# Network Brownian Motion: A New Method to Measure Vertex-Vertex Proximity and to Identify Communities and Subcommunities

Haijun Zhou and Reinhard Lipowsky

Max-Planck-Institute of Colloids and Interfaces, D-14424 Potsdam, Germany
{zhou,lipowsky}@mpikg-golm.mpg.de

**Abstract.** The networks considered here consist of sets of interconnected vertices, examples of which include social networks, technological networks, and biological networks. Two important issues are to measure the extent of proximity between vertices and to identify the community structure of a network. In this paper, the proximity index between two nearest-neighboring vertices of a network is measured by a biased Brownian particle which moves on the network. This proximity index integrates both the local and the global structural information of a given network, and it is used by an agglomerative hierarchical algorithm to identify the community structure of the network. This method is applied to several artificial or real-world networks and satisfying results are attained. Finding the proximity indices for all nearest-neighboring vertex pairs needs a computational time that scales as $O(N^3)$, with $N$ being the total number of vertices in the network.

## 1 Introduction

Network models are necessary to understand the behavior of complex systems, such as a biological organism, an ecological system, a human society, or the Internet. A Network is composed of a set of vertices and a set of edges which connect these vertices. Many complex networks were constructed and studied in recent years, and the *small-world* and *scale-free* properties of real-world networks were discovered (for a review, see [1–3]).

As far as the dynamics of networks is considered, the concept of network Brownian motion (or random walks) has aroused some interest among statistical physicists [4–10]. For example, the diffusion constant on a small-world network was investigated in [4]; random walks were used in [5, 6] to study network search problems and in [7, 8] to study network traffic and congestion. In [9, 10] a new approach based on Brownian motion was introduced, by which one can measure the extent of proximity between neighboring vertices of a given network and cluster vertices of this network into different communities and subcommunities.

The present work extends the basic idea of [9, 10]. Intuitively, a community of a network should consist of a subset of vertices which are more "near" to each other than to vertices not included in this subset. We give a quantitative definition of proximity measure and show how to calculate its value efficiently. This proximity measure is based on biased Brownian motion on a network. We apply this proximity measure in identifying the community structure of several networks.

Section 2 introduces a class of biased network Brownian motions and define a vertex-vertex proximity index. Section 3 outlines the clustering algorithm *Netwalk* and shows its performance by application on random modular networks. Section 4 applies the *Netwalk* algorithm to several social and biological networks. We conclude this work in section 5 together with some discussion.

## 2 Biased Brownian motion and proximity index

Consider a connected network of $N$ vertices and $M$ edges, with a weight matrix $\omega$. If there is no edge between vertex $i$ and vertex $j$, $\omega_{ij} = 0$; if there is an edge in between, $\omega_{ij} \equiv \omega_{ji} > 0$ and its value corresponds to the interaction strength of this edge. In a social friendship network, for example, $\omega_{ij}$ may be proportional to the frequency of contact between person $i$ and person $j$. A Brownian particle moves on the network, and at each step it jumps from its present position, say $i$, to a nearest-neighboring position, say $j$. We assume that the jump probability $P_{ij}$ has the form

$$P_{ij} = \frac{1}{K_i}\omega_{ij}(c_{ij}+1)^{\gamma}, \tag{1}$$

where $K_i = \sum_k \omega_{ik}(c_{ik}+1)^{\gamma}$, and $c_{ij}$ is the number of common nearest-neighbors of vertex $i$ and vertex $j$. For $\gamma > 0$, the Brownian particle has greater probability at each vertex to jump to a nearest-neighboring vertex that shares more nearest-neighbors with the original vertex. Equation (1) thus defines a biased Brownian motion, with the degree of bias being controlled by the bias exponent $\gamma$. For $\gamma = 0$, eq. (1) reduces to the unbiased Brownian motion discussed in refs. [9, 10].

