Publication 6

Jarkko Venna, and Samuel Kaski. Visualizing gene interaction graphs with local multidimensional scaling. In Michel Verleysen, editor, *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN'2006)*, Bruges, Belgium, April 26–28, pp. 557–562, d-side, Evere, Belgium, 2006.

6

ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

Visualizing Gene Interaction Graphs with Local Multidimensional Scaling

Jarkko Venna and Samuel Kaski *

Helsinki University of Technology - Adaptive Informatics Research Centre
P.O. Box 5400, FI-02015 TKK - Finland

Abstract. Several bioinformatics data sets are naturally represented as graphs, for instance gene regulation, metabolic pathways, and protein-protein interactions. The graphs are often large and complex, and their straightforward visualizations are incomprehensible. We have recently developed a new method called *local multidimensional scaling* for visualizing high-dimensional data sets. In this paper we adapt it to visualize graphs, and compare it with two commonly used graph visualization packages in visualizing yeast gene interaction graphs. The new method outperforms the alternatives in two crucial respects: It produces graph layouts that are both more trustworthy and have fever edge crossings.

1 Introduction

It is obvious that the various cellular networks are crucial in studying gene function and more generally in systems biology. Such networks are naturally represented as graphs, where nodes are the key elements (genes or proteins) and the interaction between two elements is represented by an edge connecting the two nodes. Often the edge is given a weight or length that represents the interaction strength. When the size of the interaction network increases it becomes practically impossible to draw it manually and sophisticated automatic methods are needed for visualization [1, 2, 3].

In this paper we introduce a new graph drawing algorithm by adapting our earlier local multidimensional scaling method, and evaluate it and two alternatives in the task of visualizing gene interaction networks.

2 Graph visualization methods

There are two common principles for designing graph layout algorithms for general nondirected graphs: the so-called spring model, and a cost function that aims at preserving graph distances. The two comparison methods are representatives of the two principles.

2.1 Graphviz

Graphviz [4] is a software package that implements two graph layout methods. We focus here on the method called Neato, designed for undirected graphs. The

*This work was supported by the Academy of Finland, decision numbers 79017 and 207467 and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-202-506778.

cost function of Neato which, is almost the same as Sammon's mapping, is

$$E = \sum_{i<j} \frac{(d(\mathbf{x}_i, \mathbf{x}_j) - d_{ij})^2}{d_{ij}^2},$$

where the $\mathbf{x}_i$ is the locations of node i on the visualization, $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between the nodes i and j in the layout, and d_ij is the shortest path graph distance between nodes i and j. Graphviz uses a majorization algorithm to optimize this cost function.

Graphviz is available from http://www.graphviz.org/.

2.2 LGL

The LGL graph layout algorithm [3] is based on a spring model where attractive and repulsive forces affect the graph nodes. Each edge induces an attractive force between connected nodes and nodes too near each other are affected by a repulsive force. Each node is given a starting position and then the model is iterated until the forces reach equilibrium.

The LGL algorithm starts by first selecting a root node and calculating the minimum spanning tree (MST) of the graph. The layout begins with the root node and iteratively adds nodes based on their distance from the root in the MST. A new equilibrium is calculated after adding each new set of nodes.

LGL is available at http://bioinformatics.icmb.utexas.edu/lgl/.

2.3 Local Multidimensional Scaling (MDS)

We have recently [5] introduced a new visualization method for nonlinear projection of data sets. It minimizes a cost function which is a tunable compromise between two types of errors: errors in preserving distances for nodes that are neighbors on the *layout* (trustworthiness of the visualization), and for points that are proximate in the original *graph* ("continuity" of the projection). The tradeoff is tunable by a parameter λ. The cost function of local MDS is

$$\begin{aligned} E &= \frac{1}{2}\sum_i \sum_{j \neq i} (d(\mathbf{x}_i, \mathbf{x}_j) - d_{ij})^2 [(1-\lambda) F(d(\mathbf{x}_i, \mathbf{x}_j), \sigma_i) + \lambda F(d_{ij}, \sigma_i)] \,, \\ F(d, \sigma)) &= \begin{cases} 1 & \text{if } d \leq \sigma \\ 0 & \text{if } d > \sigma \,. \end{cases} \end{aligned}$$

Here $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between the nodes i and j in the visualization and d_{ij} is the shortest path distance on the graph between nodes i and j. We optimize the cost function with the stochastic gradient descent introduced for Curvilinear Component Analysis [6]. During the optimization the radius of the area of influence around each data point i, σ_i, is slowly brought down. The final radius is set to equal the distance of the K:th nearest neighbor of the data point i in the original space.

ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

3 Evaluating the goodness of a graph visualization

There have been relatively few studies on what makes a graph easy to analyze. One feature that has been found to have a strong degrading effect is the number of times edges cross in the graph [7, 8]. We will use this measure to evaluate the graphs produced by the three methods, and additionally measure how well the proximity structure of the graph is preserved in the node layout process, using the measures of trustworthiness and continuity defined below.

