Journal of Biomedical Informatics 44 (2011) 760–774

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Sequential patterns mining and gene sequence visualization to discover novelty from microarray data

A. Sallaberry [a], N. Pecheur [b], S. Bringay [b,c], M. Roche [b], M. Teisseire [d,*]

[a] LaBRI, INRIA Bordeaux Sud-Ouest, Pikko, 351, cours de la Libération, 33405 Talence Cedex, France
[b] LIRMM, Univ. Montpellier 2 - CNRS, 161 rue Ada 34095 Montpellier Cedex 5 France
[c] MIAp Department, Univ. Montpellier 3, route de Mende 34199 Montpellier cedex 5, France
[d] Cemagref, UMR TETIS, Maison de la teledetection, 500 rue Jean-François Breton, 34093 Montpellier, France

ARTICLE INFO

Article history:
Received 30 August 2010
Available online 16 April 2011

Keywords:
Visualization
Data mining
Bioinformatics
Sequential patterns
Microarray data
Gene data

ABSTRACT

Data mining allow users to discover novelty in huge amounts of data. Frequent pattern methods have proved to be efficient, but the extracted patterns are often too numerous and thus difficult to analyze by end users. In this paper, we focus on sequential pattern mining and propose a new visualization system to help end users analyze the extracted knowledge and to highlight novelty according to databases of referenced biological documents. Our system is based on three visualization techniques: clouds, solar systems, and treemaps. We show that these techniques are very helpful for identifying associations and hierarchical relationships between patterns among related documents. Sequential patterns extracted from gene data using our system were successfully evaluated by two biology laboratories working on Alzheimer's disease and cancer.

1. Introduction

DNA microarrays have been successfully used for many applications (e.g. diagnosis and characterization of physiological states). They allow researchers to compare gene expression in different tissues, cells or conditions [1] and provide some information on the relative expression levels of thousands of genes that are compared in a few samples, usually less than a hundred (e.g., Affymetrix U-133 plus 2.0 microarrays measure 54,675 values). Nevertheless, due to the huge amount of data available, how process and interpret them to make biomedical sense of them remains a challenge. Data mining techniques, such as [2–4], have played a key role in discovering previously unknown information and shown that they can be very useful to biologists in identifying relevant subsets of microarray data for further analysis [5].

However, the number of results is usually so huge that they cannot easily be analyzed by the experts concerned. In [6], we proposed a general process, called GeneMining, based on the DBSAP algorithm for extracting sequential patterns from DNA microarrays [7]. We obtained patterns of correlated genes ordered according to their level of expression. Although this method is useful, the way to select relevant patterns is still not highly efficient. For instance, depending on the values of parameters, between 1000 and 100,000 patterns may be extracted, which are not easy to interpret. Thus, the main aim of this new work is to propose new visualization techniques to help biologists to navigate through the extracted patterns. Biologists are also faced with the problem of locating relevant publications about the genes involved in the patterns. Even if some tools are now available to automatically extract information from microarray data (e.g., [8] or [9]), there are still no user-friendly literature search tools available to analyze patterns.

In this paper, we describe an efficient tool to help biologists focus on new knowledge by navigating through large numbers of sequential patterns (i.e., sequences of ordered genes). Our contribution is twofold. First, we adapt three different techniques (i.e., point clouds, solar systems, and treemaps) to deal with data organized as a sequence and to produce an effective solution to the above problem. Second, using our system, the biologist can now be automatically provided with relevant documents extracted from the PubMed/MEDLINE database.[1] Although the methods described in this paper mainly focus on sequences extracted from DNA microarrays, they could easily be adapted to any other kind of sequential data.

The paper is organized as follows. In Section 2, we describe the data we are working with and give an overview of related

* Corresponding author. Fax: +33 467 548 700.
E-mail addresses: arnaud.sallaberry@labri.fr (A. Sallaberry), pecheur@lirmm.fr (N. Pecheur), bringay@lirmm.fr (S. Bringay), mroche@lirmm.fr (M. Roche), maguelonne.teisseire@cemagref.fr (M. Teisseire).

[1] www.ncbi.nlm.nih.gov/pubmed.

1532-0464/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2011.04.002

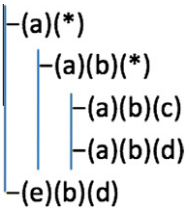

Fig. 1. Representation of the hierarchy.

work. In Section 3, we describe our proposal and the associated tool. In Section 4, we evaluate the new systems, and in Section 5 we present our conclusions and future work.

2. Preliminaries

In the framework of the Gene Mining[2] project (PEPS project funded by ST2I Institute of CNRS – France), we mined real data produced by the analysis of DNA microarrays (Affymetrix DNA U133 plus 2.0) to study Alzheimer's disease (AD) using the DBSAP algorithm [7]. This dataset was used to discover classification tools to distinguish between two *classes* (AD and healthy individuals). In [7], we proposed to extract patterns of correlated genes ordered according to their level of expression. An example of pattern is $\langle(MRVI1)(PGAP1,GSK3B)\rangle$ meaning that "the level of expression of gene *MRVI*1 is lower than that of genes *PGAP*1 and *GSK*3*B*, whose levels are very similar".

Although this method was useful since it proved that sequential patterns could be very useful for biologists, the way of selecting relevant patterns remained a challenge. Actually, depending on the values of parameters, 1000 to 100,000 patterns could be extracted and were consequently not easy to interpret. Biologists still needed a visualization tool to enable them to navigate through the huge amount of sequences, to select and order relevant novel sequences (e.g. sequences in which new gene correlations may exist), and to automatically query specific publications from Pubmed/MEDLINE (or other publication database) on the selected genes.

To summarize, an appropriate visualization tool needs to explore both kinds of data:

1. Gene sequences described by an ordered list of sets of genes and the class *supports* (i.e., the number of occurrences of this class in the database respecting this expression). As already mentioned, too many patterns are extracted. By using the k-means clustering algorithm with a sequence-oriented measure (S2MP [10]), we are able to identify groups of similar sequences and highlight a representative sequence called the centre. According to the centre, sequences can also be summarized [11] and organized according to a *hierarchy*. For example, the three sequences $\langle(a)(b)(c)\rangle$, $\langle(a)(b)(d)\rangle$ and $\langle(e)(b)(d)\rangle$ can be presented as a tree (Fig. 1). The two first are summarized by $\langle(a)(b)(*)\rangle$.
2. Documents in the literature dealing with genes from sequences. The documents are obtained from the bibliographical database Pubmed-MEDLINE (i.e. free digital archive of biomedical and life sciences literature) with or without gene synonyms [6]. We define a distance between a document and the gene sequence taking into account the publication date as well as the number of genes mentioned in the paper. The more recent the document and the more genes described in the paper, the closer the document will be to the sequence concerned.

[2] This work was conducted in collaboration with the MMDN lab ('Molecular mechanisms in neurodegenerative dementias' laboratory, University of Montpellier 2).

The visualization tool, which is described in the following section, combines all these elements: support, class, groups, hierarchy, and sets of documents. To facilitate specific tasks, we propose three different visual representations [12]. The "Point Cloud" representation is mainly used to show the set of sequences while the "Solar System" is mainly used to focus on a specific sequence. Finally, the treemap is very useful when hierarchies and volumes have to be represented.

The combination of these representations allows the user to explore gene sequences efficiently and to identify relevant information. A typical use of the application consists in looking at the clusters and identifying those containing particular genes of interest. The user can visualize interesting clusters in more detail and select the sequences that appear to be the most relevant according to their support and the users previous conjectures. Users also need easy access to the bibliography related to a particular sequence to (in)validate their arguments. Indeed, they need to access the supports of higher levels of a sequence in the hierarchy to evaluate the potential role of each gene in this sequence and to access the groups containing sequences beginning with high levels elements of the hierarchy

In [13], a visualization tool based on point clouds representing groups of sequences is proposed. Sequences are placed according to an alignment in a 3-dimensional space. However, this approach is not able to account for the hierarchy of sequences. Indeed, most previous works concerning visualization of biological sequences focus on the representation of sequence alignments [14–16]. To the best of our knowledge, no method is currently available to visualize sequences and associated documents, as most previous works deal with visualization of parts of a document [17], information about documents [18] or a collection of documents [19]. None of these methods is suitable for our context.

3. Sequencesviewer

SequencesViewer [20] helps biomedical experts to browse and explore sequences of genes identified by knowledge discovery techniques (see Fig. 2). In the following we describe the main representations selected according to Shneiderman's information visualization mantra [12]: "overview first (see Section 3.1), zoom and filter (group of sequences in Section 3.2), details-on-demand (sequences with documents in Section 3.2)". A fourth view based on a treemap have been added to give another point of view of the input data (see Section 3.3).

3.1. Point cloud

The Point Cloud representation allows biologists to visualize groups of gene sequences (see Figs. 3, 4). It gives an overview of the centres of the groups, the distance from the centres, and associated sequences. Three steps are required to compute the relevant positions of centres to limit the number of occlusions. We combine three algorithms and adapt them to our problem. An efficient interaction mode is also added to help users find the information they require.

3.1.1. Main placement of the centres

The basic idea is to locate the centres in such a way that the Euclidean distances between them are proportional to the distances between the sequences given by a matrix of distances D containing S2MP measures [10].

Let d_{ij} be the matrix value for a centre i and a centre j. We want to find the coordinates $p_i = (x_i, y_i)$ for each centre i so that $\|p_i - p_j\| \approx d_{ij}$ where $\|p_i - p_j\|$ is the Euclidean distance between the centres i and j.

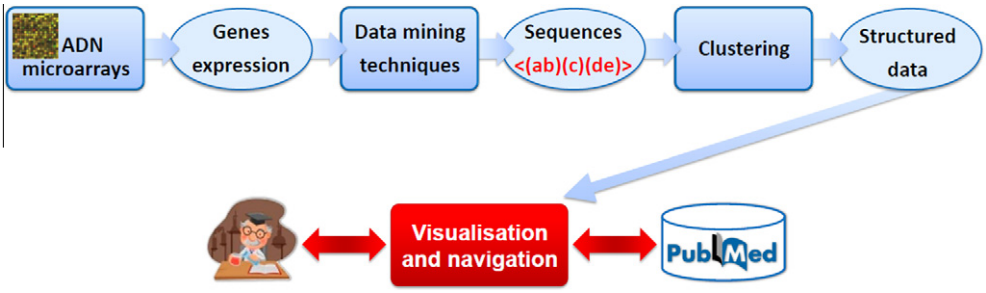

Fig. 2. SequencesViewer enables biologists to browse and explore sequences of genes and their related papers in Pubmed. These sequences are extracted and divided into groups using data mining and clustering techniques.