For convenience, we introduce a generalized adjacency matrix $\mathbb{A}$ with matrix elements $\mathbb{A}_{ij} = \mathbb{A}_{ji} = \omega_{ij}(c_{ij}+1)^{\gamma}$. In addition, we define a diagonal matrix $\mathbb{K}$ such that $\mathbb{K}_{ii} = K_i$. The transfer matrix $P$ in eq. (1) is then given by $P = \mathbb{K}^{-1}\mathbb{A}$.

Suppose the Brownian particle is initially located at vertex $i$. The mean-first-passage-time $d_{ij}$ [9] is the average number of steps the Brownian particle takes before it reaches vertex $j$ (or return to $i$ in the case of $i = j$) for the first time. This quantity is given by

$$d_{ij} = P_{ij} + \sum_{m=1}^{+\infty}(m+1)\sum_{k_1\neq j;\ldots;k_m\neq j} P_{ik_1}P_{k_1k_2}\ldots P_{k_mj}. \tag{2}$$

It has been shown in ref. [9] that $d_{ij}$ is the solution of the linear equation

$$[I - B(j)]\begin{pmatrix} d_{1j} \\ \vdots \\ d_{Nj} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{3}$$

where $B(j)$ is the matrix formed by replacing the $j$-th column of matrix $P$ with a column of zeros. Equation (3) seems to imply that $N$ matrix inversion operations are needed to calculate the values of $d_{ij}$ for all pairs $i$, $j$. This would lead to a computational time of $O(N^4)$. However, what we really need to know is the difference of mean-first-passage-times, $\Delta(i,j;k) = d_{ik} - d_{jk}$. In this article, we describe a method by which one can calculate all the $N^2(N-1)/2$ different $\Delta(i,j;k)$ values with a computational time of $O(N^3)$.

Equation (3) is equivalent to

$$[\mathbb{K} - \mathbb{A}] \begin{pmatrix} d_{1j} \\ d_{2j} \\ \vdots \\ d_{Nj} \end{pmatrix} = \begin{pmatrix} K_1 - \mathbb{A}_{1j} d_{jj} \\ K_2 - \mathbb{A}_{2j} d_{jj} \\ \vdots \\ K_N - \mathbb{A}_{Nj} d_{jj} \end{pmatrix}. \tag{4}$$

Because the determinant of $\mathbb{K} - \mathbb{A}$ is zero, one cannot invert this matrix directly to find the solution of eq. (4). However, one can construct two $(N-1) \times (N-1)$ matrices $\mathbb{K}_r$ and $\mathbb{A}_r$ by removing the last rows and columns[1] of the matrices $\mathbb{K}$ and $\mathbb{A}$ [11]. This leads to

$$d_{ij} = d_{Nj} + \sum_{l=1}^{N-1} \left( \frac{1}{\mathbb{K}_r - \mathbb{A}_r} \right)_{il} K_l - \frac{\mathrm{Tr}\mathbb{K}}{K_j} \sum_{l=1}^{N-1} \left( \frac{1}{\mathbb{K}_r - \mathbb{A}_r} \right)_{il} \mathbb{A}_{lj}, \qquad (i < N). \tag{5}$$

In deriving eq. (5) we have used the fact that $d_{jj}$, the average returning time, is independent of network topology, with $d_{jj} = \sum_{l=1}^{N} K_l / K_j = \mathrm{Tr}\mathbb{K}/K_j$ [12].

With eq. (5) one only needs to invert the matrix $\mathbb{K}_r - \mathbb{A}_r$ to obtain the values of all quantities $\Delta(i,j;k)$. The total computation time scales as $O(N^3)$.

For each nearest-neighboring pair of vertices $i$ and $j$ with $\omega_{ij} > 0$, we define the proximity index

$$\Lambda(i,j) = \frac{\sqrt{\sum_{k \neq i,j} \Delta^2(i,j;k)}}{(N-2)} \tag{6}$$

in order to quantity the extent of proximity between $i$ and $j$. If two nearest-neighboring vertices $i$ and $j$ belong to the same community, then the mean-first-passage-time $d_{ik}$ from $i$ to any another vertex $k$ $(k \neq i,j)$ will be approximately equal to that from $j$ to $k$; in other words, the "coordinates" of the two vertices will be near to each other. Consequently, $\Lambda(i,j)$ will be small if $i$ and $j$ belong to the same community and large if they belong to different communities. The proximity index eq. (6) gives a quantitative measure of vertex-vertex proximity for a network that has no metric otherwise.

