trees [66].

4.1.1 Dendrograms. A dendrogram, also called a binary tree (see Fig. 2), is a visualization technique commonly used in representing groups of similarities (clusters) in the data produced by the hierarchical clustering method [16][25].

It has a traditional tree-like structure, where leaves are placed at the same level. The y-axis (height) shows the distance at which a cluster is formed. The labels across the x-axis are equally distributed for readability purposes. The dotted line is an example of a selected distance cut-off that enables the reader to see the number of clusters that found within that distance. Figure 2 illustrates that four distinct clusters were identified (represented in red, purple, blue, and green colours) if the closeness of objects was defined as a distance of value two. In biology, a clustering approach
Table 5: The hierarchical data as GO. The example data is taken from REVIGO [66], and modified for explanatory purposes.

<table>
<thead>
<tr>
<th>Term_ID</th>
<th>Representative (parent)</th>
<th>Description (child)</th>
<th>Freq</th>
<th>log10 pval</th>
<th>Uniqueness</th>
<th>Dispensability</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0009141</td>
<td>nucleoside triphosphate metabolism</td>
<td>nucleoside triphosphate metabolic pr.</td>
<td>1.61%</td>
<td>-300</td>
<td>0.689</td>
<td>0</td>
</tr>
<tr>
<td>GO:0006523</td>
<td>nucleoside triphosphate metabolism</td>
<td>alanine biosynthetic process</td>
<td>0.05%</td>
<td>-26</td>
<td>0.791</td>
<td>0.273</td>
</tr>
<tr>
<td>GO:0006021</td>
<td>nucleoside triphosphate metabolism</td>
<td>inositol biosynthetic process</td>
<td>0.02%</td>
<td>-10</td>
<td>0.817</td>
<td>0.331</td>
</tr>
<tr>
<td>GO:0006102</td>
<td>nucleoside triphosphate metabolism</td>
<td>isocitrate metabolic process</td>
<td>0.02%</td>
<td>-10</td>
<td>0.839</td>
<td>0.376</td>
</tr>
<tr>
<td>GO:0010447</td>
<td>response to herbicide</td>
<td>response to acidic pH</td>
<td>0.01%</td>
<td>-14</td>
<td>0.863</td>
<td>0.546</td>
</tr>
<tr>
<td>GO:0034220</td>
<td>ion transmembrane transport</td>
<td>ion transmembrane transport</td>
<td>3.53%</td>
<td>-300</td>
<td>0.912</td>
<td>0</td>
</tr>
<tr>
<td>GO:0055085</td>
<td>ion transmembrane transport</td>
<td>transmembrane transport</td>
<td>8.92%</td>
<td>-300</td>
<td>0.932</td>
<td>0.497</td>
</tr>
<tr>
<td>GO:0006818</td>
<td>ion transmembrane transport</td>
<td>hydrogen transport</td>
<td>1.15%</td>
<td>-300</td>
<td>0.866</td>
<td>0.366</td>
</tr>
<tr>
<td>GO:0015797</td>
<td>ion transmembrane transport</td>
<td>mannitol transport</td>
<td>0.01%</td>
<td>-11</td>
<td>0.903</td>
<td>0.379</td>
</tr>
<tr>
<td>GO:0042026</td>
<td>protein refolding</td>
<td>protein refolding</td>
<td>0.07%</td>
<td>-204</td>
<td>0.957</td>
<td>0.029</td>
</tr>
</tbody>
</table>

is commonly used to find groups of genes that share similar features based on results from Differential Expression (DE) analysis. Heatmaps represent a matrix of values of genes expression in a color-coded way and are accompanied with dendrograms. Dendrograms are illustrated along the heatmap on the top and/or left sides. The left side dendrogram represents the similarity between genes, and the top dendrogram the similarity between samples. Dendrograms are also common structures in representing phylogenetic trees [7].

Figure 2: Dendrogram, (visualized with R).

4.1.2 Circular Trees. There are another two ways of visualizing trees as radial trees, and circular trees [76]. In the radial tree the hierarchical tree structure is visualized in an annulus wedge; the algorithm was proposed by P. Eades [9]. In the circular tree visualization, the root is placed at the central positions and leaf nodes are equally distributed around on the perimeter of a circle (see Fig. 3) [76]. The hierarchy in this case is shown with a tree-structure graph. Coloring and labelling are used to improve representation of tree graphs. The circular tree layout that is also explicit method used for representing phylogenetic trees [7].

Figure 3: Circular Tree, (visualized with R).

4.1.3 Intended Trees. The intended layout is another way of representing hierarchies (Fig. 4). The data is plotted along the vertical axis and indentations are used in representing parent/children relationships. This type of visualization is commonly used for interface systems or online, as it allows easy access to required information by scrolling down. However, this technique has an unpublishable format and hence cannot be used as an effective overview of the data.