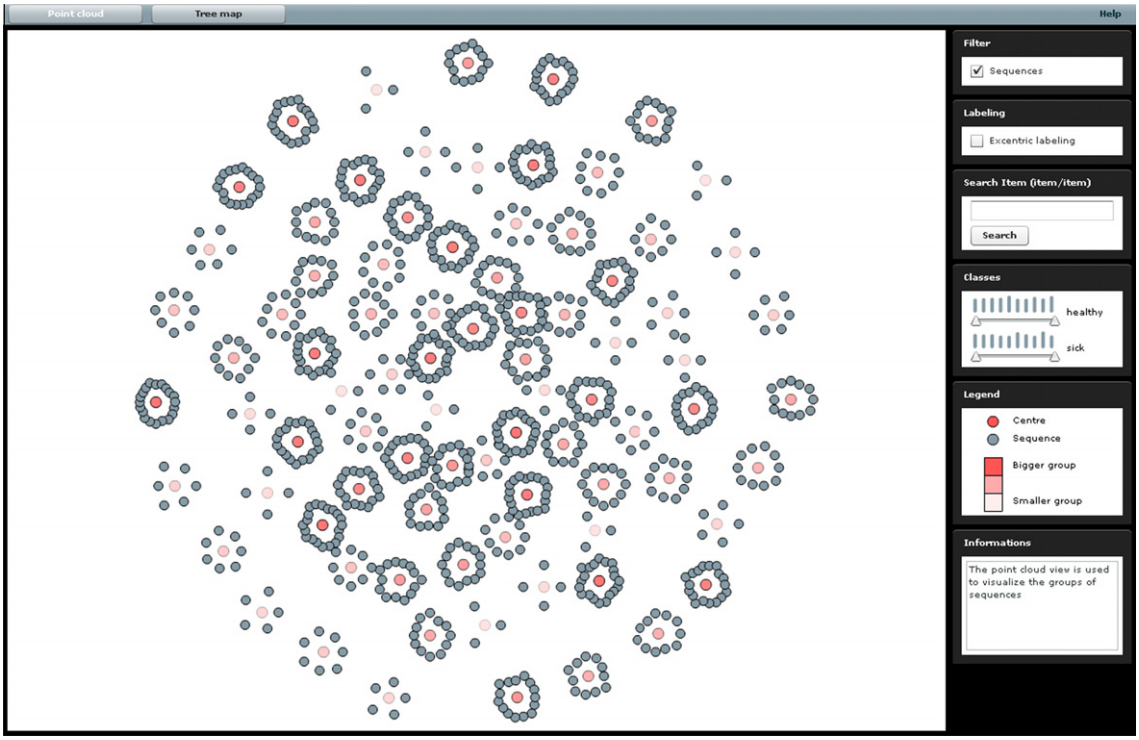

Fig. 3. Point cloud representation of sequences: the red nodes represent the centres of the groups and the grey one represents the related sequences.

Different techniques are described in the literature to assign a location to items in an N-dimensional space. Multidimensional Scaling (MDS) technique [21] is often used in information visualization and was first introduced by Togerson [22]. This technique produces representations that reveal similarities and dissimilarities in the dataset using a matrix of ideal distances. In our application, we want to find positions in a 2-dimensional space. We use a MDS optimization strategy called Stress Majorization [23], which consists of minimizing a cost function (i.e. stress function) that measures the square differences between ideal distances and Euclidean distances in 2-dimensional space:

$$\sigma(p) = \sum_{i<j\leqslant n} \omega_{ij}(d_{ij} - \|p_i - p_j\|)^2 \quad (1)$$

where $\omega_{ij} = d_{ij}^{-\alpha}$ and $p = p_1, p_2, \ldots, p_n$ is the actual configuration. We use $\alpha = 2$, which appears to produce good results in most cases, as shown by [24].

Several techniques have been developed to minimize the stress function (see [21] for an overview). In our application, we chose a method introduced in [24] for its simplicity, fast convergence and for the quality of the results. It consists of successively computing a simple function that returns position p_i:

$$p_i^{[t+1]} \leftarrow \frac{\sum_{j:j\neq i}\omega_{ij}\left(p_j^{[t]} + s_{ij}^{[t]}\cdot\left(p_i^{[t]} - p_j^{[t]}\right)\right)}{\sum_{j:j\neq i}\omega_{ij}} \quad (2)$$

where $p_i^{[t]}$ is the position of the centre i at time t and

$$s_{ij}^{[t]} = \begin{cases} \frac{d_{ij}}{\|p_i^{[t]} - p_j^{[t]}\|} & \text{if } \left\|p_i^{[t]} - p_j^{[t]}\right\| \neq 0 \\ 0 & \text{else} \end{cases} \quad (3)$$

This iterative updating is performed for each node and repeated until a stable configuration is reached. At each step, $\sigma(p)^{[t]} \geqslant \sigma(p)^{[t+1]}$ and the stress function converge to a local minimum [25].

3.1.2. Initial placement of the centres

One important aspect of these methods is to find an initial placement of the centres before performing the iterative process. Random placement is not efficient because each time the algorithm is executed for the same data, the final layout changes. Moreover, the stress majorization converges slowly and it can fall into local minima. In our system, we use the fold-free embedding defined in [26]. The algorithm selects four centres c_1, c_2, c_3 and c_4 so that they are in the periphery of the point cloud. The pair (c_1, c_2) has

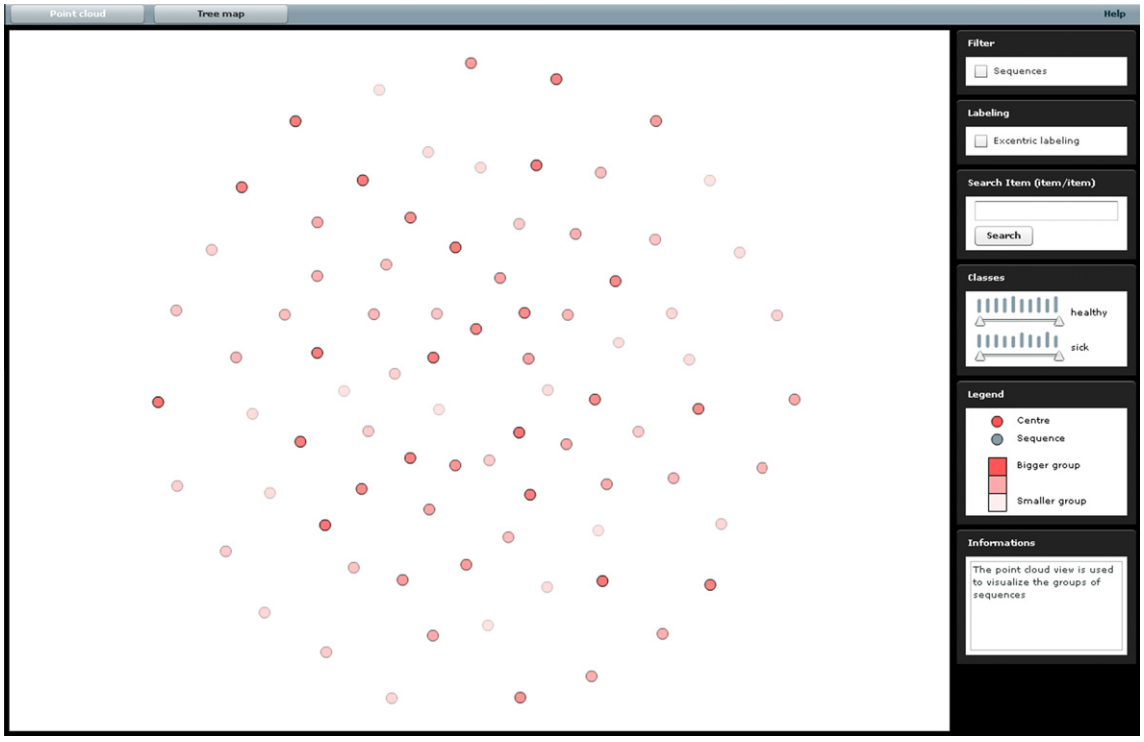

Fig. 4. Centres in the point cloud representation: checking a box enables the user to hide the sequences that are not the centre of their group.

to be roughly perpendicular to the pair (c_3, c_4) in the layout. A fifth centre c_5 is selected so that it lies in the middle of the point cloud. These centres are selected as follows:

1. Arbitrarily select a centre c_0. c_1 is the centre so that the $d_{c_0c_1} \geqslant d_{c_0i}$ for each centre i with $d_{c_0c_1}$ the distance between c_0 and c_1 in the matrix D.
2. c_2 is the centre so that $d_{c_1c_2} \geqslant d_{c_0i}$ for each centre i. Thus, c_1 and c_2 are roughly opposite one another in the point cloud.
3. c_3 is the centre so that $|d_{c_1c_3} - d_{c_2c_3}| \leqslant |d_{c_1i} - d_{c_2i}|$ for each centre i. It is roughly equidistant from c_1 and c_2.
4. As in the previous step, c_4 is one of the centres so that $|d_{c_1c_4} - d_{c_2c_4}| \leqslant |d_{c_1i} - d_{c_2i}|$ for each other centre i. Among this set of candidates, pick the one that maximizes $d_{c_3c_4}$. Thus, c_4 is roughly equidistant from c_1 and c_2 and roughly opposite c_3.
5. As in the previous step, c_5 is one of the centres so that $|d_{c_1c_5} - d_{c_2c_5}| \leqslant |d_{c_1i} - d_{c_2i}|$ for each other centre i. Among these candidates, pick the one that minimizes $|d_{c_3c_5} - d_{c_4c_5}|$. Thus, c_5 is roughly in the middle of the graph.

