Protein ranking: From local to global structure in the protein similarity network

Jason Weston[†‡], Andre Elisseeff[‡], Dengyong Zhou[‡], Christina S. Leslie[§], and William Stafford Noble[¶‖]

[†]NEC Laboratories America, 4 Independence Way, Princeton, NJ 08540; [‡]Max Planck Institute for Biological Cybernetics, Spemannstrasse 38, 72076 Tübingen, Germany; [§]Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, MC 0401, New York, NY 10027; and [¶]Department of Genome Sciences, University of Washington, Health Sciences Center, P.O. Box 357730, Seattle, WA 98195

Edited by Michael S. Waterman, University of Southern California, Los Angeles, CA, and approved March 16, 2004 (received for review December 4, 2003)

Biologists regularly search databases of DNA or protein sequences for evolutionary or functional relationships to a given query sequence. We describe a ranking algorithm that exploits the entire network structure of similarity relationships among proteins in a sequence database by performing a diffusion operation on a precomputed, weighted network. The resulting ranking algorithm, evaluated by using a human-curated database of protein structures, is efficient and provides significantly better rankings than a local network search algorithm such as PSI-BLAST.

Pairwise sequence comparison is the most widely used application of bioinformatics. Subtle sequence similarities frequently imply structural, functional, and evolutionary relationships among protein and DNA sequences. Consequently, essentially every molecular biologist working today has searched an online database of biosequences. This search process is analogous to searching the World Wide Web with a search engine such as Google: the user enters a query (a biological sequence or a word or phrase) into a web form. The search engine then compares the query with each entry in a database, and returns to the user a ranked list, with the most relevant or most similar database entry at the top of the list.

The World Wide Web consists of a network of documents connected to one another by means of hypertext links. A database of protein sequences can also be usefully represented as a network, in which edges may represent functional, structural, or sequence similarity. Two protein sequences are considered similar if they contain subsequences that share more similar amino acids than would be expected to occur by chance. We refer to the network of sequence similarities as a protein similarity network.

Early algorithms for detecting sequence similarities did not exploit the structure of the protein similarity network at all, but focused instead on accurately defining the individual edges of the network (1–3). Subsequent work used statistical models based on multiple alignments to model the local structure of the network (4, 5) and to perform local search through the protein similarity network by using short paths (6), average- or single-linkage scoring of inbound edges (7, 8), and iterative model-based search (9, 10). The popular PSI-BLAST (11) algorithm falls into the latter category: PSI-BLAST builds an alignment-based statistical model of a local region of the protein similarity network and then iteratively collects additional sequences from the database to be added to the alignment.

The critical innovation that led to the success of the Google search engine is its ability to exploit global structure by inferring it from the local hyperlink structure of the Web. Google's PAGERANK algorithm (12) models the behavior of a random web surfer, who clicks on successive links at random and also periodically jumps to a random page. The web pages are ranked according to the probability distribution of the resulting random walk. Empirical results show that PAGERANK is superior to the naive, local ranking method, in which pages are simply ranked according to the number of inbound hyperlinks.

We demonstrate that a similar advantage can be gained by including information about global network structure in a protein sequence database search algorithm. In contrast to iterative protein database search methods such as PSI-BLAST, which compute the local structure of the protein similarity network on the fly, the RANKPROP algorithm begins from a precomputed protein similarity network, defined on the entire protein database. Querying the database consists of adding the query sequence to the protein similarity network and then propagating link information outward from the query sequence. After propagation, database proteins are ranked according to the amount of link information they received from the query. This algorithm ranks the data with respect to the intrinsic cluster structure (13, 14) of the network. We evaluate the RANKPROP output by using a 3D-structure-based gold standard, measuring the extent to which known homologs occur above nonhomologs in the ranked list. Our experiments suggest that RANKPROP's ranking is superior to the ranking induced by the direct links in the original network.

The protein similarity network represents the degree of similarity between proteins by assigning weights to each edge. The degree of similarity between two sequences is commonly summarized in an E value, which is the expected number of times that this degree of sequence similarity would occur in a random database of the given size. By using a weighting scheme that is a function of the E value, an edge connecting two similar sequences is given a large weight, and *vice versa*.