3.1 Trustworthiness and continuity of a weighted graph layout

We consider a graph layout of a weighted graph to be *trustworthy* if the set of k closest neighbors of a node on the display are also close-by on the graph.

Let N be the number of nodes and $r(i,j)$ be the rank of the node j in the ordering according to the shortest path distance from node i on the graph. Denote by $U_k(i)$ the set of those nodes that are in the neighborhood of size k of the node i in the layout but not in the graph. Our measure of trustworthiness of the visualization is

$$M_1(k) = 1 - \frac{2}{Nk(2N-3k-1)} \sum_{i=1}^{N} \sum_{j \in U_k(i)} (r(i,j) - k) \; . \tag{1}$$

For more details see [9, 10], where the measures have been used for evaluating quality of dimensionality reduction.

The errors caused by discontinuities may be quantified analogously to the errors in trustworthiness. Let $V_k(i)$ be the set of those data samples that are in the neighborhood of the node i in the graph but not in the layout, and let $\hat{r}(i,j)$ be the rank of the node j in the ordering according to the Euclidean distance from i in the layout. The measure of continuity, $M_2(k)$, is defined analogously to Eq. (1). The set $V_k(i)$ replaces the set $U_k(i)$ and $\hat{r}(i,j)$ replaces $r(i,j)$ in the equation.

3.2 Trustworthiness and continuity of an unweighted graph layout

For unweighted graphs the natural neighborhood to preserve for a node is the set of nodes connected to it. Hence, instead of selecting the k closest neighbors based on distance which would be equal for many nodes, we choose all directly connected nodes as the neighborhood. The trustworthiness, defined analogously to (1), is

$$M_1^u = 1 - \frac{2}{N} \sum_{i=1}^{N} \sum_{j \in U_k(i)} \frac{(r(i,j) - m_i)}{m_i(2N - 3m_i - 1)} \; ,$$

where m_i is the number of edges connecting to the node i. The rank $r(i,j)$ of all points that are at the same distance from i in the graph is the rank of the first one. This measure could also be used on weighted graphs if we consider direct connections to be more important than short distances.

The measure of continuity M_2^u of an unweighted graph layout is defined similarly.

4 Experiments

4.1 Data

We tested the graph visualization algorithms on two data sets.

Harbison. The Harbison data set [11] on yeast (*Saccharomyces*) contains p-values for the bindings of several transcriptional regulators (some in several conditions) to 6229 genes. We first dropped out genes that could not be matched to the transcriptional regulators. Then we combined the p-values that were collected in different treatments by only keeping the smallest (most significant) one. The graph was formed by connecting genes that had a p-value ≤ 0.001. Both weighted and unweighted version of the largest connected component graph (147 nodes) were used in the experiments. Because the scale of the p-values varies over several magnitudes we transformed the p-values by first taking the negative logarithm and then subtracting the result from the maximum value. A small value (0.1) was added to make the shortest distance nonzero.

Lee. The Lee data set [12] contains the regulator–regulator interaction network of 106 yeast (*Saccharomyces cerevisiae*) genes. The data only indicates whether interaction was present or not. This leads to an unweighed graph. The data is available from
http://jura.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f= .

4.2 Results

All algorithms were run four times to avoid local minima. On local MDS different values (5, 10, 15 and 20) for the K parameter were tested. On Graphviz and LGL the run producing the best trustworthiness is reported. The best run on local MDS was selected based on the λ-weighted average of trustworthiness and continuity for each value of λ.

Table 1 summaries the results. Local MDS with $\lambda = 0.1$ or $\lambda = 0.2$ produced the best results on all three graphs: these layouts had the highest trustworthiness values together with the smallest numbers of edge crossings. Also continuity of the local MDS layouts was equal to or better than on the other layouts. On the two unweighted graphs Graphviz outperformed LGL slightly. On the weighted graph it performed poorly. It is possible that this was caused by problems in the implementation and not by the algorithm, however.

A visualization of the gene interaction network (based on the Lee data) with each of the three methods is presented in Figure 1. It is difficult to judge the relative quality of the visualizations without a detailed study, but it is clear that local MDS has produced a layout with fewer edge crossings. This is visible especially in the central part of the graph.