x_i denotes $d_{c_3i} - d_{c_4i}$ and y_i denotes $d_{c_1i} - d_{c_2i}$. We can use (x_i, y_i) coordinates directly to locate each centre i. Unfortunately, this solution disregards the distance between i and c_5. To overcome this problem, the method described in [26] computes the polar coordinate (ρ_i, θ_i) of a centre i so that $\rho_i = d_{c_5i} \times R$ and $\theta_i = tan^{-1}\left(\frac{y_i}{x_i}\right)$. Actually, we compute θ_i more accurately according to trigonometry:

$$\theta_i = \begin{cases} tan^{-1}\left(\frac{y_i}{x_i}\right) & \text{if } x_i > 0 \text{ and } y_i \geqslant 0 \\ tan^{-1}\left(\frac{y_i}{x_i}\right) + 2\pi & \text{if } x_i > 0 \text{ and } y_i < 0 \\ tan^{-1}\left(\frac{y_i}{x_i}\right) + \pi & \text{if } x_i < 0 \\ \frac{\pi}{2} & \text{if } x_i = 0 \text{ and } y_i \geqslant 0 \\ \frac{3\pi}{2} & \text{if } x_i = 0 \text{ and } y_i < 0 \end{cases}$$

3.1.3. Removing central overlap

The MDS method we implemented does not avoid overlapping of centres. Node occlusions can mislead the user by hiding information. We thus run a node overlap removal algorithm after the MDS placement step described above. Gansner and Hu [27] implemented a simple but effective solution based on a nice adaptation of the stress majorization process.

This solution is based on a Delaunay triangulation [28] computed for the set of centres and their current positions. A Delaunay triangulation is a triangulation that maximizes the minimum angle of all the angles of the triangles. We can represent the results of a triangulation on our centres as a planar graph $G(V,E)$ where V is the set of the centres and E is the set of the edges of triangles. The node overlap is removed iteratively:

1. First, we compute a Delaunay triangulation on the current layout. Let $G^{DT}(V,E^{DT})$ be the graph produced by the triangulation.
2. For each $\{i,j\} \in E^{DT}$ an *overlap factor* is computed:

$$t_{ij} = \max\left(\frac{a_i + a_j}{\|p_i - p_j\|}, 1\right) \quad (4)$$

where a_i is the radius of the centre i. $t_{ij} = 1$ if the centres i and j do not overlap. If $t_{ij} < 1$, we can remove the overlap by extending the length of the edge $\{i,j\}$ by this factor. A new ideal distance matrix is then computed: $d_{ij}^{DT} = s_{ij}^{DT}\|p_i - p_j\|$ where s_{ij}^{DT} is a factor computed from t_{ij} to damp it: $s_{ij}^{DT} = \min\{s_{max}, t_{ij}\}$, with $s_{max} > 1$ (1.5 in our implementation). s_{max} is the maximum amount of overlap we can remove at each step while keeping the same global configuration.
3. We now minimize the stress function using the process described above (see Eq. (2)) with d_{ij}^{DT} and s_{ij}^{DT} in spite of d_{ij} and s_{ij}.

$$\sigma^{DT}(p) = \sum_{i<j\leqslant n} \omega_{ij}\left(d_{ij}^{DT} - \|p_i - p_j\|\right)^2 \quad (5)$$

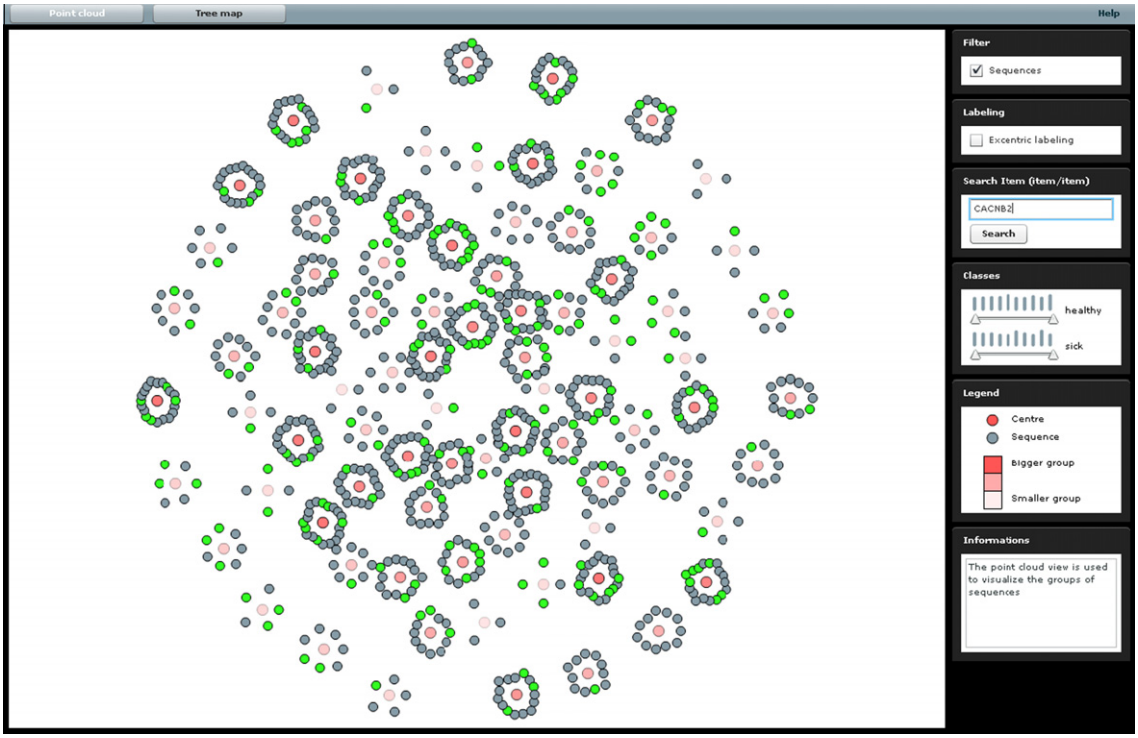

Fig. 5. Point cloud with sequences and highlighted searched items.

3.1.4. Interactions and navigation

The user can choose to visualize the centres (see Fig. 4) or the centres plus their associated sequences using the check box labelled *Sequences* (see Fig. 3). The colour of the centres is of different intensity, which is proportional to the number of sequences associated with the centre concerned. The legend on the right helps the user evaluate the size of the groups. An item can be searched and the sequences containing the searched term are highlighted. The screenshot in Fig. 5 represents a map with the highlighted sequences (in green[3]) resulting from a search operation.

Moreover, the user can move the whole map by dragging and dropping with a mouse. Zoom In/Out options are also available to the user by using the mouse wheel. A tooltip containing sequence informations is displayed when the user clicks on a sequence (see Fig. 6).

For each classes, a slider has been added to select a range of their corresponding support (see box *Classes* in Fig. 7). The sequences with a support out from this range are then filtered from the view. Inspired by Scented Widgets [29], a bar chart is displayed over each slider to represent the number of sequences sharing the corresponding support value.

Finally, we have implemented an excentric labelling technique [30]. When the user clicks on a free space in the map, labels of the sequences positioned inside a circle around the clicked point are displayed (see Fig. 8). To avoid overlaps, they are positioned far from their corresponding sequences: colours and lines are used to link the sequences with their own labels.

3.2. Solar system

When the user double-clicks on a sequence in the point cloud view, he/she accesses a second view (see Fig. 9) based on a solar system metaphor [31]. This view allows only the group of the selected centre to be explored. The centre is positioned in the middle of the visualization area (position (0,0)). Then, each sequence i is placed at a coordinate (d_i, θ_i) where d_i is the S2MP measure between the sequence and its centre and $\theta_i = i \cdot \frac{2\pi}{n}$ where n is the number of sequences. Grey circles have been added to the visualization to help the user approximate the value of d_i.

Interactions techniques previously described (i.e. zoom, search, sliders, tooltip, moving the whole map, excentric labelling) are also available in this view except the removing of the sequences,i.e. it is useless to display the centre alone. The legend is also displayed.

A second view based on the solar system can be accessed from the first view by double-clicking on a sequence (see Fig. 10). This view represents the sequence and its associated set of text documents i.e. scientific papers dealing with the genes belonging to

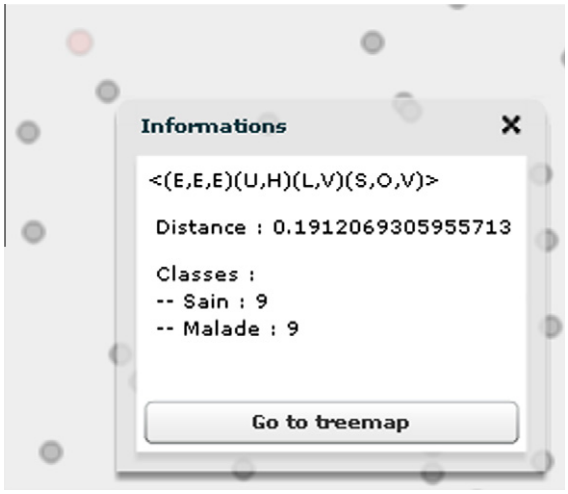

Fig. 6. Clicking on a sequence opens a tooltip containing its information.

[3] For interpretation of colour in Figs. 1–19, the reader is referred to the web version of this article.