To accommodate edge weights, the RANKPROP algorithm adopts recently described diffusion techniques (15) from the field of machine learning, which are closely related to the spreading activation networks of experimental psychology (16, 17). RANKPROP takes as input a weighted network on the data, with one node of the network designated as the query. In the protein ranking problem, the edges of the network are defined by using PSI-BLAST. The query is assigned a score, and this score is continually pumped to the remaining points by means of the weighted network. During the diffusion process, a protein P pumps to its neighbors at time t the linear combination of scores that P received from its neighbors at time $t - 1$, weighted by the strengths of the edges between them. The diffusion process continues until convergence, and the points are ranked according to the scores they receive. The RANKPROP algorithm is described formally in Fig. 1. This algorithm provably converges, and an exact closed form solution can be found (see *Supporting Information*, which is published on the PNAS web site).

Methods

We tested the quality of the protein rankings produced by RANKPROP, using the human-annotated SCOP database of protein 3D structural domains as a gold standard (18). SCOP has

This paper was submitted directly (Track II) to the PNAS office.

Abbreviation: ROC, receiver operating characteristic.

‖To whom correspondence should be addressed. E-mail: noble@gs.washington.edu.

1. **Initialization:** $y_1(0) = 1;\ y_i(0) = 0$
2. **for** $t = 0, 1, 2, \ldots$ **do**
3. **for** $i = 2$ **to** m **do**
4. $y_i(t+1) \leftarrow K_{1i} + \alpha \Sigma_{j=2}^{m} K_{ji} y_j(t)$
5. **end for**
6. **until convergence**
7. **Termination:** Let y_i^* denote the limit of the sequence $\{y_i(t)\}$. Then y_i^* is the ranking score of the i^{th} point (largest ranked first).

Fig. 1. The RANKPROP algorithm. Given a set of objects (in this case, proteins) $X = x_1, \ldots, x_m$, let x_1 be the query and $x_2, \ldots, x_m$ be the database (targets) we would like to rank. Let K be the matrix of object–object similarities, i.e., K_{ij} gives a similarity score between x_i and x_j, with K normalized so that $\Sigma_{j=2}^{m} K_{ji} = 1$ for all i. For computational efficiency, we set $K_{1i} = K_{i1}$ for all i, so that we can compute weights involving the query using a single execution of PSI-BLAST. Let y_i, $i = 2, \ldots, m$, be the initial ranking "score" of a target. In practice, for efficiency, the algorithm is terminated after a fixed number I of iterations, and $y_i(I)$ is used as an approximation of y_i^*. The parameter $\alpha \in [0,1]$ is set *a priori* by the user. For $\alpha = 0$, no global structure is found, and the algorithm's output is just the ranking according to the original distance metric. These experiments use $\alpha = 0.95$, looking for clear cluster structure in the data.

been used as a gold standard in many previous studies (19–21). Sequences were extracted from version 1.59 of the database, purged by using the web site http://astral.berkeley.edu so that no pair of sequences share more than 95% identity. For the purposes of selecting the RANKPROP parameter σ, the resulting collection of 7,329 SCOP domains was split into two portions: 379 superfamilies (4,071 proteins) for training and 332 (2,899 proteins) for testing. Note that training and testing sequences never come from the same superfamily. The SCOP database is organized hierarchically into classes, folds, superfamilies, and families. For the purposes of this experiment, two domains that come from the same superfamily are assumed to be homologous, and two domains from different folds are assumed to be unrelated. For pairs of proteins in the same fold but different superfamilies, their relationship is uncertain, and so these pairs are not used in evaluating the algorithm.

Three protein similarity networks were computed by using the BLAST and PSI-BLAST (version 2.2.2) algorithms. Two networks were defined by applying BLAST and PSI-BLAST to a database comprised only of the 7,329 SCOP domains. An additional network was created by applying PSI-BLAST to a larger database that also included all 101,602 proteins from SWISS-PROT (version 40). In each case, the programs were run by using the default parameters, including the BLOSUM 62 matrix, but with an E value threshold for reporting results of 10,000. PSI-BLAST was allowed to run a maximum of six iterations, which previous work indicates is sufficient for good performance (21), using the default E value threshold for inclusion in the model of 0.005. Each of these networks induces a ranking with respect to each query sequence.

