3D VISUALIZATION OF GENE CLUSTERS

Leishi Zhang, Xiaohui Liu and Weiguo Sheng
Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.

Abstract: An essential step in the analysis of gene expression profile data is the detection of gene groups that have similar expression patterns. Although many clustering algorithms have been proposed for such task, problems such as visualizing the clustering results are still not satisfactorily addressed. In this paper, a novel methodology for drawing the gene clusters in 3D is proposed. The algorithm firstly allocates the genes within a cluster to a local area – InfoCube using Force-Directed Placement Spring Model; it then allocates all the InfoCubes within a global area using the same method. The bottom-up approach saves time in coordinates' computation and successfully avoids the space partition problem in multi-layer graph drawing. It is not only effective in displaying the double-layer clustering results but also can be extended to display other multi-layer graphs with hierarchical relationships.

Key words: gene cluster, visualization, graph layout

1. INTRODUCTION

The most common method for analyzing gene expression data is clustering that groups together genes with similar expression profiles. Genes that are similarly expressed often participate in the same cellular processes, so clustering suggests functional relationships between the clustered genes. The outputs of clustering algorithms provide the basis for understanding the biological process underlying the data. However, to extract concrete biological understanding, further time-consuming post-analysis of such outputs is required. This post-analysis usually focuses on relating gene expression patterns with other form of biological knowledge. This type of analysis requires a multitude of scripts, visualizations, comparisons to multiple biological databases, and more[1].

K. Wojciechowski et al. (eds.), Computer Vision and Graphics, 349–354.

Visualizations use computer graphics to present data or information in visual form. Most techniques currently applied in gene clusters visualization are in two dimensions, they are restrictive in terms of display space and they do not correlate with the way people see the world. In this paper, we attempt to solve this problem by apply novel and effective 3D visualization techniques instead of traditional 2D techniques to display the gene clusters. 3D portrayal may help to "open-up" a model of the system contents so that the items and their inter-relationships can be perceived more easily[2].

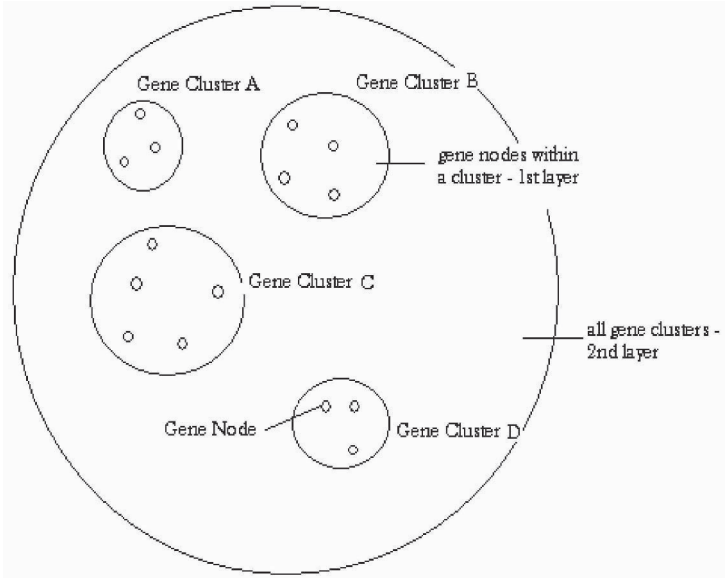

Figure 1. Two Layer Graph Representation of Gene Clusters.

Problems have been raised when we try to visualize the clustering results in 3D. The output of the clustering algorithms is basically a list of clusters each of which consists of a number of genes. It should be considered as a double layer graph where genes within a cluster form the first layer and the gene clusters form the second layer (see Figure 1). Traditional Force-Directed placement layout algorithm has been proven to be effective to allocate objects into 2D or 3D display[3,4] but is not suitable to draw multi-layer graph. An effective graph layout algorithm needs to be developed to visualize the double-layer graph generated.

In this paper, we propose a new algorithm based on the traditional spring model to solve the multi-layer graph drawing problem. In our algorithm, we firstly use Force Directed spring modeling algorithm to locate genes within a cluster (small vertices) into local area - InfoCube. After allocating all genes

to InfoCubes, we treat all objects in an InfoCube as a single object (big vertex in a graph) and allocate all the InfoCubes into 3D space using the same layout algorithm. The bottom-up approach draws the double-layer graph efficiently because it avoids the space partition problem and saves time on computation. It is not only applicable to visualize double-layer gene clusters but also other hierarchical structured data.