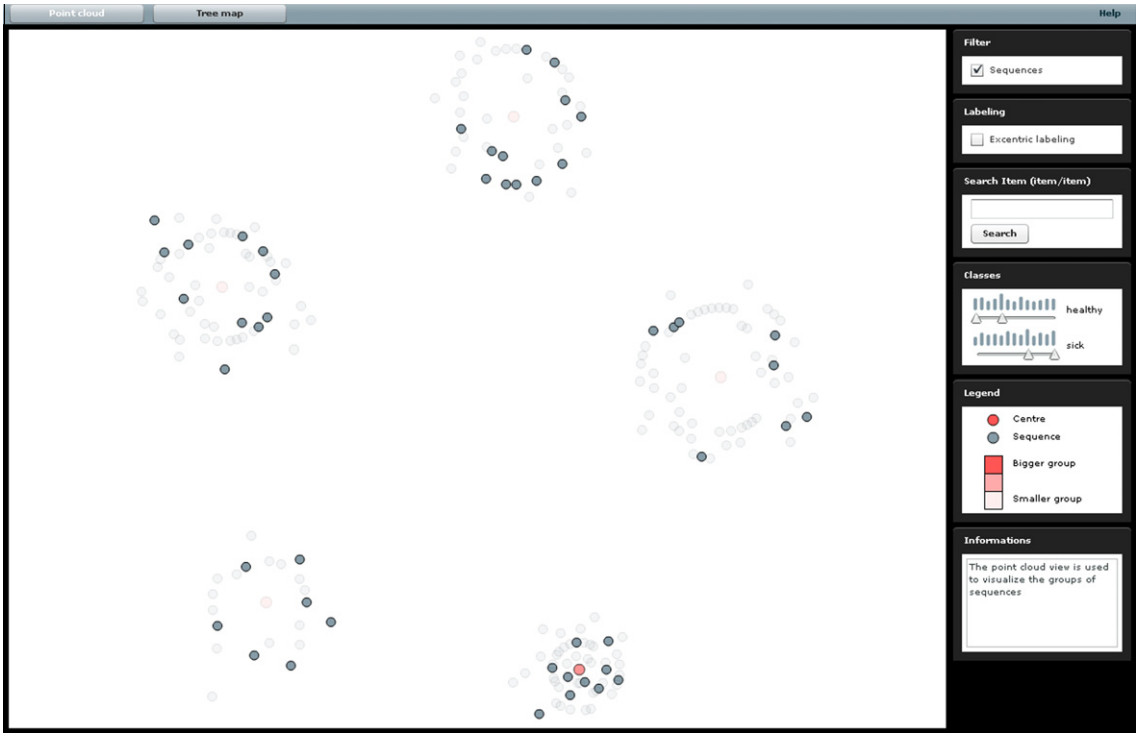

Fig. 7. Sliders are used to filter sequences according to their supports. The bar charts over them represent the number of sequences holding the corresponding values.

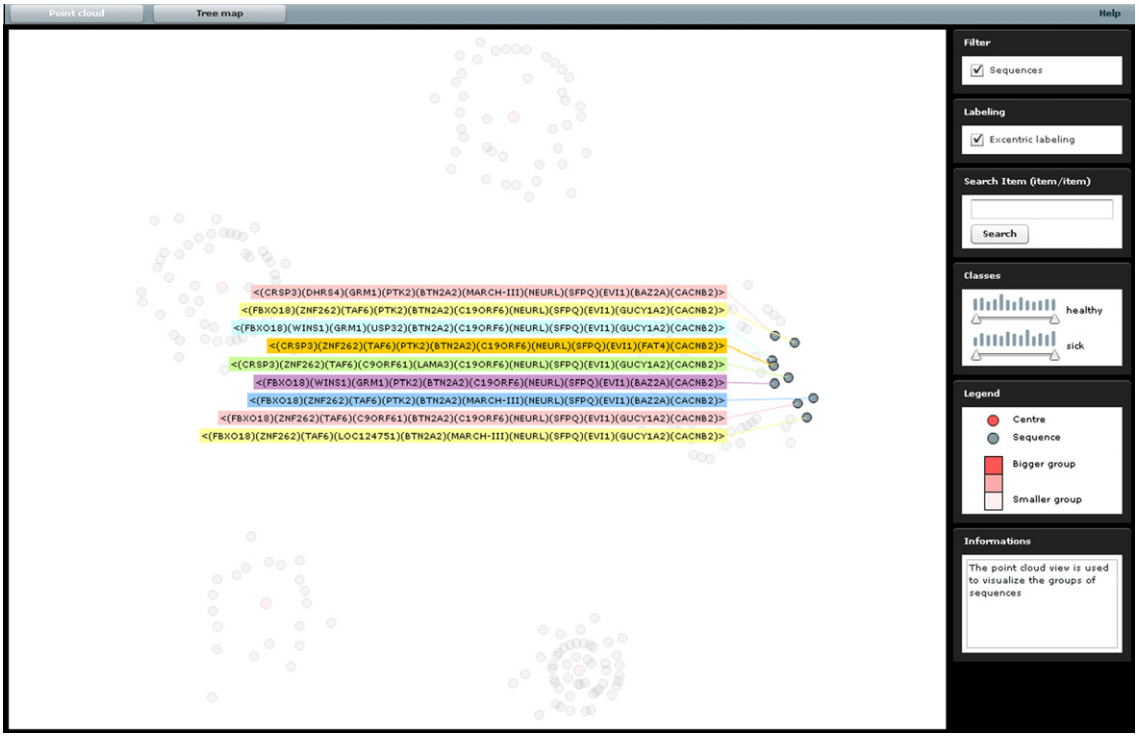

Fig. 8. Excentric labelling: the labels of the sequences lying around a click point are displayed at the periphery of the view to avoid overlaps.

the sequence concerned. These papers are extracted from the biomedical library Pubmed/MEDLINE.

The sequence is positioned in the middle of the visualization as the centre in the previous view. Documents are positioned around it. The distance between a document and the sequence is proportional to its proximity. The proximity depends on the year of publication and on the number of genes of the sequence referenced in the paper. The year of the publication is represented by different colour intensities. A tooltip containing node information is displayed when the user clicks on the sequence or on a document. The document can be opened by double clicking on it.

This type of visualization is convenient in the case of documents associated with sequences because the position of the documents helps the user select the sub-sets of documents of interest. Of

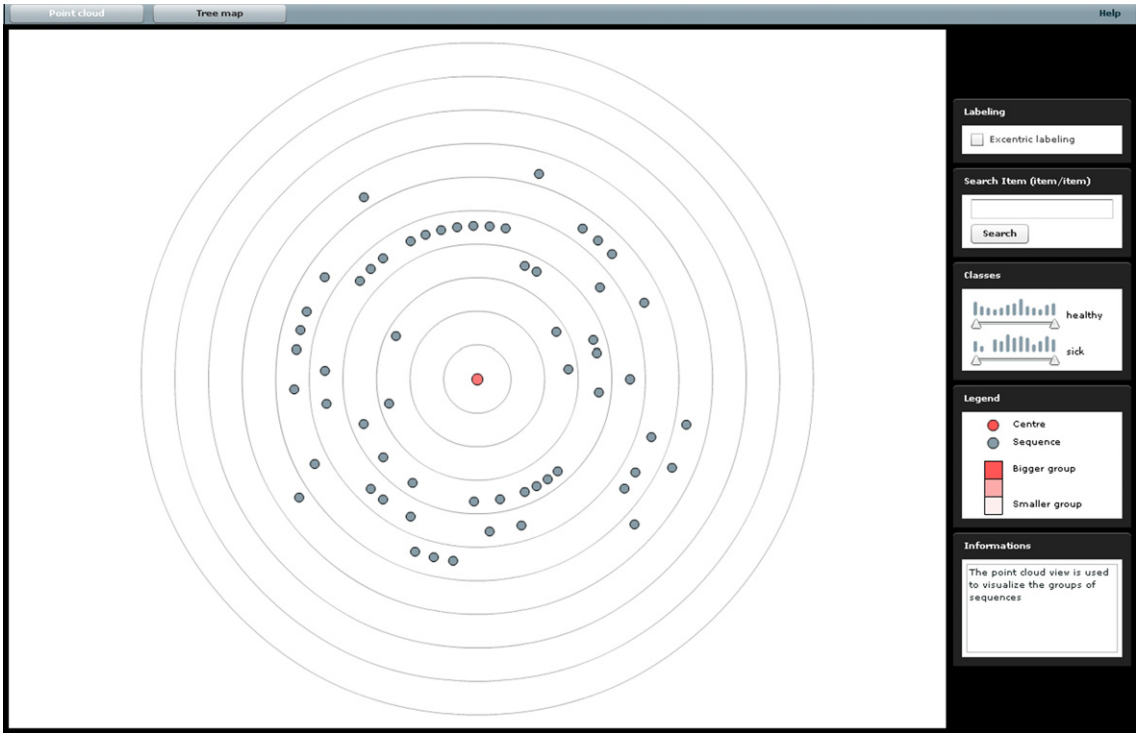

Fig. 9. Group of sequences: The red node corresponds to a sequence that is the centre of the group. The other nodes represent the other sequences of the same group. The distance between the centre and a sequence is proportional to the S2MP measure. While this measure computes values between 0 to 1, the grey circles represent values from 0.1 to 1 with a step of 0.1.

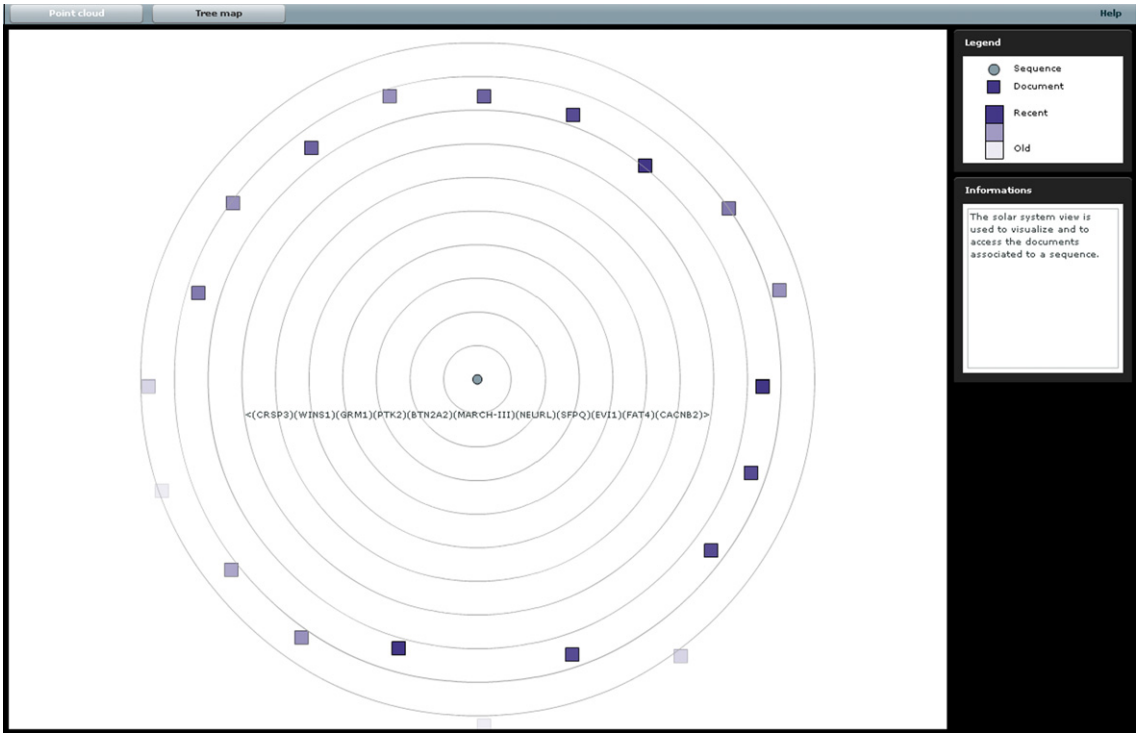

Fig. 10. Sequence of genes and its associated documents: the grey node corresponds to the sequence selected. The other nodes represent documents related to it. The distance between the sequence and a document is proportional to the year of publication and the number of genes of the sequence referenced in the paper. While this measure computes values between 0 and 1, the grey circles represent values from 0.1 to 1 with a step of 0.1.