Finally, we applied RANKPROP to the larger PSI-BLAST protein similarity network. In the network K used by RANKPROP, the weight K_{ij} associated with a directed edge from protein i to protein j is $exp(-S_j(i)/\sigma)$, where $S_j(i)$ is the E value assigned to protein i given query j. The value of $\sigma = 100$ is chosen by using the training set (see supporting information). For efficiency, the number of outgoing edges from each node is capped at 1000, unless the number of target sequences with E values <0.05 exceeds 1000. For each query, RANKPROP runs for 20 iterations, which brings the algorithm close to convergence (see supporting information).

We measure the performance of a protein database search algorithm by using a modified version of the receiver operating characteristic (ROC) score (22). The ROC score is the area under a curve that plots false-positive rate versus true-positive rate for various classification thresholds. The ROC score thus measures, for a single query, the quality of the entire ranking produced by the algorithm. In practice, only the top of this ranking is important. Therefore, we compute the ROC_{50} score (23), which is the area under the ROC curve up to the first 50 false-positives. A value of 1 implies that the algorithm successfully assigns all of the true relationships higher scores than the false relationships. For a random ranking of these data, the expected ROC_{50} score is close to 0 because most of the sequences are not related to the query.

Results

The experimental results, summarized in Fig. 2, show the relative improvements offered by the various algorithms. Even when using the small SCOP database, the PSI-BLAST protein similarity network improves significantly upon the network created using the simpler BLAST algorithm: PSI-BLAST yields better performance than BLAST for 51.3% of the test queries, and worse performance for only 8.2% of the queries. PSI-BLAST benefits from the availability of a larger sequence database: increasing the database size by adding the SWISS-PROT database yields an additional improvement of the same magnitude (50.9% and 11.4%, respectively). Finally, running RANKPROP on the larger protein similarity network defined by PSI-BLAST yields improved rankings for 55.3% of the queries, and decreases performance on only 9.7%. All of these differences are statistically significant at $P = 0.01$ according to a Wilcoxon signed-rank test. A comparison of PSI-BLAST and RANKPROP ROC scores by query is shown in Fig. 3, and a diagram illustrating how RANKPROP successfully re-ranks homologs of a single query is shown in Fig. 4.

Note that there is some obvious structure in Figs. 2 and 3. The steep slope in the RANKPROP plot (Fig. 2) at around 0.9 ROC_{50} corresponds to queries mostly from the largest superfamily in the database, the immunoglobulins with 623 proteins. These queries are also visible as a cluster at around (0.9, 0.7) in Fig. 3. RANKPROP's improved rankings for these queries suggests that the algorithm successfully exploits cluster structure in the protein similarity network.

RANKPROP is not misled by the presence of multidomain proteins in the database. Previous network-based protein similarity detection algorithms explicitly deal with multidomain proteins. For example, the INTERMEDIATE SEQUENCE SEARCH algorithm (6) includes a step that extracts the region of the target sequence that matched the query and then recalculates the statistical significance of that region with respect to the target sequences. This step prevents the algorithm from inferring a false relationship between protein domains A and B through an intermediate protein containing both A and B. RANKPROP delivers excellent performance, even when the database contains $\approx$100,000 full-length proteins, many of which contain more than one domain. Furthermore, Fig. 3 shows that RANKPROP generally performs better than PSI-BLAST, even when the SCOP query domain lies on the same protein as another domain in the test set. A closer investigation (see supporting information) reveals that RANKPROP does indeed rank these transitive domains higher than would be expected by chance. However, in general, as long as the query sequence is connected to many other proteins, then the true relationships will be mutually reinforcing during network propagation.