This proximity index is used in the *Netwalk* algorithm of the following section.

[1] Actually one can remove an arbitrary row and an arbitrary column and the result is unchanged. Here for definiteness, we remove the $N$-th row and the $N$-th column.

## 3 *Netwalk* algorithm

We exploit the proximity index to reveal the community structure of a network. The *Netwalk* algorithm works as follows:

1. Calculate the inverse of $\mathbb{K}_r - \mathbb{A}_r$.
2. Calculate the proximity index $\Lambda(i,j)$ for all nearest-neighboring pairs based on eq. (5) and eq. (6).
3. Initially, the network has $N$ communities, each contains a single vertex. We define the proximity index between two communities $\alpha$ and $\beta$ as

$$\Lambda_{\alpha,\beta} = \frac{1}{n_{\alpha,\beta}} \sum_{(i,j):\omega_{ij}>0, i\in\alpha, j\in\beta} \Lambda(i,j), \tag{7}$$

   where the summation is over all edges $(i,j)$ that connect communities $\alpha$ and $\beta$, and $n_{\alpha,\beta}$ is the total number of such edges. Merge the two communities with the lowest proximity index into a single community and then update the proximity index between this new community and all the other remaining communities that are connected to it. This merging process is continued until all the vertices are merged into a single community corresponding to the whole network.
4. Report the community structure and draw a dendrogram.

We tested the *Netwalk* algorithm on an ensemble of modular random networks. Each network in this ensemble has 128 vertices, 1024 edges and, hence, an average degree of 16 for each vertex. The vertices are divided into four communities or modules of size 32 each. The connection pattern of each network is random, except that each edge has a probability $p$ to be between two different modules. For each value of $p$, 100 random networks have been generated and studied. The results are listed in Table 1. The performance of *Netwalk* is remarkable. For example, when $p = 0.4$, i.e., when each vertex has on average 6.4 between-community edges and 9.6 within-community edges, only about eight vertices (6% of all vertices) are misclassified by this algorithm using linearly-biased Brownian motion.

Table 1 also suggests that, the performance of the linearly-biased Brownian motion ($\gamma = 1$ in eq. (1)) is considerably superior to those of $\gamma = 0$ and $\gamma = 2$. Indeed, we have observed that, for each generated random network, in most cases the number of misclassified vertices by using $\gamma = 1$ is less than that reported by using $\gamma = 0$ or $\gamma = 2$. Therefore, in our later applications, linearly-biased Brownian motion ($\gamma = 1$) will be used[2].

## 4 Applications

We apply the *Netwalk* algorithm to several real-world networks in order to detect their community structures. In the following, we first discuss the results for two social networks and then for two biological networks.

[2] In general, one may also use non-integer values of $\gamma$ but this has not been explored so far.

**Table 1.** Number of misclassified vertices as a function of the between-community probability $p$. For each value of $p$, 100 random networks with 128 vertices are generated. The results obtained by using unbiased ($\gamma = 0$), linearly-biased ($\gamma = 1$) and squarely-biased ($\gamma = 2$) Brownian motions are compared.

| $p$ | $\gamma = 0$ | $\gamma = 1$ | $\gamma = 2$ |
|---|---|---|---|
| 0.3 | $0.43 \pm 0.78$ | $0.43 \pm 0.70$ | $0.62 \pm 0.90$ |
| 0.35 | $2.9 \pm 2.9$ | $1.96 \pm 2.1$ | $2.59 \pm 2.4$ |
| 0.4 | $13.0 \pm 7.5$ | $8.3 \pm 5.4$ | $10.2 \pm 6.5$ |
| 0.45 | $38.1 \pm 14.6$ | $26.5 \pm 10.8$ | $29.8 \pm 10.6$ |

**Karate club network of Zachary (1977)** The fission process of a social network was studied by Zachary in 1977 [13]. This network is composed of 34 members of a karate club, it has 77 weighted edges. It broke up into two parts because of a disagreement between the club's officer and its instructor. Figure 1 shows the community structure of the network just before the fission event. The two main communities identified by our algorithm are in full agreement with the actual fission pattern [13].