course, other ways of visualizing text documents are described in the literature. There are two main approaches: visualization of specific subsections of large documents or visualization of clustered collections [32]. Here, we focus on the second type.

Even if only one variable is mapped so far in the solar system views (distance between the nodes and the centre of the view), we decided to use this kind of representation to enable users to map another variable. For example, they could order the documents according to the date of publication. The angle obtained in this way would represent this date.

Fig. 11 summarizes how to navigate through the point cloud and the solar systems views. In the point cloud view, the user has to double-click on a sequence to access a group view. In the group view, the user can reach the point cloud view by clicking on the point cloud button located at the top of the window. The user can also access the documents by double-clicking on a sequence. Double-clicking on a document node in the documents view opens the Pubmed web page of the corresponding article. The user can also click on the point-cloud button to reach the point-cloud view or double-click on the sequence to reach the group view.

3.3. Treemap

As described in Section 2, sequences are organized hierarchically. The representations described above do not enable the user to visualize and navigate through this hierarchy. That is why we combine them with a new one representing the tree produced by the hierarchy.

Tree representations have been widely used to represent hierarchical information. The first intuitive approaches were based on node-link representations like the one of Reingold and Tilfort [33]. However, these kinds of layouts require a lot of space and it is difficult to visualize large tree structures. We prefer a 2D space-filling approach (Treemap).

Treemaps were introduced by Johnson and Shneiderman in 1991 [34] to represent tree structures. They allow the user to visualize large amounts of hierarchical data by representing each node as a rectangle that is proportional to some of its attributes. We use such a view to represent the hierarchy of the sequences.

We use a squarified treemap, a technique introduced in [35], to generate rectangles that approximate squares. The algorithm consists in dividing recursively a rectangle r into k rectangles corresponding to the k children of r in the tree so that $width_i/weight_i \approx 1$ for each rectangle i. This problem is NP-hard. However, a simple method to find an approximation is described in [35]. Let $S = s_1, s_2, \ldots, s_k$ be the children of r in the tree, sorted by their size (depending on the size of their children). We have to find the rectangles corresponding to the elements of S in the rectangle of r. s_1 is positioned at the right so that its height is the same as r. Then, for each s_i, we try to position it in the same column as s_{i-1} and in a new column. Then we keep the configuration with the best $width_i/weight_i$ ratio.

In our system (see Fig. 12), we have limited the number of levels displayed to three: The current level, its children, and the classes of its children. Classes were defined in Section 2 as subsets of sequences inferring the same properties as in a human being. In the screenshot Fig. 12, two classes, sick and healthy, can be observed for each sequence represented respectively by red and green rectangles. The hierarchy of sequences is in grey. A checkbox *Classes* can be used to hide the rectangles of the classes. In this case, a third level of the hierarchy is displayed (see Fig. 13).

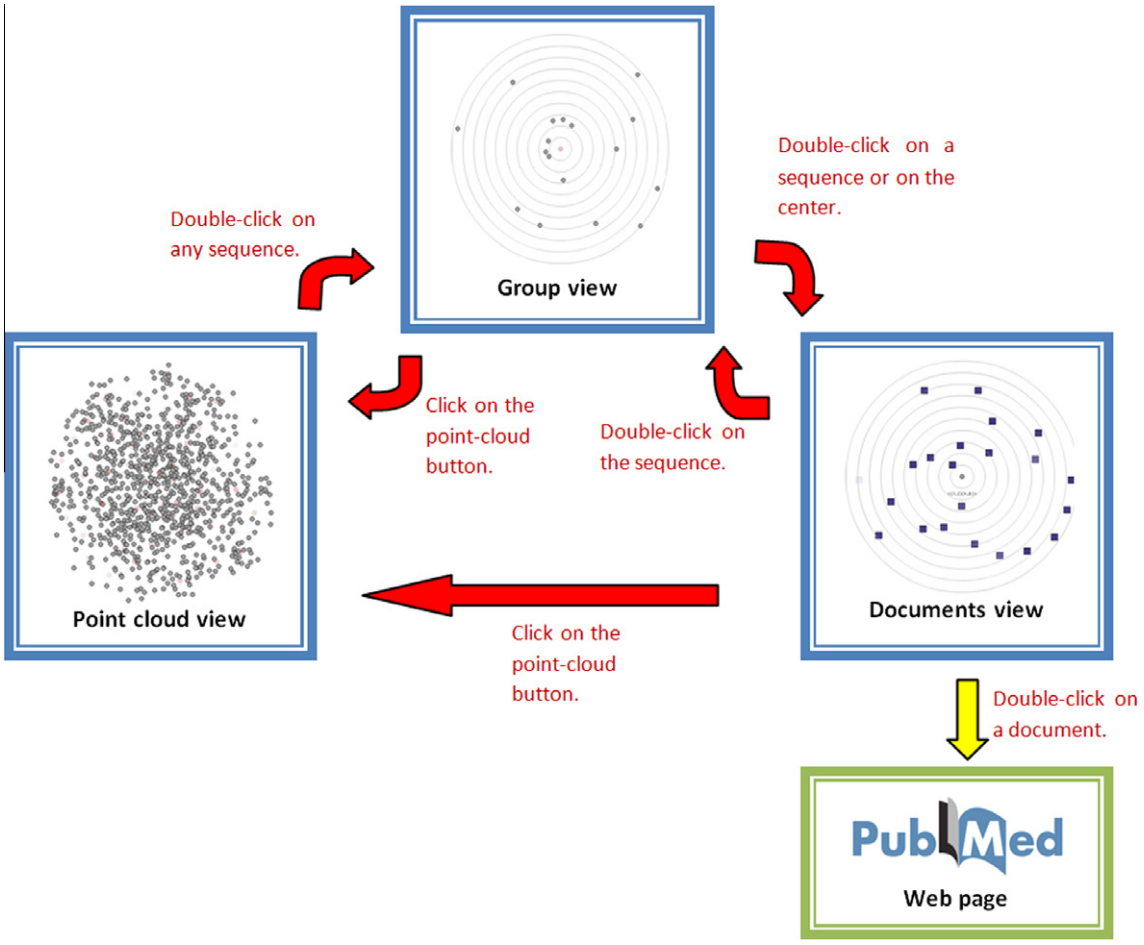

Fig. 11. Description of the navigation through the point cloud and the two solar system views.

Fig. 12. Treemap: the user can observe two levels of the sequence hierarchy and a third level corresponding to the classes of the sequences of the second level.

Double-clicking on a child enables the user to display a lower level. A path corresponding to the current level in the hierarchy from the root can be used to reach a higher level. The path appears at the top of the screen as shown in Fig. 14.

A simple click on a sequence's rectangle opens a tooltip containing the name of the corresponding sequence and a button *Go to point cloud*. Clicking on the button opens the point cloud view where all the sequences containing the sequence selected are highlighted in green like in the search process (see Fig. 5). As an example, in Fig. 15, the sequences beginning with *(C,K,Q)(D,I,C)* will be highlighted in the point cloud.

In the same way, clicking on a sequence in the point cloud or solar system views opens a tooltip (see Fig. 6) containing a button *Go to treemap*. Clicking on the button opens the treemap with the sequence selected and its classes. While sequences appear at the last level of the hierarchy, the path to reach higher levels is represented by buttons like in Fig. 14.

4. Evaluation

The evaluation of our application was undertaken following the nested model for visualization design and validation of Tamara Munzner [36]. According to this model, the visualization creation can be broken down into four nested layers:

1. Layer 1. Algorithm design.
2. Layer 2. Encoding/interaction technique design.
3. Layer 3. Data/operation abstraction design.
4. Layer 4. Domain problem characterization.

The efficiency of the selected algorithms (Layer 1) is discussed in Section 4.1. This evaluation is based on analysis of time complexities and system time measurements. Then, we discuss the effectiveness of the encoding/interaction techniques (Layer 2) testing the application on non-domain users in Section 4.2. Finally, data/operation abstraction's validation (Layer 3) is performed thanks to biologists who test the application in Section 4.3. Based on their profiles, the test subjects fit into one of the two categories summarized in Table 1. Our evaluation protocol for Layers 2 and 3 is summative (i.e. the evaluation is conducted at the end of the design stage of the tool just before its release), experimental (the evaluation is conducted on an usable tool), empirical (the evaluation is based on behavioural knowledge collected when the users actually use the tool) and non-automatic (the observations are made by a human observer). The validation of the characterization of the domain problem (Layer 4) is not treated in the following because it depends on the adoption rates of the system and it is too soon to evaluate it. The protocol and the results of the evaluation of the tree first layers are detailed in the following sections.

4.1. Validation of the algorithms

In this section, the complexity of the algorithms and their limitations are discussed.