A well known problem with PSI-BLAST is the occasional case in which it mistakenly pulls in a false-positive match during an early iteration. This false-positive may then pull in more false-positives

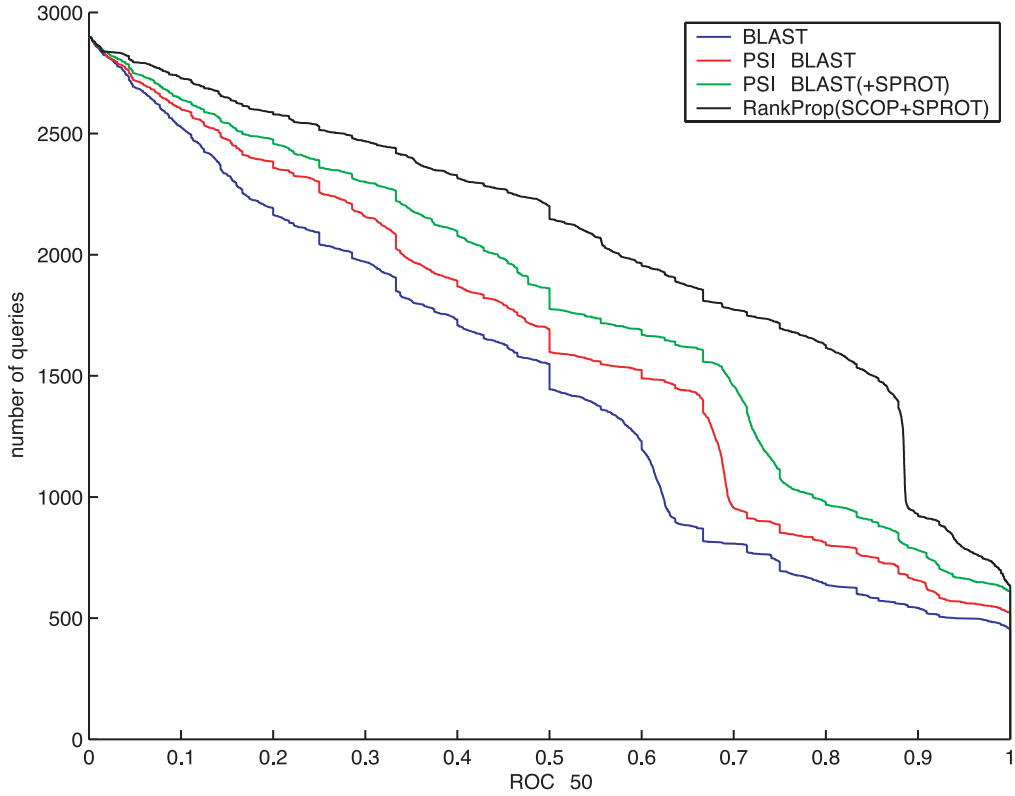

Fig. 2. Relative performance of protein ranking algorithms. The graph plots the total number of test set SCOP queries for which a given method exceeds an ROC_{50} score threshold. ROC_{50} is the area under a curve that plots true-positive rate as a function of false-positive rate, up to the 50th false-positive. In the plot, the lower three series correspond to the three protein similarity networks described in the text; the upper series is created by running RANKPROP on the larger PSI-BLAST network. For these data, the mean ROC_{50} for the four methods are 0.506 (BLAST), 0.566 [PSI-BLAST (SCOP)], 0.618 [PSI-BLAST (SCOP plus SPROT)], and 0.707 (RANKPROP).

in subsequent iterations, leading to corrupted results. Among the test set queries, there are 139 queries for which the PSI-BLAST ROC_{50} score is worse than the corresponding BLAST score, indicating that iteration hurt the performance of the algorithm. For these queries, RANKPROP outperforms BLAST in 106 cases, despite using as input a protein similarity network defined by PSI-BLAST. Furthermore, the degree of improvement produced by RANKPROP relative to BLAST is often large, with a difference in ROC_{50} >0.1 for 71 of the 106 queries (see supporting information).

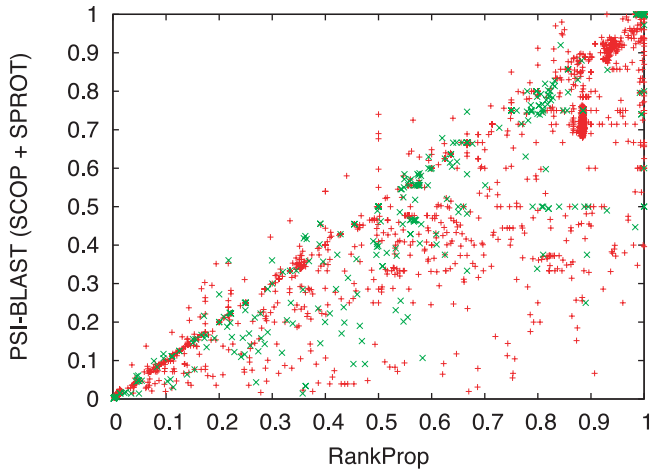

Fig. 3. Scatter plot of ROC_{50} scores for PSI-BLAST versus RANKPROP. The plot contains 2,899 points, corresponding to all queries in the test set. Green points correspond to query domains that lie on the same protein with another domain in the test set. All other queries are red.