**Football network of Girvan and Newman (2002)** The American football network collected by Girvan and Newman [14] contains 115 vertices (football teams) and 613 unweighted edges (matches). The community structure of this network is shown in fig. 2. Comparing the predicted pattern with the actual conference structure of these football teams, we see the following differences: (1) Conference 9 is divided into two parts by our algorithm. We have checked that there is no direct connection between the two parts of conference 9. (2) Vertex 111 is grouped into conference 11. We have checked that it has eight edges to conference 11 and only three edges to other conferences; similarly, we have checked that vertex 59 has stronger interaction with conference 9 than with any other conference. (3) Vertices in conference 12 are distributed into several conferences. We have also checked that there are very few direct interactions between the five members of this conference.

**Protein-protein interaction network of yeast *Saccharomyces cerevisiae*** The yeast protein-protein interaction network is constructed according to experimental data [15, 16]. Each vertex of this network represents a protein, and each edge represents some physical interaction between the two involved proteins. The reduced form of this network contains 871 vertices and 2043 unweighted edges [10].

The community structure of this network as obtained via *Netwalk* is shown in fig. 3, which is strikingly different from those of the two social networks as described above. On the global scale, the protein-protein interaction network cannot be divided into two or more large communities of similar size. At each proximity level, the network has one giant community and many small commu-

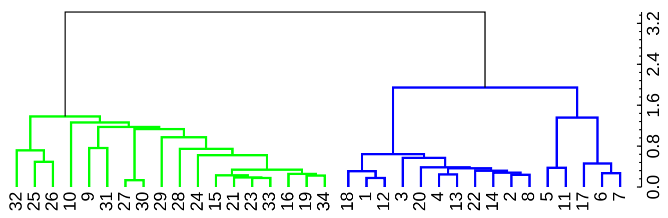

**Fig. 1.** The community structure of a karate club network obtained by *Netwalk* ($\gamma = 1$). The ruler specifies the proximity indices between (sub)communities. This figure and following figures are generated by the **den** program of P. Kleiweg (`http://odur.let.rug.nl/~kleiweg/levenshtein/`).

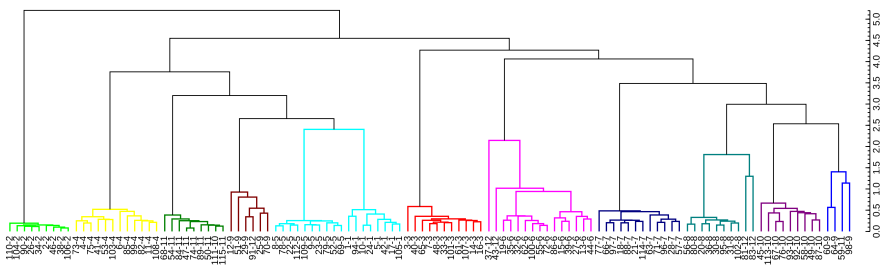

**Fig. 2.** The community structure of a football-team network obtained by *Netwalk*. In the name pattern *xx-yy*, the number *yy* after the hyphen denotes the group identity of vertex *xx* according to information from other sources.

