Discovering Novelty in Gene Data : From Sequential Patterns to Visualization

Arnaud Sallaberry, Nicolas Pecheur, Sandra Bringay, Mathieu Roche, Maguelonne Teisseire

Discovering Novelty in Gene Data: From Sequential Patterns to Visualization

Arnaud Sallaberry[1], Nicolas Pecheur[2], Sandra Bringay[3], Mathieu Roche[2], and Maguelonne Teisseire[4]

[1] LaBRI & INRIA Bordeaux - Sud Ouest & Pikko, France
arnaud.sallaberry@labri.fr
[2] LIRMM - Université Montpellier 2, France
{pecheur,mathieu.roche}@lirmm.fr
[3] LIRMM - Université Montpellier 3, France
bringay@lirmm.fr
[4] CEMAGREF - UMR TETIS, France
maguelonne.teisseire@cemagref.fr

Abstract. Data mining techniques 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 analyse by end-users. In this paper, we focus on sequential pattern mining and propose a new visualization system, which aims at helping end-users to analyse extracted knowledge and to highlight the novelty according to referenced biological document databases. Our system is based on two visualization techniques: Clouds and solar systems. 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 Alzheimers disease and cancer.

1 Introduction

DNA microarrays have been successfully used for many applications (diagnosis and characterisation of physiological states). They allow researchers to compare gene expression in different tissues, cells, or conditions 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, the way to process and interpret them in order to make biomedical sense of them remains a challenge. Data mining techniques, such as [1], have played a key role in discovering previously unknown information and shown that they could be very useful to biologists to identify subsets of microarray data, which could be relevant for further analysis.

However, the number of results is usually so huge that they cannot easily be analysed by the experts concerned. In [2], the authors propose a general process, called GeneMining, based on the DBSAP algorithm for extracting sequential

G. Bebis et al. (Eds.): ISVC 2010, Part III, LNCS 6455, pp. 534–543, 2010.

patterns from DNA microarrays. They obtained patterns of correlated genes ordered according to their level of expression. Although this method is useful, the way to select relevant patterns remains inefficient. For instance, depending on the values of parameters, they extract between 1,000 and 100,000 patterns that are not easy to interpret. Thus, the main aim of this work is to propose new visualization techniques to help biologists to navigate through the 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., [3]), there are still no user-friendly literature search tools available for analysing patterns.

In this paper, we present 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 two different techniques (i.e., point clouds and solar systems) to deal with data organized as a sequence and to produce an effective solution to the problem mentioned above. Second, using our system, the biologist can now be automatically provided with relevant documents extracted from the PubMed repository. Although the methods described in this paper mainly focus on sequences extracted from DNA arrays, they could easily be adapted to any other kind of sequential data.

The paper is organized as follows. In Section 2, we present the data we are working with and give an overview of related work. In Section 3, we describe our proposal and the associated tool. The evaluation of our systems are discussed in Sections 4 and 5. Section 6 concludes.

2 Preliminaries

In the framework of the PEPS-ST2I Gene Mining project[1], 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 [4]. This dataset was used to discover classification tools to distinguish between two *classes* (AD animals and healthy animals). In [4], the authors proposed to extract patterns of correlated genes ordered according to their level of expression. An example of pattern is $< (MRVI1)(PGAP1, GSK3B) >$ meaning that "the level of expression of gene $MRVI1$ is lower than that of genes $PGAP1$ and $GSK3B$, whose levels are very similar".

Although this method was interesting since they 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, 1,000 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 innovative sequences (e.g. sequences where new gene correlations may exist), and to automatically

[1] This work was conducted in collaboration with the MMDN 'Molecular mechanisms in neurodegenerative dementias' laboratory, University of Montpellier 2, France. `http://www.mmdn.univ-montp2.fr`

query specific publications from Pubmed (or other publication database) on the selected genes. To summarize, an appropriate visualization tool needs to explore two 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 [5]), we are able to identify groups of similar sequences and highlight a representative sequence called the center.
2. Documents in the literature dealing with genes from sequences. The documents are obtained from the Pubmed bibliographical database with or without gene synonyms [2]. 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 concerned sequence.

The visualization tool, which is described in the following section, combines all these elements: Support, class, groups, and sets of documents. To facilitate specific tasks, it proposes two different visual representations [6]. 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.

In [7], 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 take into account the hierarchy of sequences. Indeed, most previous works concerning visualization of biological sequences mainly focus on the representation of sequence alignments [8]. To the best of our knowledge, no method is currently available to visualize sequences and associated documents.

3 SequencesViewer

SequencesViewer[2] helps biomedical experts to browse and explore sequences of genes. In the following we describe the main representations.

3.1 Point Cloud

The Point Cloud representation allows biologists to visualize groups of gene sequences (see figures 1, 2). It gives an overview of the centers of the groups, the distance from the centers, and associated sequences. Three steps are required to compute relevant positions of centers 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.

