LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Theresa Scharl & Friedrich Leisch

Visualizing Gene Clusters using Neighborhood Graphs in R

Technical Report Number 16, 2008
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Visualizing Gene Clusters using Neighborhood Graphs in R

Theresa Scharl[1] and Friedrich Leisch[2]

[1] Department of Statistics and Probability Theory, Vienna University of Technology
Wiedner Haupstraße 8-10/1071, 1040 Vienna, Austria,
theresa.scharl@ci.tuwien.ac.at

[2] Department of Statistics, University of Munich
Ludwigstraße 33, D-80539 München, Germany,
friedrich.leisch@stat.uni-muenchen.de

Abstract. The visualization of cluster solutions in gene expression data analysis gives practitioners an understanding of the cluster structure of their data and makes it easier to interpret the cluster results. Neighborhood graphs allow for visual assessment of relationships between adjacent clusters. The number of clusters in gene expression data is for biological reasons rather large. As a linear projection of the data into 2 dimensions does not scale well in the number of clusters there is a need for new visualization techniques using non–linear arrangement of the clusters. The new visualization tool is implemented in the open source statistical computing environment R. It is demonstrated on microarray data from yeast.

Keywords: Cluster analysis, graphs, microarray data, R

1 Introduction

Gene expression microarray experiments yield large and complex multivariate datasets that consist of several thousands of genes at multiple states. A typical question during the analysis is to find groups in the data. Cluster analysis is commonly used to reduce the complexity of the data from multidimensional space to a single nominal variable, the cluster membership. In the analysis of microarray data clustering is used as vector quantization and the set of genes is divided into artificial subsets. Genetic interactions are so complex that the definition of gene clusters is not clear. Additionally microarray data are very noisy and co–expressed genes can end up in different clusters. As no clear density clusters exist in the data the relationship between clusters is very important.

Clusters of co–expressed genes can help to discover potentially co–regulated genes or association to conditions under investigation. Usually cluster analysis provides a good initial investigation of microarray data before actually focusing on functional subgroups of interest. In the literature numerous cluster algorithms for clustering gene expression data have been proposed. Besides traditional methods like hierarchical clustering , K–means, partitioning

around medoids (PAM, K–medoids) or self–organizing maps there are several algorithms dealing with time–course gene expression data (e.g., Heyer et al., 1999, De Smet et al., 2002, Ben-Dor et al., 1999).

The display of cluster solutions particularly for a large number of clusters is very important in exploratory data analysis. Visualization methods give practitioners an understanding of the relationships between segments of a partition and make it easier to interpret the cluster results. Neighborhood graphs (Leisch, 2006) can be used for visual assessment of the cluster structure of centroid–based cluster solutions. A linear projection of the data into 2 dimensions using for example linear discriminant analysis (LDA) does not scale well in the number of clusters. Gene expression data are high–dimensional data and are usually separated into a lot of clusters (e.g., over 25 clusters in Heyer et al., 1999). As the vast amount of information cannot be shown in the plane there is a need for new visualization techniques. Using non–linear arrangement of the clusters the cluster structure can be displayed up to a very large number of clusters. In this work the layout algorithms implemented in the open source graph visualization software Graphviz are used for non–linear arrangement of the clusters. The new visualization tool is currently available at the homepage of the first author (http://www.ci.tuwien.ac.at/ scharl/Software/) and will be released as an R package (R Development Core Team, 2007, http://www.R-project.org) soon. The functionality is demonstrated on a publicly available data set from yeast.

2 Methods

2.1 Cluster Algorithms

In this work we focus on centroid–based cluster algorithms like K–means and PAM or others where clusters can be represented by centroids (e.g., QT–Clust, Heyer et al., 1999). For a given data set $X_N = \{x_1, \ldots, x_N\}$ the distance between points x and y is given by $d(x, y)$, e.g., the Euclidean or absolute distance. $C_K = \{c_1, \ldots, c_N\}$ is a set of centroids and the centroid closest to x is denoted by

$$c(x) = \underset{c \in C_K}{\operatorname{argmin}}\, d(x, c).$$

The set of all points where c_k is the closest centroid is given by

$$A_k = \{x_n | c(x_n) = c_k\}.$$

Minimizing the average distance between each data point and its closest centroid

$$D(X_n, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(x_n, c(x_n)) \rightarrow \min_{C_K}$$

is the task of most cluster algorithms.