4.1.1. Complexity of point cloud

The point cloud remains the most complex view to produce. The calculation of the positions $P^{[t]} = \{p_1^{[t]}, p_2^{[t]}, \ldots, p_n^{[t]}\}$ (see Eq. (2)) needs $O(n^2)$ time where n is the number of centres. We tested the convergence of the iterative process using several random datasets (see Fig. 16). Empirical results indicate that no significant improvement in the placement occurs after 15 steps for each dataset. Thus, the algorithm used for the main placement, and the node overlap removal executes in $O(n^2)$ time.

We already mentioned that a deterministic initial placement is more appropriate than a random one to obtain the same final layout for the same dataset, to make the stress majorization converge quickly and to avoid getting trapped in local minima. Fig. 17 highlights the two last points. The values were computed using the stress function values obtained with a random dataset of 500 centres. We chose the fold-free layout because of the quality of its results and low time complexity ($O(n)$,n is the number of centres).

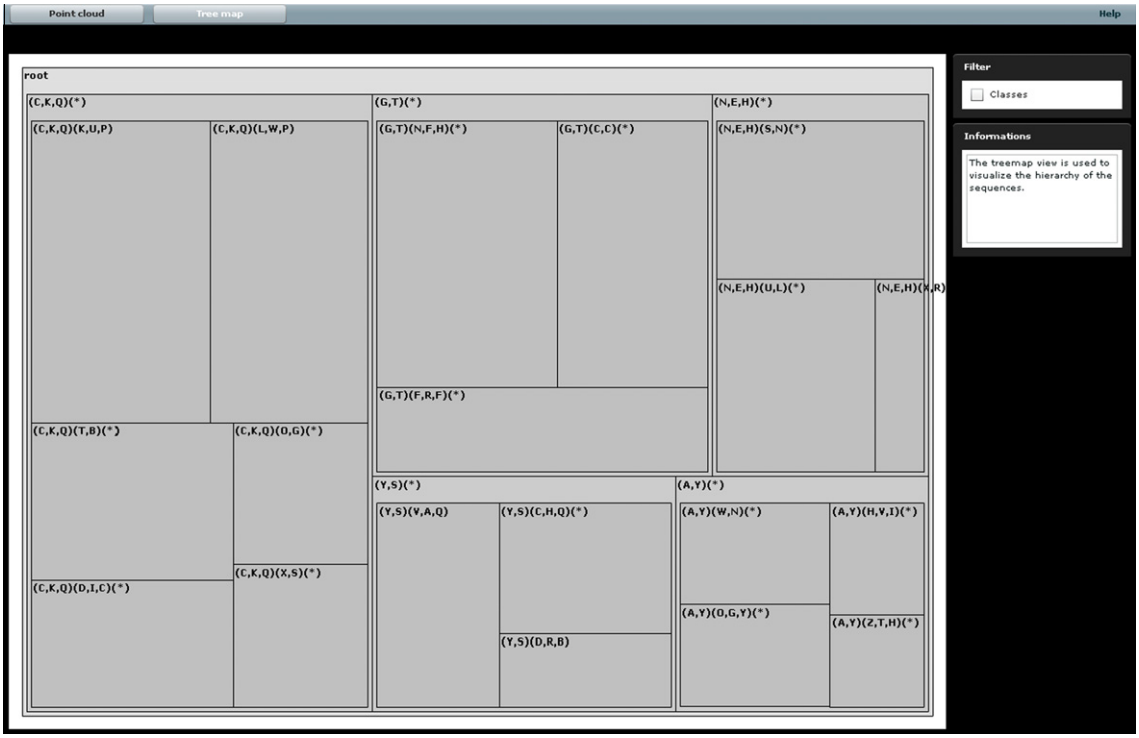

Fig. 13. Treemap: when the user removes the classes using the corresponding checkbox, a third level of the hierarchy is displayed.

Fig. 14. Treemap: Double-clicking on a node enables the user to display a lower level. A path corresponding to the current level in the hierarchy from the root can be used to reach a higher level. The path appears at the top of the screen.

4.1.2. Complexity of the solar system and Treemap

The solar system algorithm is performed in linear time. In the point cloud view, each group is placed using this method. Thus, it runs in $O(n)$ where n is the total number of sequences. The complexity of the solar system view is rather insignificant as the number of sequences/documents is small. The squarified treemap algorithm runs in $O(n)$ where n is the number of edges in the tree displayed. We only compute rectangles for subtrees of the hierarchy with three levels. Thus, the level of complexity is very low.

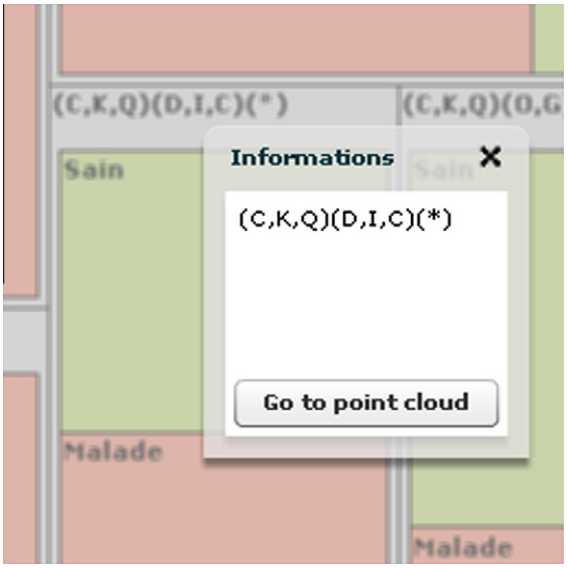

Fig. 15. Treemap: Clicking on a sequence opens a tooltip. Clicking on the button *Go to point cloud* opens the point cloud view where the sequences containing $(C,K,Q)(D,I,C)$ are highlighted in green.

Table 1
This table summarizes the number of participants and their background. To prevent undue advantage and to measure learning time, we checked that each participant had never previously used the tool or a similar tool.

Category	Participant background	Number
Non-domain experts	Volunteers from the university community. We required all users to have at least a Master's degree in informatics and not to be familiar with microarray concepts.	5
Domain experts	Senior researchers with extensive experience in microarray experiments and microarray data analysis.	2

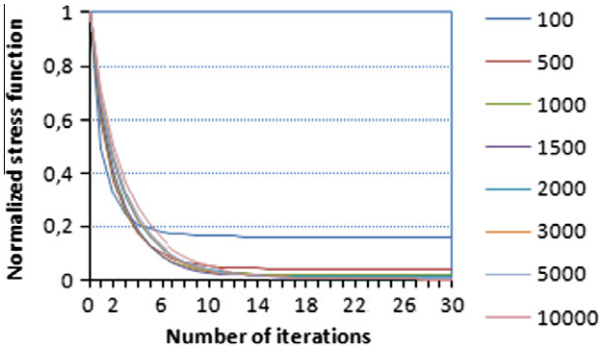

Fig. 16. Convergence of the *stress majorization*: the numbers in the legend correspond to the number of the centres.

4.1.3. Limitations of the visualization

We developed our application in ActionScript 3. The complexity of the point cloud view prevents the user from displaying more than 500 groups. On the other hand, it is possible to visualize up to 25,000 sequences. Unfortunately, the representation of more than 5000 sequences makes navigation slow and tedious. Anyway, cluttering problems occur when more than a thousand of nodes are displayed for a standard resolution. For example, Figs. 18 and 19 show datasets of 992 and 2726 on a 15.4" screen with a resolution

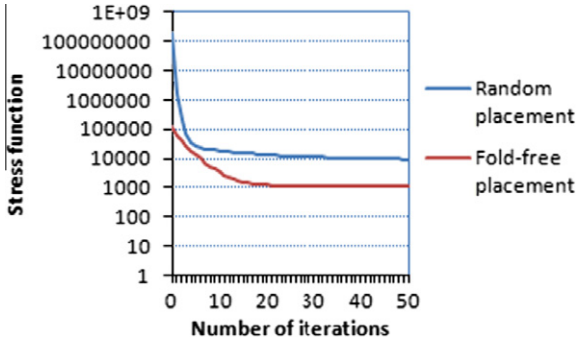

Fig. 17. Convergence of the *stress majorization* using the inital placement *fold-free*.

of 1680 × 1050 pixels. One can change the zoom level to overcome this lack of efficiency: as the sizes of the nodes remain the same, node overlaps are removed. The treemap cannot represent a tree containing more than 4000 nodes.

4.2. Validation of the encoding/interaction techniques

4.2.1. Protocol

To evaluate the effectiveness of the encoding/interaction techniques, a specific protocol and a set of measures were used, inspired by the method presented by [37]. This method combines elements of the controlled experiment and usability-testing methods. This approach seeks to identify individual insights as well as the overall amount of learning that occurs when test subjects manipulate the functionalities. We work with volunteers from the university who are not biology experts. We focus on test subjects with minimal tool training. Before starting their manipulations, users were only given a brief 10-min description of the major functionalities of the tool. Then they were instructed to continue to manipulate the functionalities. Users were allowed to ask questions about the tool if they did not understand a manipulation but we did not note down all their comments until the end of the experiment. When they felt they had finished, users were asked to fill in a form to assess their overall experience with the tool including any difficulties or benefits. With this form, we collected 160 scores about the usefulness and the usability of the three visualizations. Usefulness focuses on how the system answers the user's needs. The user judges the usefulness according to his/her perception of a result/effort ratio. Usability focuses on the ease with which the user used the system: Were the functionalities easy to use and to memorize? Did they include any errors? Did he/she find the system satisfactory? A system can succeed in fulfilling all the criteria of usability, but be completely useless. On the other hand, a system can be useful but too difficult to use. Table 2 summarizes the dependant variables and Table 3 summarizes the evaluated functionalities that correspond to the questions on the form filled in by the users. A score of between 0 and 10 was associated with each answer. Open questions enabled us to obtain more details on the assessment of the functionalities.

4.2.2. Results

In Table 4, results are measured for each functionality in terms of usefulness and usability. The score is an average of the scores given the different users. Since this evaluation method is more qualitative and subjective than quantitative methods and as the number of participants is limited, a general comparison of trends in the results at the level of the general functionalities is more appropriate than a comparison at a more detailed level.