[2] All the screenshots of our system are available on the following web page: `http://www.labri.fr/perso/sallaber/publications/isvc10/SequencesViewer.html`

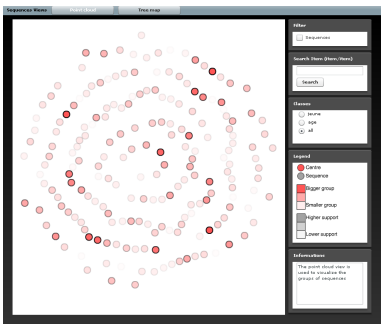

Fig. 1. Point cloud without sequences

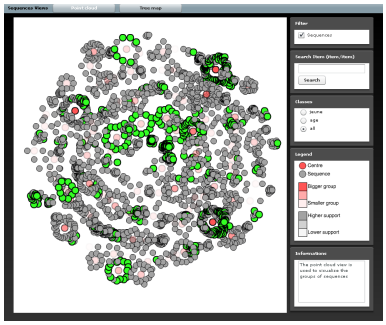

Fig. 2. Point cloud with sequences and highlighted researched items

Main placement of the centers. The basic idea is to place the centers 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 [5]. Let d_{ij} be the matrix value for a center i and a center j. We want to find the coordinates $p_i = (x_i, y_i)$ for each center i so that $\|p_i - p_j\| \approx d_{ij}$ where $\|p_i - p_j\|$ is the Euclidean distance between the centers i and j.

Different techniques are described in the literature to assign a location to items in an N-dimensional space. Multidimensional Scaling (MDS) technique [9] is often used in information visualization. It 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, 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:

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

where $\omega_{ij} = d_{ij}^{-\alpha}$ and $p = p_1, p_2, ..., p_n$ is the actual configuration. We use $\alpha = 2$, which appears to produce good results in most cases, as shown by [10].

Several techniques have been developed to minimize the stress function (see [9] for an overview). In our application, we use a method introduced in [10] due to its simplicity, fast convergence, and quality of results. It consists of successively computing a simple function that returns position p_i:

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

where $p_i^{[t]}$ is the position of the center i at time t and $s_{ij}^{[t]} = \frac{d_{ij}}{\|p_i^{[t]} - p_j^{[t]}\|}$ if $\|p_i^{[t]} - p_j^{[t]}\| \neq 0$ or $s_{ij}^{[t]} = 0$ otherwise.

This iterative updating is performed for each node and repeated until a stable configuration is reached. At each step, $\sigma(p)^{[t]} \geq \sigma(p)^{[t+1]}$ and the stress function converges to a local minimum [11].

Initial placement of the centers. One important aspect of these methods is to find an initial placement of the centers before performing the iterative process. Random placement is not efficient because every 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 [12]. The algorithm selects four centers 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 to be roughly perpendicular to the pair (c_3, c_4) in the layout. A fifth center c_5 is selected so that it lies in the middle of the point cloud. A complete description of the selection process is available in [12].

Then, let x_i be $d_{c_3 i} - d_{c_4 i}$ and let y_i be $d_{c_1 i} - d_{c_2 i}$. We can use (x_i, y_i) coordinates directly to place each center i. Unfortunatly, this solution disregards the distance between i and c_5. To overcome this lack, the method computes the polar coordinate (ρ_i, θ_i) of a center i so that $\rho_i = d_{c_5 i} \times R$ and $\theta_i = tan^{-1}\left(\frac{y_i}{x_i}\right)$.

Removal of center overlap. The MDS method we have implemented does not avoid overlapping of centers. 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 [13] implemented a simple but effective solution based on a nice adaptation of the stress majorization process.

This solution is based on a Delaunay triangulation computed for the set of centers 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 centers as a planar graph $G(V, E)$ where V is the set of the centers 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: $t_{ij} = \max\left(\frac{a_i + a_j}{\|p_i - p_j\|}, 1\right)$ where a_i is the radius of the center i. $t_{ij} = 1$ if the centers 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 equation 2) with d_{ij}^{DT} and s_{ij}^{DT} in spite of d_{ij} and s_{ij}.

$$\sigma^{DT}(p) = \sum_{i<j\leq n} \omega_{ij}(d_{ij}^{DT} - \|p_i - p_j\|)^2 \tag{3}$$

Interactions and navigation. The user can choose to visualize centers (figure 2) or both centers and their associated sequences using the check box labelled *Sequences*. The color of the centers is of different intensity which is proportional to the number of sequences associated with the center. The legend on the right helps the user evaluate the size of the groups. The research can be refined by applying different filters. First, sequences can be hidden by clicking on a filter button (see figure 1). An item can be searched and the sequences containing the search term are highlighted. The screenshot in figure 2 represents a map with the highlighted sequences (in green) resulting from a search operation.