2.2 Neighborhood Graphs

Neighborhood graphs (Leisch, 2006) use the idea of topology–representing networks (TRNs, Martinetz and Schulten, 1994) to count the number of data points a pair of centroids is closest and second–closest. In TRNs the counts are used as weights for the edges of the graph. Silhouette plots (Rousseeuw, 1987) are diagnostic plots revealing the goodness of a partition. The distance from each point to the points in its own cluster is compared to the distance to points in the second closest cluster. The larger the silhouette values the better a cluster is separated from the other clusters. But silhouette plots do not show the proximity of clusters. They only give an indicator how well-separated single points are from other clusters. Neighborhood graphs combine these two approaches and use the mean relative distances as edge weights in order to measure how separated pairs of clusters are. Hence they display the distance between clusters. In the graph each node corresponds to a cluster centroid and two nodes are connected by an edge if there exists at least one point that has these two as closest and second–closest centroid.

As described above the centroid closest to x is denoted by $c(x)$ and the second closest centroid to x is denoted by

$$\tilde{c}(x) = \underset{c \in C_K \setminus \{c(x)\}}{\operatorname{argmin}} \; d(x, c).$$

Now the set of all points where c_i is the closest centroid and c_j is second–closest is given by

$$A_{ij} = \{x_n | c(x_n) = c_i, \tilde{c}(x_n) = c_j\}.$$

For each observation x we define

$$s(x) = \frac{2d(x, c(x))}{d(x, c(x)) + d(x, \tilde{c}(x))}.$$

$s(x)$ is small if x is close to its cluster centroid and close to 1 if it is almost equidistant between the two cluster centroids. The average s–value of all points where cluster i is closest and cluster j is second closest can be used as a proximity measure between clusters and as edge weight in the graph.

$$s_{ij} = \begin{cases} |A_i|^{-1} \sum_{x \in |A_{ij}|} s(x), & A_{ij} \neq \emptyset \\ 0, & A_{ij} = \emptyset \end{cases}$$

$|A_i|$ is used in the denominator instead of $|A_{ij}|$ to make sure that a small set A_{ij} consisting only of badly clustered points with large shadow values does not induce large cluster similarity.

3 Data

In this work a publicly available dataset from yeast was investigated, the seventeen time point mitotic cell cycle data (Cho et al., 1998) available at

http://genome-www.stanford.edu. The dataset was preprocessed adapting the instructions given by Heyer et al. (1999). The outlier time points 10 and 11 were removed from the original 17 variables. The gene vectors were standardized to have median 0 and MAD 1. Finally genes that were either expressed at very low levels or did not vary significantly over the time points were removed. This procedure yields gene expression data on $N = 2832$ genes (observations) for $T = 15$ time points (variables). The data was clustered using the K–means algorithm. In this example 15 clusters were selected.

4 Software and Implementation

All cluster algorithms and visualization methods used are implemented in the statistical computing environment R. R package `flexclust` (Leisch, 2006) is a flexible toolbox to investigate the influence of distance measures and cluster algorithms. It contains extensible implementations of the K–centroids and QT–Clust algorithm and offers the possibility to try out a variety of distance or similarity measures as cluster algorithms are treated separately from distance measures. New distance measures and centroid computations can easily be incorporated into cluster procedures. The default plotting method for cluster solutions in `flexclust` is the neighborhood graph.

The visualization of partitioning cluster solutions is commonly accomplished by linear projection of the data into 2 dimensions using for example LDA. In Figure 1 the best possible separation using LDA is shown. The relationships between the centroids of 15 clusters can hardly be displayed in the plane. Therefore a new visualization tool is presented using non-linear arrangement of the nodes. Infrastructure for creating, manipulating, and visualizing graphs is provided in Bioconductor (Gentleman et al., 2005, http://www.bioconductor.org) package `graph`. The package contains functionality for data structure, classes and methods to manipulate graphs and enables efficient representations of very large graphs. An interface to the open source graph visualization software Graphviz (http://www.graphviz.org/) is provided in Bioconductor package `Rgraphviz` which returns the layout information for a graph object, x- and y–coordinates of the graph's nodes as well as the parameterization of the trajectories of the edges. Several layout algorithms can be chosen.

dot: hierarchical layout algorithm for directed graphs
neato and fdp: layout algorithms for large undirected graphs
twopi: radial layout
circo: circular layout

Additionally global and local properties (e.g., labels, shape, color, ...) can be assigned to both nodes and edges.