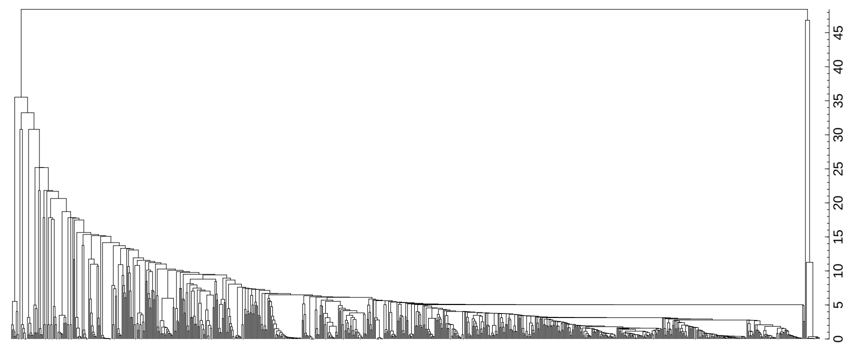

**Fig. 3.** Community structure of yeast's protein-protein interaction network predicted by *Netwalk*.

nities, each of these small communities containing of the order of ten proteins. As the community-community proximity index is increased, these small communities are integrated into the giant community in a hierarchical order. This hierarchical organization of small modules seems to be a universal feature of biological networks. It is unlikely to be caused by a particular clustering method. Similar hierarchical patterns are observed in metabolic networks using different clustering methods [17, 18] and in the gene-regulation network which will be discussed as the next example, see fig. 5. The construction principles underlying such hierarchical structures are still to be fully appreciated. It is plausible that such hierarchical structures contain information about the evolutionary history of the biological organisms (H. W. Peng and L. Yu, private communication).

Based on fig. 3 many communities of proteins can be obtained. Figure 4 gives some examples. The communities and subcommunities predicted by the *Netwalk* algorithm are composed of proteins that have similar cellular locations and are involved in similar biological processes.

**Transcription factor inter-regulation network of yeast** As a last example, we mention the inter-regulation network of yeast's transcription factors (TF). This network is constructed based on the data of [19]. It is consist of 62 vertices (transcription factors) and 91 edges. An edge is included between two transcription factors if one TF regulates the expression of the gene of the other TF.

The community structure of this network is found to have the form of fig. 5. Similar to the protein-protein interaction network, the elements of this biological network are also organized hierarchically. More work has to be done to understand the biological meaning of this hierarchical organization. Here we mention the following observation: According to fig. 5, the TFs *YAP6*, *ROX1*, *YAP1*, *CIN5*, *SKN7*, and *NRG1* form part of the core of this hierarchical network. On the other hand, when we removed from the original (directed) network [19] those TFs that are not regulating the expression of other TFs, we found that the whole network reduces to a network of 22 elements, where *YAP6*, *ROX1*, and *CIN5* form a central loop and all the remaining TFs serve as inputing elements.

Efforts is being undertaken by the present authors to integrate the protein-protein interaction network and the gene regulation network of budding yeast into a comprehensive network of protein-protein interaction and regulation.

## 5 Conclusion and discussion

In this paper, we have discussed biased Brownian motion on networks. A quantitative measure for the degree of proximity between neighboring vertices of a network was described based on this concept of biased network Brownian motion. This proximity index integrates both the local and the global structural information of a given network. Based on this proximity measure, we have constructed a powerful algorithm, called *Netwalk*, of network community structure identification.