2. GRAPH DRAWING

Graph drawing addresses the problem of visualizing structural information. A graph G (V, E) is a set V of vertices and a set E of edges, in which an edge joins a pair of vertices. An important task for a graph-based application is generating the layouts of the graphs. Force-Direct Placement is a well-known technique for drawing general undirected graphs[5]. Many models have been proposed for the Force-Directed Placement, such as spring model by Eades[6], magnetic spring model by Sugiyama[7], and others[8,9,10,11].

The spring modeling algorithm is a heuristic approach of graph drawing based on a physical system in which graph's edges are replaced by springs and replace the vertices (nodes) by rings. Figure 2 shows the spring model.

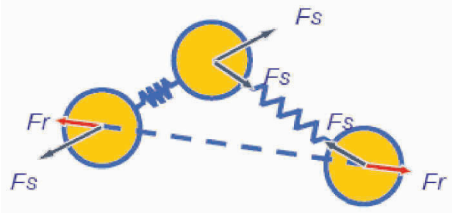

Figure 2. Spring Model.

The forces acting on every node include spring force F_s and repulsion force F_r. The resultant of force F_s and F_r can be calculated using Eq. (1) where d is the distance between a pair of nodes. The C_s, C_r and C_d are the constant parameters of the system. Among the parameters that control the forces acting on the nodes and causing their movements are spring length, spring stiffness, spring type and initial configuration. Under the influence of spring force between connected nodes and repulsion force between unconnected nodes, the graph will automatically calculate the position of each vertex until the system reaches a stable state[12].

$$F_s = C_s \log (d/C_d)$$
$$F_r = C_r / d^2 \quad (1)$$

3. MULTI-LAYER GRAPH LAYOUT – A NOVEL APPROACH

As we mentioned in section 1, the global view of gene clusters should be treated as a double-layer graph where nodes within a cluster form a small graph while clusters form a higher level graph. We propose a new algorithm based on the traditional spring model algorithm to layout two layers of graph simultaneously. In the algorithm, we first use spring modeling algorithm to locate genes within a cluster into a local area - InfoCube. After allocating all genes to InfoCubes, we treat all objects in an InfoCube as a single object and allocate all the InfoCubes using the same layout algorithm.

The algorithm takes a list of gene clusters as input and outputs the graphical layout of the gene clusters. Figure 3 illustrates a global view of the gene clusters generated. To make the visualization result clearer, a half-transparent cube is also added to highlight the boundary of the cluster and each cluster is assigned a randomly generated color. The outline of the new algorithm is as following:

Table 1. Auto layout algorithm.

Input	a List of gene clusters
Step 1	Layout genes within a cluster Randomly allocate each gene to a position within an InfoCube For *i*:=0 to iteration Begin calculate spring force of each node calculate repulsion force of each node update location of each node reduce the temperature as the layout approaches a better configuration End
Step 2	Layout InfoCubes Randomly allocate each InfoCube to a position to global 3D information space For *j*:=0 to iteration Begin calculate spring force of each InfoCube calculate repulsion force of each InfoCube update location of each InfoCube reduce the temperature as the layout approaches a better configuration End
Output	3D visualization of the gene clusters

The algorithm treats all genes within a cluster as a single object. This saves a lot of computations because when updating location of each

InfoCube after each round of iteration, there is no need to recalculate the coordinates of each gene. Another major achievement of this algorithm is that it avoids the space partition problem when we try to allocate gene clusters. This bottom-up approach is not only applicable in displaying clusters but also can be applied in other situations where more than one layer of details of a graph need to be visualized, for example, hierarchical structured graph.