3.2 Solar System

When the user double-clicks on a center in the point cloud view, he/she accesses a second view (figure 3) based on a solar system metaphor [14]. This view allows only the group of the selected center to be explored. The user can Zoom In/Out, move the whole map, search for an item or display a tooltip. The legend is also available. The center of the group is positioned in the middle of the visualization area (position $(0, 0)$). Then, each sequence i is placed as follows: $\theta_i = i \cdot \frac{2\pi}{n}$ where n is the number of sequences.

A second solar system based view is reachable from the first one by double-clicking on a sequence (figure 4). This view represents the sequence and its associated set of text documents, i.e. scientific papers dealing with the genes of the concerned sequence. These papers are extracted from the Pubmed biomedical library.

The sequence is positioned in the middle of the visualization as the center 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 color 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.

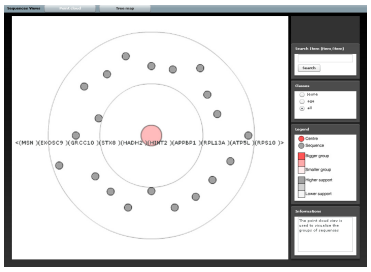

Fig. 3. Group of sequences

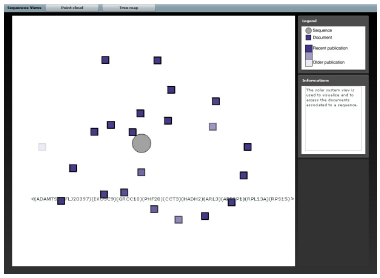

Fig. 4. Sequence of genes and its associated documents

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 that interest him/her. Of course, other ways of visualizing text documents are described in the literature. There are two main approaches: The visualization of specific subsections contained in large documents or the visualization of clustered collections [15]. Here, we focus on the second situation.

4 Discussion

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

Complexity of Point Cloud. The point cloud remains the most complex view to produce. The calculation of the positions $P^{[t]} = \{p_1^{[t]}, p_2^{[t]}, ..., p_n^{[t]}\}$ (see equation 2) needs $O(n^2)$ time where n is the number of centers. We tested the convergence of the iterative process using several random datasets (see figure 5). 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. Figure 6 highlights the two last remarks. The values were computed using the stress function values obtained with a random dataset of 500 centers. We chose the fold-free layout because of the quality of its results and low time complexity ($O(n)$, n is the number of centers).

Complexity of the Solar System. 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.

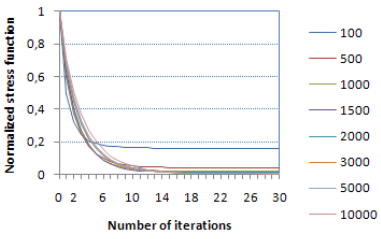

Fig. 5. Convergence of the *stress majorization*: the numbers in the legend correspond to the number of the centers

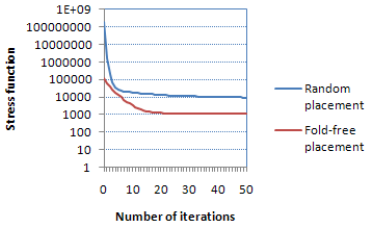

Fig. 6. Convergence of the *stress majorization* using the inital placement *fold-free*

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 5,000 sequences makes the navigation slow and tedious.

5 Evaluation

Evaluating a visualization system is complex, but in the context of a business application, when experts such as biologists and health professionals 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 [16]. In our context, the aim of the evaluation was to measure to what extent our tool answered the needs of two teams of biologists. We undertook a semi-realistic evaluation in collaboration with the potential users to check the interest of the two visualizations. We worked with two laboratories on building the protocol and implemented it with the team working on Alzheimer's disease.

Our evaluation protocol is summative (i.e. the evaluation is conducted at the end of the design stage of the tool and 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 evaluation is based on a cooperative technique. This one is a variant of the "think aloud" method during which an observer asks the user to use the tool and encourages him/her 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 1) interactions between the user and the tool to be evaluated in controlled conditions and the user's perception of the different functionalities to be recorded; 2) Questionnaires are used to complement experimental methods. They quantify the user's impressions before and after using the system (satisfaction, anxiety, etc.) and often help him to take a step back because he/she is no longer involved in handling the tool.