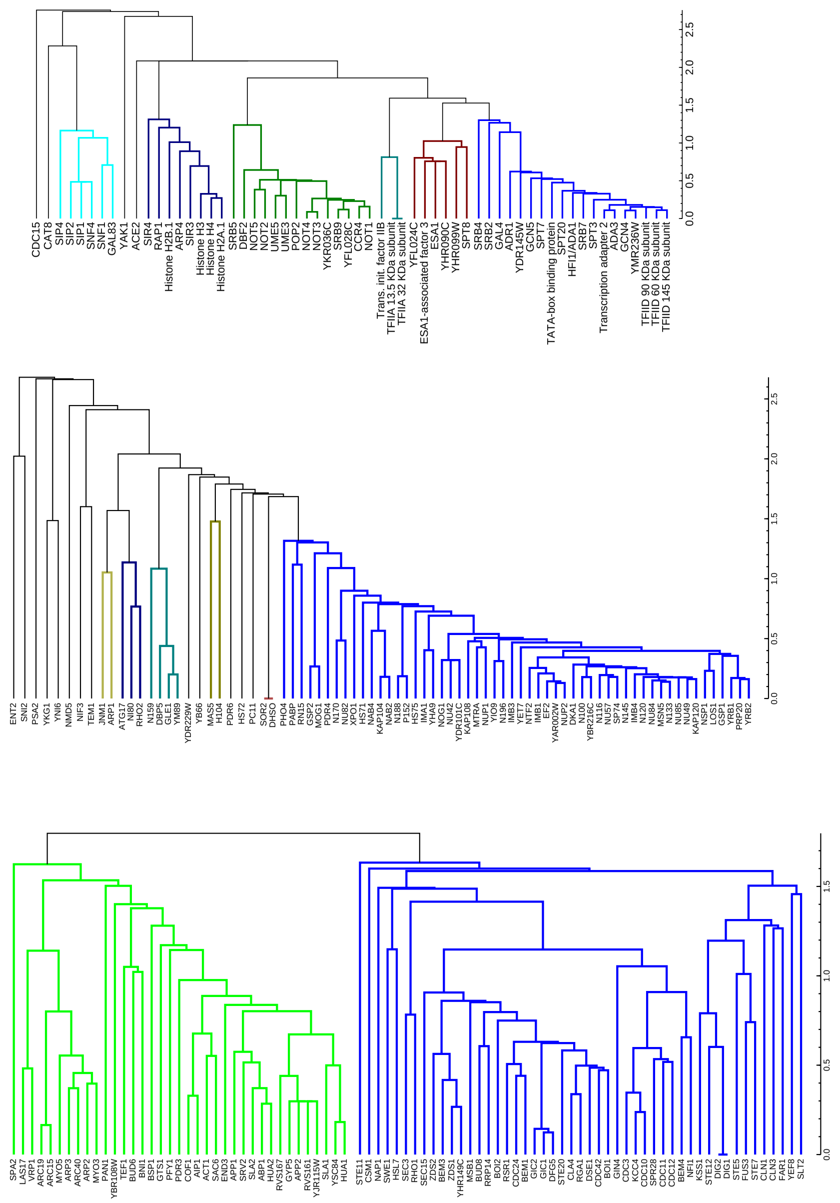


**Fig. 4.** Three subcommunities of the protein-protein interaction network of yeast at the deepest level of hierarchy of fig. 3. (Upper) A group of nuclear proteins involved in gene transcription. (Middle) Another group of nuclear proteins including various nucleoporins and proteins involved in importing proteins into and exporting proteins from the cell nucleus. (Bottom) A group of cytoskeletal proteins, among them are some proteins involved in cell division.

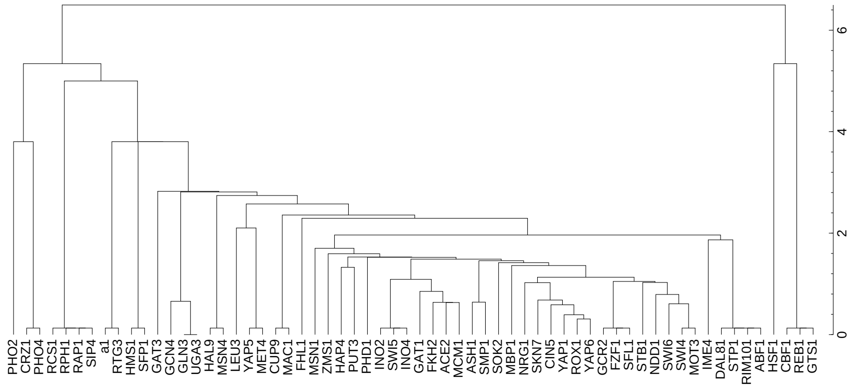


**Fig. 5.** Community structure of yeast's transcription factor regulation network predicted by *Netwalk*.

We have tested the performance of *Netwalk* on random modular networks and found good performance. The algorithm was then applied to two real-world social networks and to two biological networks. For the two biological networks, namely (i) the protein-protein interaction network and (ii) the inter-regulation network of transcription factors, the communities are organized in a hierarchical way. More work is needed to understand the evolutionary origin of this hierarchical organization and its biological significance.