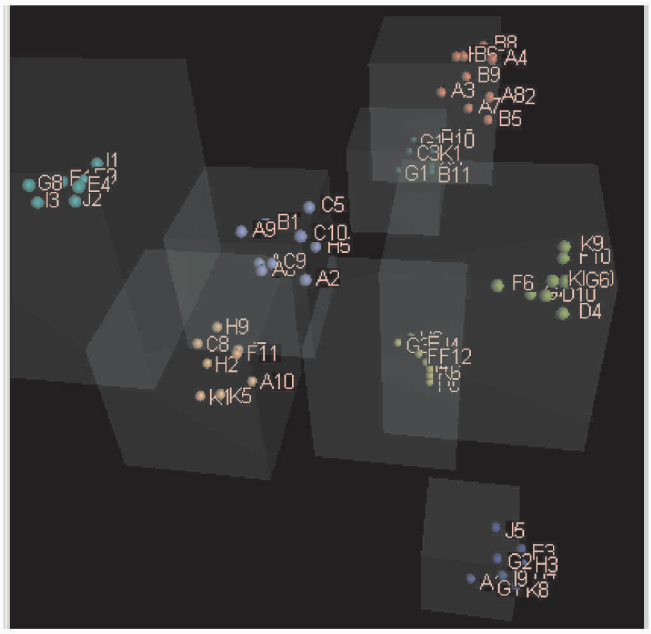

Figure 3. Global view of gene clusters.

4. CONCLUSION

Graph layout is extensively used in the field of mathematics and computer science. However those ideas and methods have not been fully extended in a general fashion to the construction of graphs for biological data. In this paper we propose a new layout algorithm to display the gene clusters when the similarity between genes and clusters is not indicated from the original data. The algorithm is efficient in terms of visualization speed because it solves both graph layout and space partition problem at the same time, typically taking 50 iterations to reach a stable graph layout. In object-oriented programming environment, the algorithm saves computation because when updating location of the InfoCubes after each round of

iteration, there is no need to recalculate the coordinates of individual objects inside the InfoCube.

Note that the final graph layout is not guaranteed to be optimal; here we do not explicitly seek an optimal solution but merely a stable solution that generates an "aesthetically pleasing" layout of a complex graph [9]. As we can see, the graphical representation of the gene clusters is simple, clear and easy to understand. Color and location are used to visualize genes that belong to different gene clusters. Within graphical user interface the user can play with the model by setting clustering methods and parameters, zooming in and out and inspecting gene clusters from different view angles. The algorithm can also be extended to display other multi-level graphs with hierarchical relationships.

REFERENCES

1. E.Segal R. Yelensky, A. Kaushal, T. Pham, A. Regev, D. Koller, and N. Friedman, "GeneXPress: A Visualization and Statistical Analysis Tool for Gene Expression and Sequence Data", Proceedings of the 11th Inter. Conf. on Intelligent Systems for Molecular Biology (ISMB), 2004
2. J. L. Arnott, A. Taylor, and N. Alm, "2-D and 3-D Information Visualization in the AAC User Interface", Proceeding of 10th international Conference on Augmentative and alternative Communication) ISAAC 2002), Odense, Denmark, 10th -15th August 2002.
3. A.J. Enright, C.A. Ouzounis, (2001) "BioLayout--an automatic graph layout algorithm for similarity visualization.", Bioinformatics, 17(9):853-854
4. A. Morrison, G. Ross, M. Chalmers, "Combining and comparing clustering and layout algorithms" 2002 EUROGRAPHICS –IEEE TCVG Symposium on Visualization
5. N. Quinn and M. Breur. A Force Directed Component Placement Procedure for Printed Circuit Boards. IEEE Transactions of Circuits and Systems, 26(6): 377-388, 1979.
6. P. Eades, "A heuristic for graph drawing", Congr. Number, 42:149-160, 1984.
7. K. Sugiyama and K. Misue. Graph Drawing by Magnetic-Spring Model, Res. Rep. ISIS-RR-94-14E, Inst. Social Information Science, Fujitsu Labs Ltd., 1994.
8. T. Fruchterman and E. Reingold., "Graph Drawing by Force-Directed Placement", Software – Practice and Experience 21, 1129-1164, 1991.
9. T. Kamada. Visualizing Abstract Objects and Relations, World Scientific, 1989.
10. T. Kamada and S. Kawai. An Algorithm for Drawing General Undirected Graph. Information letters, 31(1): 7-15, 1989.
11. A. Frick, A. Ludwig and H. Mehldau. A Fast Adaptive Layout Algorithm for Undirected Graphs. Proceedings of the Symposium on Graph Drawing GD'95, Springer-Verlag, 389-403, 1994.
12. Thomas M. J. Fruchterman, "Graph Drawing by Force-directed Placement", Software—Practice & Experience, VOL. 21(11), 1129-1164 (NOV. 1991)