Harbison data, weighted graph, 147 nodes 802 edges

Method	M_1/M_2	crossings (best/average/worst)
Graphviz	0.57 / 0.72	11536 (9533/12749/17128)
LGL	0.64 / 0.80	5262 (4504/4662/4765)
lMDS $\lambda = 0.1$	**0.86 / 0.85**	**4068** (3884/4139/4527)
lMDS $\lambda = 0.2$	**0.86 / 0.85**	4399 (3891/4154/4399)

Harbison data, unweighted graph, 147 nodes 802 edges

Method	M_1^u/M_2^u	crossings (best/average/worst)
Graphviz	0.81 / **0.87**	4073 (3964/4182/4432)
LGL	0.81 / **0.87**	4455 (4212/4574/5262)
lMDS $\lambda = 0.1$	**0.97** / 0.86	**3625** (3619/3871/4220)
lMDS $\lambda = 0.2$	0.96 / **0.87**	3825 (3635/3965/4441)

Lee data, unweighted graph, 106 nodes 182 edges

Method	M_1^u/M_2^u	crossings (best/average/worst)
Graphviz	0.93 / **0.96**	68 (58/63/68)
LGL	0.92 / 0.95	71 (71/79/94)
lMDS $\lambda = 0.1$	**0.99** / 0.95	45 (40/50/68)
lMDS $\lambda = 0.2$	**0.99 / 0.96**	**33** (33/50/79)

Table 1: Trustworthiness (M_1), continuity of the mapping (M_2) and number of edge crossings produced by different methods. On the weighted graph trustworthiness and continuity are reported for a neighborhood size $k = 6$ which equals the average number of edges connected to a node. On the unweighted graphs the actual node-specific neighborhood size is used. In addition to the number of crossings on the selected layout, the smallest (best), average, and the highest (worst) number produced on the different runs of each method is given.

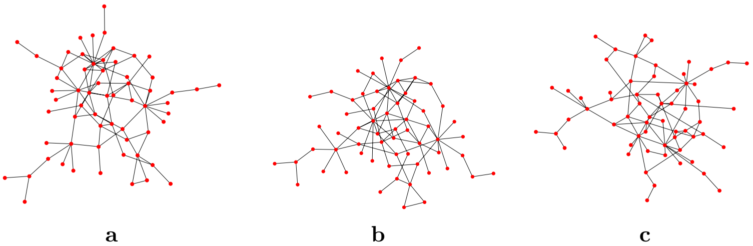

Fig. 1: Examples of gene interaction graph layouts for the Lee data: **a**) Graphviz **b**) LGL **c**) local MDS ($\lambda = 0.2$). Each node in the graph is a gene and an edge indicates interaction between genes.

5 Discussion

We introduced a new graph layout algorithm, and evaluated it and two earlier algorithms in the task of visualizing gene interaction networks. Two of the methods, Graphviz and LGL, have been previously used in bioinformtatics for visualizing network data and the third is our new method, local Multidimensional Scaling. We also extended a pair of measures previously used to evaluate the trustworthiness and continuity of different visualizations to measure the quality of unweighted graph layouts.

It turns out that local MDS produced graph layouts that were both more trustworthy and had the least number of edge crossings, which makes them easier to analyze.

References

[1] A.J. Enright and C.A. Ouzounis. Biolayout—an automatic graph layout algorithm for similarity visualization. *Bioinformatics*, 17:853–854, 2001.

[2] A. Goesmann, M. Haubrock, F. Meyer, J. Kalinowski, and R. Giegerich. Pathfinder: reconstruction and dynamic visualization of metabolic pathways. *Bioinformatics*, 18:124–129, 2002.

[3] Alex T. Adai, Shailesh V. Date, Shannon Wieland, and Edward M. Marcotte. LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks. *Journal of Molecular Biology*, 340:179–190, 2004.

[4] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, 30:1203–1233, 2000.

[5] Jarkko Venna and Samuel Kaski. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *Proceedings of 5th Workshop on Self-Organizing Maps*, pages 695–702, Paris, France, 2005.

[6] Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154, 1997.

[7] H. C. Purchase, R. F. Cohen, and M. I. James. An experimental study of the basis for graph drawing algorithms. *ACM Journal of Experimental Algorithmics*, 2, 1997.

[8] Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1:103–110, 2002.

[9] Samuel Kaski, Janne Nikkilä, Merja Oja, Jarkko Venna, Petri Törönen, and Eero Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4:48, 2003.

[10] Jarkko Venna and Samuel Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of ICANN 2001, International Conference on Artificial Neural Networks*, pages 485–491, Berlin, 2001. Springer.

[11] C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. Macisaac, T.W. Danford, N.M. Hannett, J.B. Tagne, D.B. Reynolds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, and R.A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99–104, 2004.

[12] Tony I. Lee, Nicola J. Rinaldi, Francois Robert, Duncan T. Odom, Ziv Bar-Joseph, Georg K. Gerber, Nancy M. Hannett, Christopher R. Harbison, Craig M. Thompson, Itamar Simon, Julia Zeitlinger, Ezra G. Jennings, Heather L. Murray, D. Benjamin Gordon, Bing Ren, John J. Wyrick, Jean-Bosco Tagne, Tom L. Volkert, Ernest Fraenkel, David K. Gifford, and Rick A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298:799–804, 2002.