Figure 4: Intended Tree.
4.2 Implicit Visualizations

4.2.1 Tree-maps. Tree-maps are a space-filling technique, also known as implicit, used to represent hierarchical structures, proposed in 1992 and are described in [55][67][61]. Tree-maps apply a recursive algorithm for visualizing nested rectangles. The tree-map uses outer rectangle as a tree’s root and the inner space of this rectangle is filled with nested rectangles representing the parent/children relationship. This space is divided between parent nodes according to its assigned weight in the shape of rectangles, and each parent is subdivided into the amount of related children as further rectangles (see Fig. 5). Alternative algorithms have been proposed [5][79], as the original method suffers from the creation of narrow rectangles that impair the visualization’s readability. The original tree-map layout was “slice and dice”; the idea has since been extended with the development of the web-based tree-map by Wattenberg [75], the strip or ordered tree-algorithm [3] and the spiral layout algorithm that enables the reader to see changes in hierarchical data [66].

![Figure 5: Tree-map, R(treemaps).](image)

Tree-maps offer efficient usage of the available display space, and provides a good overview of the entire data hierarchy. The size of rectangles is relative to the size of the related data object, which simplifies data interpretation and evaluation. The color-coding helps to distinguish between different cluster groups and also helps show the relationship between children to parent nodes. The main graphical parameters for tree-map plotting are visualization area size, position and color-coding [71].

On the other hand, tree-map visualization becomes poor with the increase of input data size. While tree-map graphs still provide a data overview, supporting visualization objects such as labels cannot be drawn on small rectangles. Visualization of GO terms with tree-maps is an example of a using tree-map layout in biology [66].

4.2.2 Sunbursts. An alternative space-filling visualization is to represent data in a radial layout such as the Sunburst (see Fig.6) [65][29]. The hierarchy is represented from the center outwards from it. The inner circle is the root of the hierarchical data, and multiple layers of rings represent the parent-child relationships next to each other [67]. As the Sunburst is a circular space-filling technique, the edges of the provided display space are unused. The wedge size is relative to the cluster size. The interpretation of wedges sizes is relatively easy for the reader as each slice is represented in a familiar proportional way. However, in the case of narrow wedge sizes, the readability and evaluation of the visualization becomes poor. This leads to the similar problem of losing some graph labels, but can be addressed by using the empty space around the circular layout. As with tree-maps, the Sunburst uses colouring to improve readability of the visualization. Other space-filling visualization techniques available are the Voronoi diagram[2], Ellimaps [47], icicle plots [34] and Beamtree [17].

![Figure 6: Sunburst diagram.](image)

5 CONCLUSIONS

We reviewed the most common visualization techniques available for hierarchical structured data in 2D space within the scope of this paper. Within the conclusion section, we pinpoint to the 7 most relevant categories for classifying and characterizing biological visualizations to support the development of visualization taxonomies.

5.1 Visualization Technique

Visualization techniques for hierarchical structured data in 2D can be classified as explicit (dendrograms, circular tree, intended trees) and implicit (tree-maps, Sunburst) methods. All of them have advantages and disadvantages, and the choice of the most suitable visualization technique depends on the final representation goal. For example, space-filling techniques are the best for representing a global overview of final results. However, with the increasing size of the data, details such as labels are often omitted to avoid cluttering the final picture. Tree-maps utilize the complete display space, while the Sunburst uses only part of it. On the other hand, the Sunburst provides a more intuitive understanding of the relationship between data values due to proportional representation of relationships [29]; it is harder to see the size difference between rectangles in tree-maps.

5.2 Visualization Design

In addition to appropriate selection of the visualization method, it is important to apply suitable supporting visualization features:
5.3 Interactive Multimedia Features

Modern technologies provide various techniques for exploring big data in real-time, such as interactive methods and web-based visualizations. However, representation of final results for big datasets in 2D space remains a challenge. In this review, we focus on hierarchical structured data specifically visualization methods. Taking into consideration the best features of existing techniques and applying them into development of new visualization methods may help in representing big data results in a clear, informative way.

5.4 Primary Visualization Tasks

Visualization aims to facilitate perception, comprehension of the data. The primary visualization tasks are presented in Table 1, which enable the exploration of data, and decision making during the analysis process of biological data.

5.5 Algorithms and Data Processing

Algorithms and pipelines involved in the analysis process as e.g. clustering (see Section 3).

5.6 Data Representation, Storage, and Query

Another issue to be addressed, is data representation, especially regarding the particularities of the organization of hierarchical data as tree-like structures. Table 3 lists these relevant features.

5.7 Software Tools and Analysis Pipelines

Utilized software tools as e.g. Web3DMol and analysis pipelines as listed in Table 1.

5.8 Future Work

In future works, we will focus on the development of a taxonomy for reviewing and classifying visual techniques of hierarchical biological data. We will utilize the criteria for the taxonomy, listed within the conclusion section.

REFERENCES