Using non–linear arrangement of clusters clearly improves the visualization of the cluster solution (Figure 2). In the new visualization method related

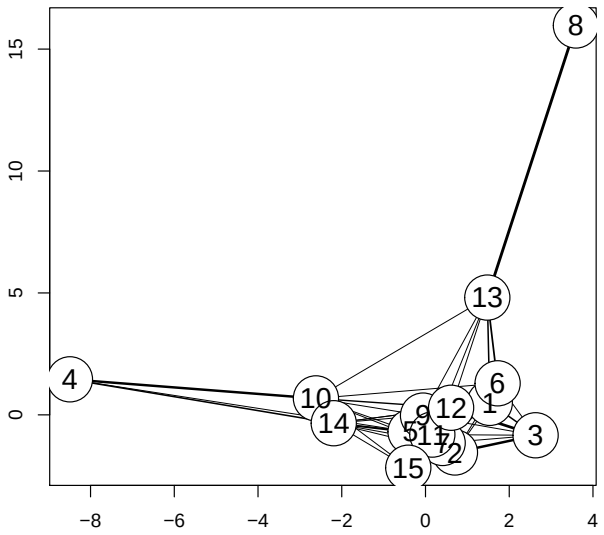

Fig. 1. Projection of neighborhood graph of a K–means cluster solution of the yeast data into 2 dimensions.

clusters are not forced to lie next to each other. For example cluster 11 located at the bottom end of the graph is related to cluster 1 located at the top end of the graph. Additionally the graph is simplified by only drawing edges between nodes if the similarity of a cluster to another cluster is at least 10%. In Figure 2 the cluster structure of the cluster solution can easily be investigated. Cluster 15 is very different from the remaining clusters as no edge is drawn to cluster 15 and the similarities to connected clusters are very small. This indicates that the genes in cluster 15 are very different from the remaining genes. As cluster 4 is similar to clusters 10 and 14 which are also strongly connected the 3 clusters seem to be highly related.

4.1 Edge Methods

The neighborhood graph is a directed graph as the similarity of cluster 1 to cluster 2 is different from the similarity of cluster 2 to cluster 1. Besides plotting the original directed graph there are several possibilities how to plot edges taking into account for instance the mean, minimum or maximum of the similarities between two clusters.

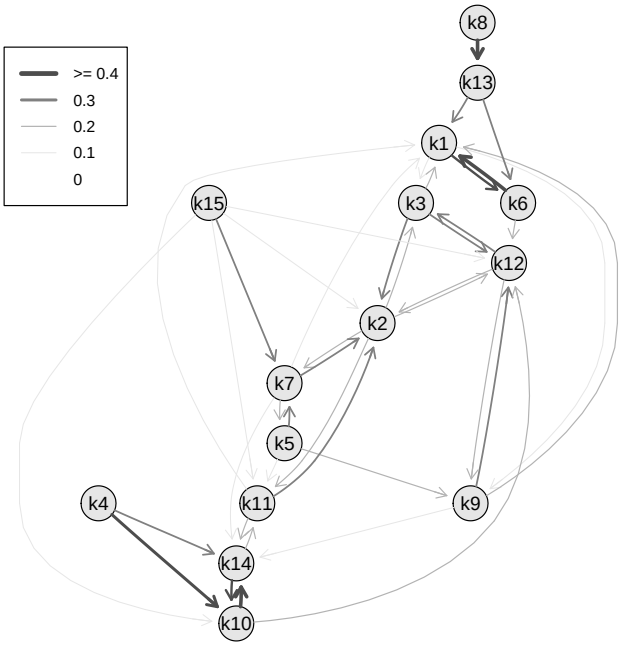

Fig. 2. Neighborhood graph of a K–means cluster solution for the yeast data using the *dot* layout algorithm.

4.2 Node Methods

In the simple visualization of a neighborhood graph one single kind of node symbol is used for all nodes. By just looking at the graph no information about the different clusters is revealed. There are several possibilities how to include additional information in the representation of nodes. The most simple method is to use color coding, e.g., to color nodes by size or tightness of the corresponding clusters. Another possibility is to use different shapes or symbols for nodes representing clusters with specific properties.

In Figure 3 the cluster solution is shown with tight clusters highlighted, i.e., clusters with small average distance of the genes to the cluster centroid. In this example the tightest clusters are numbers 15 and 8 indicating genes very different from the rest of the genes (colored darkgrey) and numbers 13 and 4 (lightgrey).