The comments of the non-expert users allowed us to complete these numerical results. Indeed, the two first functionalities

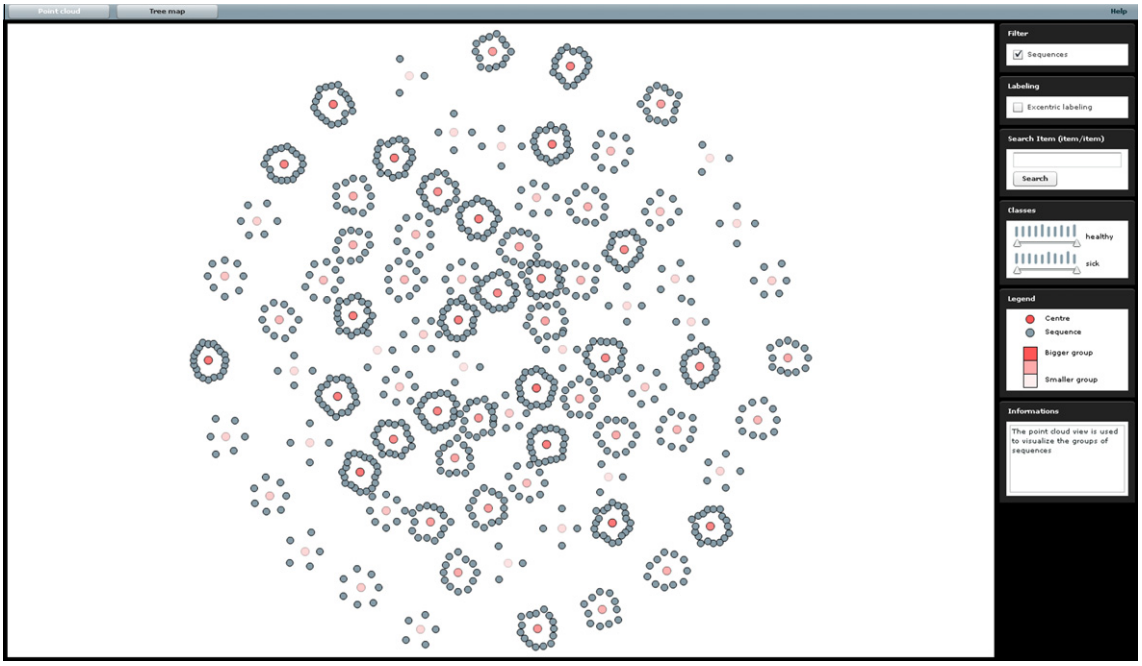

Fig. 18. 992 sequences are displayed on a 15.4" screen with a resolution of 1680 × 1050 pixels: there is no cluttering problems under roughly more than thousand sequences.

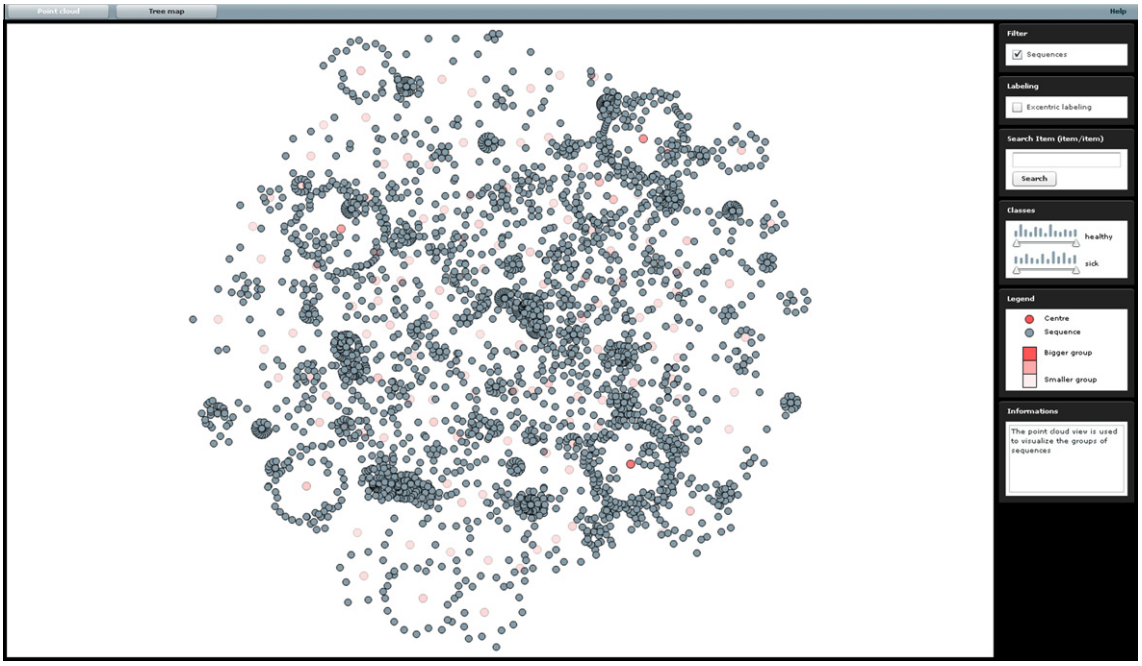

Fig. 19. 2726 sequences are displayed on a 15.4" screen with a resolution of 1680 × 1050 pixels: there is cluttering problems when more than about thousand sequences are displayed.

obtained higher scores because they are extremely simple to use and easy to interpret. The users felt that the information given by the cloud view or the document view is basic but useful to explore sets of sequences. The treemap view was less appreciated in particular because the interest of the visualization of the hierarchy of the sequences was not clear for non-domain users. However, the score was still positive and the fact that this last visualization gives an overview of the distribution of the sequences in the class was underlined as a quality by most of the users.

4.3. Validation of data/operation abstraction

4.3.1. Protocol

Evaluating a visualization system is complex, but in the context of a business application, when experts such as biologists are involved, the evaluation should focus not only on technical and human aspects but also on the impact of the new system on their practice [38]. In our context, the aim of the evaluation of the third layer, based on data/operation abstraction design, was to measure

Table 2
Dependant variables.

1	Participant demographics
2	Total time spent with the tool
3	Background about visualization techniques

Table 3
Evaluated functionalities.

1	Point cloud view
1.1	General visualization
1.2	Placement of the clusters
1.3	Zoom interaction
1.4	Colour code about the volume of a group
1.5	Research of sequence by gene
1.6	Research of sequence by class
1.7	Information about a sequence
1.8	Adequacy of the representation for huge volumes of sequences
2	Solar system for documents view
2.1	General visualization
2.2	Placement of the documents
2.3	Colour code associated with the year of publication
2.4	Information about a document
3	Treemap view
3.1	General visualization
3.2	Placement of the squared
3.3	Colour code associated with the class
3.4	Summary efficiency

Table 4
Average of the marks (on 10) given to the SequencesViewer tool (encoding/interaction techniques).

Functionalities	Average of usefulness marks (/10)	Average of usability marks (/10)
Point cloud	6.68	7.46
Solar system	7.60	7.50
Treemap	6.39	6.53

to what extent our tool answered the needs of two teams of biologists. To this end, we undertook a semi-realistic evaluation in collaboration with potential users to check the interest of the three visualizations.

The domain experts manipulated the patterns obtained from their own microarray datasets. Even though the data used as inputs in the system were their own, patterns were new material that biologists were not used to manipulate. Consequently, we prefer to use the expression "semi-realistic evaluation" to describe our experiments.[4]

We collaborated with two laboratories to select a relevant dataset and build the protocol. We implemented this protocol with the team working on Alzheimer's disease. The evaluation protocol is based on a cooperative technique which enabled us to collect 104 scores. This protocol is a variant of the "think aloud" method during which an observer asks users to use the tool and encourages them to think aloud when interacting with the system. It is called cooperative because the observer does not remain silent during the evaluation process but guides, explains and questions the user. A cooperative assessment enables interaction with the user and enables the tool to be evaluated in controlled conditions, and the user's perception of the different functionalities to be recorded. Questionnaires were used to complement experimental methods. They quantified the users' impressions before and after they used the system (satisfaction, anxiety, etc.) and often helped them to take a step back because they were no longer actually using the tool.

Our protocol was submitted to the team of biologists working on Alzheimer's disease. The interview lasted approximately three hours per biologist. At the beginning of the test, we invited the users to fill in a pre-evaluation form. We used this form to identify the biologists' profile, data-processing competences, and current use of visualization tools. We only gave them a very brief demonstration of the tool because we wanted them to discover its functionalities on their own. We asked them to perform some tasks based on realistic scenarios. In so doing, they used the functionalities just as they would do for work. During the test, one observer guided the user and observed the way he/she used the functionalities. A second observer noted down the information given orally by the user, and his/her reactions and gestures. At the end of the test, the users filled in a post-evaluation form which focused on the same points detailed in Table 3.

4.3.2. Results

A successful system should be both useful and usable. The evaluation summarized in Table 5 revealed the quality of our system, especially the solar and the treemap systems. Unlike other users, experts are more interested in the hierarchy of sequences which provides them with valuable information particularly because it is linked to the classification of the sequences into classes.

The tool allowed biologists to gain new insights. For instance, with the cloud visualization, the expert visualized all the sequences containing the gene A2M which is known to be implicated in the Alzheimer's disease. He focused on: $S75 = \langle (MRVI1)(PGAP1)(PLA2R1)(A2M)(GSK3B)\rangle$. This pattern is particularly interesting because it connects the proteins involved in signalling mechanisms and metabolism. Some proteins interfere with cellular events in Alzheimer's disease. The tool was effective in the discovery process because it allowed the expert to identify certain combinations of genes in the patterns. It also allowed him to test their hypothesis through the interface of visualization of the documents. Patterns containing genes linked to Alzheimer's disease in the literature and new genes are particularly interesting because they could be the subject of future research.

Two future directions are envisaged: (1) As organizing the sequences in groups based on their similarity did not prove to be useful to the biologists, other types of organization based on a discrimination measure of a sequence for example, may prove to be more useful; (2) We will include other criteria to identify the most relevant documents associated with a sequence (e.g. the species involved in the studies, or the type of document). We are currently working with the second team of biologists, who also use DNA microarrays but in their case to study breast cancer. This second evaluation will help us generalize our initial results. Indeed, we need to take into account the specificity of the functionalities evaluated, their specific context of use depending on the expert domain, and the context of the evaluations themselves.