Among the 282 queries for which PSI-BLAST produces a better ranking than RANKPROP, most of the differences in ROC are small. There are, however, 20 queries for which PSI-BLAST produces an ROC_{50} that is >0.1 greater than RANKPROP's ROC_{50}, and one query for which the difference is >0.2 (see supporting information). Some of these queries belong to SCOP class 3 (α-β proteins), which contains a number of homologous Rossmann folds. In these cases, the first false-positives may in fact be true-positives. For the other queries, RANKPROP's difficulty likely arises from overpropagation through the protein similarity network. Lowering the parameter α could potentially fix this problem, because as $\alpha \to 0$, we obtain the same ranking as PSI-BLAST.

Finally, the results indicate that RANKPROP does not spoil good initial rankings. Indeed, there is only one query for which PSI-BLAST produces an ROC_{50} score of 1 (a perfect ranking) and RANKPROP produces a score worse than 0.98. This query is the C-terminal fragment of DNA topoisomerase II, with an ROC_{50} of 0.93. Conversely, there are 30 queries for which PSI-BLAST has an ROC_{50} <0.93 and RANKPROP produces a perfect ranking.

To better understand the source of RANKPROP's improvement relative to the underlying PSI-BLAST protein similarity network, we performed an additional round of experiments using two variants of the RANKPROP algorithm. Each algorithmic variant restricts RANKPROP to a subset of the protein similarity network. In the first variant, RANKPROP sees only the local network structure: the target sequences that are linked directly to the query, plus the pairwise relationships among those sequences.