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

With this evaluation, we collected 104 marks about the usefulness and the usability of the system (actually three tools have been evaluated). Usefulness focuses on how the system answers the user's needs. The user judges the usefulness according to his perception of a result/effort ratio. Usability focuses on the ease with which the expert used the system: Were the functionalities easy to use and to memorize? Did they include any errors? Did he 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. A successful system should be both useful and usable. This evaluation revealed the quality of our system, especially the solar system graded 3.75/5 for its utility and 3.70/5 for its ease (3.00/5 and 3.17/5 respectively for the point cloud). Two future directions are envisaged: 1) Organizing the sequences into groups according to their similarity did not prove to be useful to the users. Other types of organization, for example based on a discrimination measure of a sequence, may be more useful; 2) We will integrate other criteria to identify the most relevant documents associated with a sequence (e.g. the species involved in the studies, or the type of the document).

We are currently working with the second team of biologists, which also use DNA microarrays but to study breast cancer. This second evaluation will help us to generalize these first results. Indeed, we need to take into account the specificity of the evaluated functionalities, their specific context of use depending on the experts domain and the context of the evaluations themselves.

6 Conclusion

In this paper, we describe a new approach that helps the biologists to access and interpret sequential patterns extracted from DNA microarrays. Our system was developed in collaboration with biologists and with Pikko[3] company (specialized in information visualization). We combined and adapted two techniques from the information visualization domain. A point cloud view provides experts 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 improves the rapidity of searches and makes them less tedious. The algorithms we used were selected on the basis of their efficiency and their low complexity.

Our future work will be aimed at analysing the tests to evaluate the efficiency of the application in collaboration with biologists. After which, we will look for other applications to test its generalizability. 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.

[3] `http://www.pikko-software.com/`

Acknowledgments. We would like to thank Mr. Guillaume Aveline and Mr. Faraz Zaidi for their technical assistance and the members of the Pikko company who provided us material resources.

References

1. Cong, G.A., Tung, X., Pan, F., Yang, J.: Farmer: Finding interesting rule groups in microarray datasets. In: SIGMOD Conference, pp. 143–154 (2004)
2. Salle, P., Bringay, S., Teisseire, M., Chakkour, F., Roche, M., Rassoul, R.A., Verdier, J.M., Devau, G.: Genemining: Identification, visualization, and interpretation of brain ageing signatures. In: MIE, pp. 767–771 (2009)
3. Zeeberg, B.R., Feng, W., Wang, G., Wang, M.D., Fojo, A.T., Sunshine, M., Narasimhan, S., Kane, D.W., Reinhold, W.C., Lababidi, S., Bussey, K.J., Riss, J., Barrett, J.C., Weinstein, J.N.: Gominer: a resource for biological interpretation of genomic and proteomic data. Genome Biol. 4, 28 (2003)
4. Salle, P., Bringay, S., Teisseire, M.: Mining discriminant sequential patterns for aging brain. In: Combi, C., Shahar, Y., Abu-Hanna, A. (eds.) Artificial Intelligence in Medicine. LNCS, vol. 5651, pp. 365–369. Springer, Heidelberg (2009)
5. Saneifar, H., Bringay, S., Laurent, A., Teisseire, M.: S2mp: Similarity measure for sequential patterns. In: AusDM, pp. 95–104 (2008)
6. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: VL, pp. 336–343 (1996)
7. Chi, E.H.-h, Riedl, J., Shoop, E., Carlis, J.V., Retzel, E., Barry, P.: Flexible information visualization of multivariate data from biological sequence similarity searches. In: IEEE Visualization, pp. 133–140 (1996)
8. Lungu, M., Xu, K.: Biomedical information visualization. In: Kerren, A., Ebert, A., Meyer, J. (eds.) GI-Dagstuhl Research Seminar 2007. LNCS, vol. 4417, pp. 311–342. Springer, Heidelberg (2007)
9. Brog, I., Groenen, P.: Modern multidimensional scaling: Theory and applications. Springer, New York (1997)
10. Gansner, E.R., Koren, Y., North, S.: Graph drawing by stress majorization. In: GDRAWING: Conference on Graph Drawing (GD) (2004)
11. de Leeuw, J.: Convergence of the majorization method for multidimensional scaling. J. Classification 5, 163–180 (1988)
12. Priyantha, N.B., Balakrishnan, H., Demaine, E.D., Teller, S.J.: Anchor-free distributed localization in sensor networks. In: Akyildiz, I.F., Estrin, D., Culler, D.E., Srivastava, M.B. (eds.) SenSys, pp. 340–341. ACM, New York (2003)
13. Gansner, E.R., Hu, Y.: Efficient node overlap removal using a proximity stress model. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 206–217. Springer, Heidelberg (2009)
14. Nguyen, T., Zhang, J.: A novel visualization model for web search results. IEEE Trans. Vis. Comput. Graph 12, 981–988 (2006)
15. Jacquemin, C., Folch, H., Garcia, K., Nugier, S.: Visualisation interactive d'espaces documentaires. Information Interaction Intelligence 5, 59–84 (2005)
16. Ammenwerth, E.: Can evaluation studies benefit from triangulation? a case study. International Journal of Medical Informatics 70, 237–248 (2003)