4.3 Other Features

The new visualization tool offers various possibilities for the analysis of microarray data which cannot be shown here due to space constrains. The neighborhood graph is implemented in an interactive way and gene clusters can be

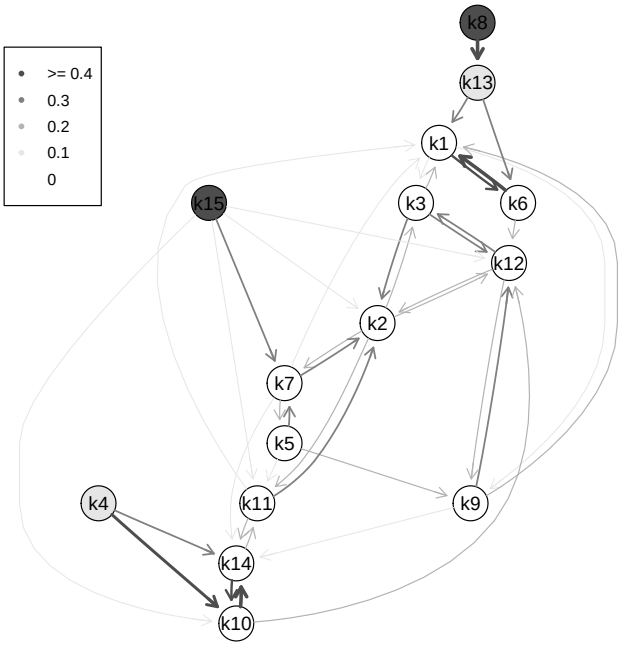

Fig. 3. The neighborhood graph with tight clusters highlighted.

investigated by clicking on the nodes. Plots of the expression profiles of the corresponding genes pop up as well as tables giving further information about the genes. Additional information about the gene clusters can be included in the neighborhood graph. External information from differential expression analysis or functional grouping can easily be included in the node representation, e.g., the accumulation of gene ontology (GO) categories in certain gene clusters.

5 Summary

Cluster analysis is commonly used in the analysis of gene expression data to find groups of co–expressed genes. But the definition of gene clusters is not very clear as genetic interactions are extremely complex. For this reason the relationship between clusters is very important as co–expressed genes can end up in different clusters. The neighborhood graph is a useful tool to visualize the underlying cluster structure. The visualization of partitioning cluster solutions is commonly accomplished by linear projection of the data into 2 dimensions. However, this is not recommended for high–dimensional data like microarray data and a large number of clusters. In our new visual-

ization method layout algorithms for non–linear arrangement of the clusters are used to display the relationships between clusters. Our interactive software tools for the analysis of gene expression data is very helpful not only for statisticians but also for practitioners.

Acknowledgement

This work was supported by the Austrian K_{ind}/K_{net} Center of Biopharmaceutical Technology (ACBT).

References

BEN-DOR, A., SHAMIR, R. and YAKHINI, Z. (1999): Clustering gene expression patterns. *Journal of Computational Biology, 6 (3–4), 281–297.*

BICKEL, D.R. (2003): Robust cluster analysis of microarray gene expression data with the number of clusters determined biologically. *Bioinformatics, 19 (7), 818–824.*

CAREY, V.J., GENTLEMAN, R., HUBER, W. and GENTRY, J. (2005): Bioconductor Software for Graphs. In: R. Gentleman, V.J. Carey, W. Huber, R.A. Irizarry and S. Dudoit (Eds.): *Bioinformatics and Computational Biology Solutions Using R and Bioconductor.* Springer, New York.

CHO, R.J., CAMPBELLl, M.J., WINZELER E.A., STEINMETZ, L., CONWAY, A., WODICKA, L., WOLFSBERG, T.G., GABRIELIAN, A.E., LANDSMAN, D., LOCKHART, D.J. and DAVIS, R.W. (1998): A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell, 2/1, 65-73.*

DE SMET, F., MATHYS, J., MARCHAL, K., THIJS, G., DE MOOR, B. and MOREAU, Y. (2002): Adaptive quality-based clustering of gene expression profiles. *Bioinformatics, 18 (5), 735–746.*

GENTLEMAN, R., CAREY, V.J., HUBER, W., IRIZARRY, R.A. and DUDOIT, S. (Eds.) (2005): *Bioinformatics and Computational Biology Solutions Using R and Bioconductor.* Springer, New York.

HEYER, L.J., KRUGLYAK, S. and YOOSEPH, S. (1999): Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research, 9, 1106-1115.*

LEISCH, F. (2006): A Toolbox for K-Centroids Cluster Analysis. *Computational Statistics and Data Analysis 51 (2), 526–544.*

MARTINETZ, T. and SCHULTEN, K. (1994): Topology representing networks. *Neural Networks, 7 (3), 507–522.*

R DEVELOPMENT CORE TEAM (2007): *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org.

ROUSSEEUW, P.J. (1987): Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20, 53–65.*