The *Netwalk* algorithm includes a matrix inversion operation, and its computation time scales as $O(N^3)$, where $N$ is the total number of vertices of the network of interest. For very large networks $N \gg 10^3$, it is impractical to calculate exactly the value of the proximity index as given by eq. (6). An approximate scheme is as follows. To calculation $\Lambda(i,j)$, one may first construct a subnetwork of (say) $N_s = 1000$ elements including vertices $i$, $j$ and their nearest-neighbors, next-nearest-neighbors, etc., and all the edges between these elements. An estimate of $\Lambda(i,j)$ can then be obtained by applying eqs. (5) and (6) on this subnetwork. Because of the scale-free property of many real-world networks [2], we expect the value of the proximity index obtained by this method to be a good approximation of the exact value. If this approximate scheme is used, all the vertex-vertex proximity indices can be calculated in a computational time that scales linearly with the total number of edges.

For sparse networks of size $N < 10^3$, the *Netwalk* algorithm is comparable in computational time and performance with the graph-theoretical Girvan-Newman algorithm [14]. The advantages of the statistical-physics based algorithm are as follows: (i) It is applicable to weighted networks. Therefore it is able to uncover some structure even for a densely connected network, provided the edges of this network have different weights. (ii) It could be easily extended to very large graphs as discussed above. (iii) The local environment of each vertex is included by the bias coefficient $\gamma$ in eq. (1).

We are confident that both the vertex-vertex proximity measure and the *Netwalk* algorithm described in this paper will find more applications in social and biological networked systems.

## References


1. Strogatz, S. H.: Nature **410** (2001) 268-276
2. Albert, R., Barabási, A.-L.: Rev. Mod. Phys. **74** (2002) 47-97
3. Dorogovtsev, S. N., Mendes, J. F. F.: Adv. Phys. **51** (2002) 1079-1187
4. Jespersen, S., Sokolov, I. M., Blumen, A.: Phys. Rev. E **62** (2000) 4405-4408
5. Tadic, B.: Eur. Phys. J. B **23** (2001) 221-228
6. Adamic, L. A., Lukose, R. M., Puniyani, A. R., Huberman, B. A.: Phys. Rev. E **64** (2001) 046135
7. Guimera, R., Diaz-Guilera, A., Vega-Redondo, F., Cabrales, A., Arenas, A.: Phys. Rev. Lett. **89** (2002) 248701
8. Holme, P.: Adv. Compl. Sys. **6** (2003) 163-176
9. Zhou, H.: Phys. Rev. E **67** (2003) 041908
10. Zhou, H.: Phys. Rev. E **67** (2003) 061901
11. Newman, M. E. J.: e-print: cond-mat/0309045 (2003)
12. Noh, J. D., Rieger, H.: e-print: cond-mat/0307719 (2003)
13. Zachary, W. W.: J. Anthropol. Res. **33** (1977) 452-473
14. Girvan, M., Newman, M. E. J.: Proc. Natl. Acad. Sci. U.S.A. **99** (2002) 7821-7826
15. Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Kim, S. M., Eisenberg, D.: Nucleic Acids Res. **30** (2002) 303-305
16. Deane, C. M., Salwinski, L., Xenarios, I., Eisenberg, D.: Mol. Cell. Proteomics **1** (2002) 349-356
17. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., Barabási, A.-L.: Science **297** (2002) 1551-1555
18. Holme, P., Huss, M., Jeong, H.: Bioinformatics **19** (2003) 532-538
19. Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K., Young, R. A.: Science **298** (2002) 799-804