5. Conclusions

In this paper, we describe a new approach to help biologists access and interpret sequential patterns extracted from DNA microarrays. Our system was developed in collaboration with biologists and with the Pikko[5] company, which is specialized in information visualization.

We combined and adapted three techniques from the information visualization domain. A point cloud view provides experts

[4] We could have used "realistic evaluation" if we had observed them manipulating their own data with their own tools but it was not the case.

[5] http://www.pikko-software.com/

Table 5
Average of the marks (on 10) given to the SequencesViewer tool.

Functionalities	Average of the utility marks (/10)	Average of the utilisability marks (/10)
Point Cloud	6.00	6.34
Solar System	7.5	7.40
Treemap	7.76	7.00

with a global representation of the sequential patterns. Combined with a first solar system view, it helps the biologist to navigate through groups of patterns and to compare and evaluate the relevance of the discovery correlations. Users can also access publications concerning each gene sequence through a second solar system view. This functionality increases the rapidity of searches and makes them less tedious. Finally, a treemap view gives a new perspective of the data allowing users to navigate through the hierarchy of the sequences. The algorithms we used were selected on the basis of their efficiency and their limited complexity. A video summarizing functionalities of SequencesViewer is available at http://www.lirmm.fr/tatoo/jbi/.

The overall scores obtained with the two kinds of experiments (Sections 4.2 and 4.3) show that SequencesViewer system was appreciated by users. The solar system was shown to be of real interest in terms of data abstraction (assessed by expert biologists) and technical interactions (assessed by non-experts). It should be noted that the treeMap is appreciated by biologists because it enables a better understanding of specialized data taking into account the hierarchy.

Our future work will focus on three main points: improving, testing and generalizing the technique. The first point will add more sophisticated interactive techniques. As an example, more dynamic widgets in the legend (see [29] for examples) could be proposed to support Dynamics Queries [39]. We also plan to endow the system with multiple coordinated views [40] to help the user to keep an overview of the data while he is performing a particular task. We will then analyze more tests to evaluate the efficiency of the application in collaboration with biologists working on cancer. After which we will look for other biological datasets with the associated requirements to obtain a generic user-oriented application for bio-researchers. Indeed, many data-mining algorithms used in different domains of application produce large amounts of information that cannot be used directly by experts. Whether our application is useful and adaptable to other data sets needs to be evaluated.

Acknowledgment

We would like to thank Guillaume Aveline and Faraz Zaidi for their technical assistance and Jarlath Slevin for his help in the software video recording. We would also like to thank the members of the Pikko company who provided our material resources.

References

[1] Hoerndli F, David D, Götz J. Functional genomics meets neurodegenerative disorders part ii: Application and data integration. Prog Neurobiol 2005;76:169–88.
[2] Cong GA, Tung X, Pan F, Yang J. Farmer: finding interesting rule groups in microarray datasets. In: SIGMOD conference; 2004. p. 143–54.
[3] Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat Med 2001;76:673–9.
[4] Pensa R, Besson J, Boulicaut JF. A methodology for biologically relevant pattern discovery from gene expression data. In: Discovery science, LNCS, vol. 3245; 2004. p. 230–41.
[5] Korotkiy M, Middelburg R, Dekker H, Harmelen FV, Lankelma J. A tool for gene expression based pubmed search through combining data sources. Bioinformatics 2004;20(12):1980–2.
[6] Salle P, Bringay S, Teisseire M, Chakkour F, Roche M, Rassoul RA, et al. Genemining: identification, visualization, and interpretation of brain ageing signatures. In: MIE; 2009a. p. 767–71.
[7] Salle P, Bringay S, Teisseire M. Mining discriminant sequential patterns for aging brain. In: AIME '09: proceedings of the 12th conference on artificial intelligence in medicine; 2009b. p. 365–9.
[8] Tanabe L, Scherf U, Smith LH, Lee JK, Hunter L, Weinstein JN. Medminer: an internet text-mining tool for biomedical information, with application to gene expression profiling. Biotechniques 1999;27:210–4.
[9] Zeeberg BR, Feng W, Wang G, Wang MD, Fojo AT, Sunshine M, et al. Gominer: a resource for biological interpretation of genomic and proteomic data. Genome Biol 2003;4:1–28.
[10] Saneifar H, Bringay S, Laurent A, Teisseire M. S2mp: Similarity measure for sequential patterns. In: AusDM; 2008. p. 95–104.
[11] Nin J, Salle P, Bringay S, Teisseire M. Using owa operators for gene sequential pattern clustering. In: 22nd IEEE international symposium on computer-based medical systems; 2009.
[12] Shneiderman B. The eyes have it: a task by data type taxonomy for information visualizations. In: VL; 1996. p. 336–43.
[13] Chi EH, Riedl J, Shoop E, Carlis JV, Retzel E, Barry P. Flexible information visualization of multivariate data from biological sequence similarity searches. In: IEEE visualization; 1996. p. 133–40.
[14] Smoot ME, Bass EJ, Guerlain SA, Pearson WR. A system for visualizing and analyzing near-optimal protein sequence alignments. Inf Vis 2005;4(3):224–37.
[15] Lungu M, Xu K. Biomedical information visualization. In: Kerren A, Ebert A, Meyer J, editors. Human-centered visualization environments. Lecture Notes in Computer Science, vol. 4417. Springer; 2006. p. 311–42.
[16] Piipari M, Down TA, Saini H, Enright A, Hubbard TJ. imotifs: an integrated sequence motif visualization and analysis environment. Bioinformatics 2010;6(26):843–4.
[17] Perlin K, Fox D. Pad: an alternative approach to the computer interface. In: SIGGRAPH '93: proceedings of the 20th annual conference on computer graphics and interactive techniques. New York, NY, USA: ACM; 1993. p. 57–64. ISBN:0-89791-601-8, <http://doi.acm.org/10.1145/166117.166125>.
[18] Stasko J, Görg C, Liu Z. Jigsaw: supporting investigative analysis through interactive visualization. Inf Vis 2008;7(2):118–32. http://doi.acm.org/10.1145/1466620.1466622.
[19] Hearst MA. Tilebars: visualization of term distribution information in full text information access. In: CHI '95: proceedings of the SIGCHI conference on human factors in computing systems; 1995. p. 59–66. ISBN:0-201-84705-1.
[20] Sallaberry A, Pecheur N, Bringay S, Roche M, Teisseire M. Discovering novelty in gene data: from sequential patterns to visualization. In: ISVC, vol. 3; 2010. p. 534–43.
[21] Brog I, Groenen P. Modern multidimensional scaling: theory and applications. New York: Springer-Verlag; 1997. ISBN:0-387-94845-7.
[22] Torgerson WS. Multidimensional scaling I. Theory and method. PSym 1952;17:401–19.
[23] deLeeuw J. Applications of convex analysis to multidimensional scaling. In: Recent developments in statistics (Proc. European meeting statisticians, Grenoble, 1976). Amsterdam: North-Holland; 1977. p. 133–45.
[24] Gansner, Koren, North. Graph drawing by stress majorization. In: GDRAWING: conference on graph drawing (GD); 2004.
[25] de Leeuw J. Convergence of the majorization method for multidimensional scaling. J Classif 1988;5(2):163–80.
[26] Priyantha NB, Balakrishnan H, Demaine ED, Teller SJ. Anchor-free distributed localization in sensor networks. In: Akyildiz IF, Estrin D, Culler DE, Srivastava MB, editors. SenSys. ACM; 2003. p. 340–1.
[27] Gansner ER, Hu Y. Efficient node overlap removal using a proximity stress model. In: Tollis IG, Patrignani M, editors. Graph drawing. Lecture notes in computer science, vol. 5417. Springer; 2008. p. 206–17.
[28] Delaunay B. Sur la sphère vide. Izvestia Akademia Nauk SSSR, VII Seria. Otdelenie Matematicheskii i Estestvennyka Nauk 1934;7:793–800.
[29] Willett W, Heer J, Agrawala M. Scented widgets: improving navigation cues with embedded visualizations. 2007;13:1129–36.
[30] Fekete JD, Plaisant C. Excentric labeling: dynamic neighborhood labeling for data visualization. In: CHI; 1999. p. 512–9.
[31] Nguyen T, Zhang J. A novel visualization model for web search results. IEEE Trans Vis Comput Graph 2006;12(5):981–8. <http://doi.ieeecomputersociety.org/10.1109/TVCG.2006.111>.
[32] Jacquemin C, Folch H, Garcia K, Nugier S. Visualisation interactive d'espaces documentaires. Inf Interact Intell 2005;5(1):59–84.
[33] Reingold EM, Tilford JS. Tidier drawings of trees. IEEE Trans Softw Eng 1981;7(2):223–8.
[34] Johnson B, Shneiderman B. Tree maps: a space-filling approach to the visualization of hierarchical information structures. In: IEEE visualization; 1991. p. 284–91.
[35] Bruls M, Huizing K, van Wijk JJ. Squarified treemaps. In: Proc joint Eurographics/IEEE TVCG symp visualization, VisSym; 2000. p. 33–42.
[36] Munzner T. A nested process model for visualization design and validation. IEEE Trans Vis Comput Graph 2009;15(6):921–8.
[37] Saraiya P, North C, Duca K. An insight-based methodology for evaluating bioinformatics visualizations. IEEE Trans Vis Comput Graph 2005;11:443–56. doi:10.1109/TVCG.2005.53. <http://portal.acm.org/citation.cfm?id=1070610.1070747>.

[38] Ammenwerth E. Can evaluation studies benefit from triangulation? a case study. Int J Med Inform 2003;70(2-3):237–48.
[39] Shneiderman B. Dynamic queries for visual information seeking. IEEE Softw 1994;11:70–7.
[40] Roberts JC. State of the art: coordinated & multiple views in exploratory visualization. In: Proceedings of the 5th international conference on coordinated & multiple views in exploratory visualization (CMV2007). IEEE Computer Society Press; 2007. p. 61–71.