GENETICS

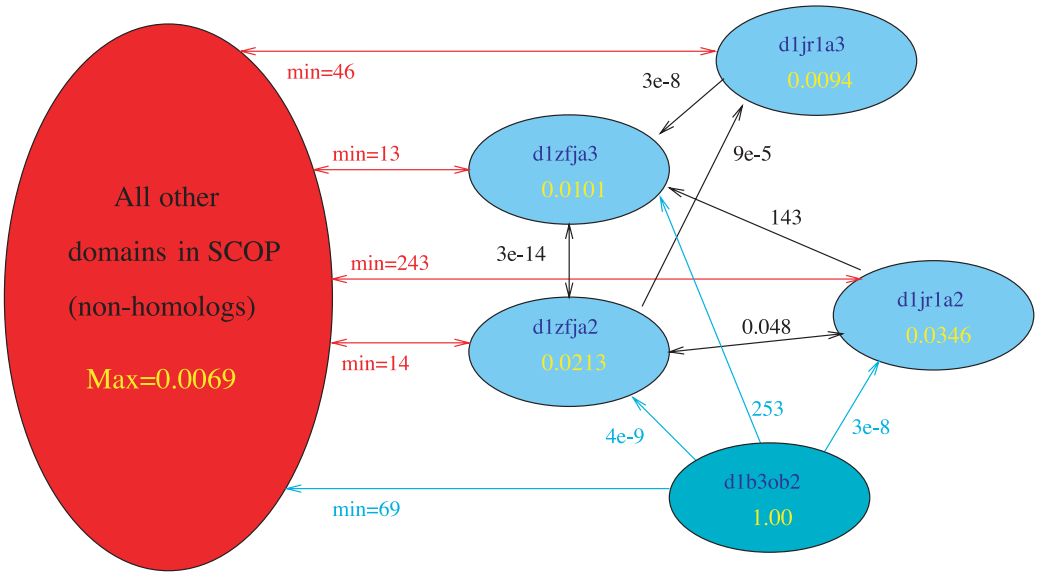

Fig. 4. Visualization of part of the similarity network. Shown is a small part of the protein similarity network, where d1b30b2 is the query, and the domains are represented by light blue nodes are its homologs. The large red node represents all other domains. The cyan-colored edges from the query to other nodes are labeled with weights equal to the PSI-BLAST *E* value, given d1b30b2 as the query. The rest of the edges indicate the similarity network which is formed of PSI-BLAST *E* values, as described in the text. Black edges are between homologs, and red edges are between all nonhomologs and a single homolog, with the minimum *E* value across all nonhomologs given as the weight of the edge. No edge is drawn if PSI-BLAST did not assign an *E* value. PSI-BLAST only correctly identifies two homologs, d1zfja2 and d1jr1a2. Although d1zfja3 is assigned an *E* value (of 253), this assignment is larger than three of the nonhomologs in SCOP. The yellow scores inside the nodes are the RANKPROP activation levels (y_i values). In this case, RANKPROP places all of the homologs at the top of the ranked list. This assignment occurs because there are very low *E* value paths (by traversing edges with not more than an *E* value of 9e-5) between the query and all homologs, whereas even the ''nearest'' nonhomologs are sufficiently far away (never closer than an *E* value of 13 to a homolog). Note that d1jr1a2 is assigned a higher score by RANKPROP than d1zfja2, even though the *E* values assigned by PSI-BLAST (although similar) indicate the opposite. This result is because d1zfja2 has much lower weighted edges to nonhomologs, and thus receives more of the nonhomologs' activation level (which are close to 0). Overall, the RANKPROP ranking gave an ROC_{50} score of 1, whereas PSI-BLAST gave an ROC_{50} score of 0.78 on this query.

This network of local relationships yields RANKPROP performance almost identical to PSI-BLAST (see supporting information). The second variant includes nonlocal edges but eliminates all weak edges, with *E* values >0.005. In contrast with the previous variant, this version of the algorithm performs only slightly worse than RANKPROP trained using the entire network. This result indicates that the improvement of RANKPROP over PSI-BLAST results primarily from RANKPROP's ability to learn from nonlocal network structure, and that the weak links in the network are of secondary importance. Data sets and FASTA files are available from the web site of J.W., which can be accessed at www.kyb.tuebingen.mpg.de/bs/people/weston/rankprot/supplement.html.

Discussion

RANKPROP is efficient enough to employ the algorithm as part of a web-based search engine. The precomputation of the PSI-BLAST protein similarity network is clearly computationally expensive; however, this operation can be performed in advance offline. Computing the ranking with respect to a given query requires first running PSI-BLAST with the query sequence (unless it is already in the network), and then propagating scores from the query through the network. In the experiments reported here, the propagation (20 iterations of RANKPROP) took on average 73 seconds to compute using a Linux machine with an Advanced Micro Devices (Sunnyvale, CA) MP 2200+ processor. BLAST and PSI-BLAST take $\approx$21 and 331 sec per query respectively on the same database (SCOP plus SPROT). The propagation time scales linearly in the number of edges in the network. The propagation time could be improved by removing weak edges from the protein similarity network [at a relatively small cost in accuracy (see supporting information)], by running the propagation in parallel, and by reducing the number of iterations. Finally, the initial query PSI-BLAST computation may be replaced with BLAST at a relatively small cost in accuracy (see supporting information), resulting in a query procedure that is faster than running a single PSI-BLAST query on the entire database.

The experiments described here were performed by using a single set of PSI-BLAST parameters. These parameters were previously selected by means of extensive empirical optimization using the SCOP database as a gold standard and ROC_n scores as the performance metric (17). However, even if better PSI-BLAST parameters were available, the resulting improved *E* values would likely lead to a similar improvement in the performance of the RANKPROP algorithm.

The results reported here are given in terms of the ROC_{50} performance measure. One might argue that a stricter (or looser) threshold might be more appropriate, depending on the cost associated with false-positives. Further experiments (see supporting information) show that RANKPROP continues to significantly outperform PSI-BLAST even for relatively small values of the ROC threshold (ROC_5 or ROC_{10}). At the most strict threshold, ROC_1 (which is equivalent to the percentage of positive examples appearing before the first negative example in the ranked output), the difference between the two algorithms is no longer statistically significant. However, by using the ROC_1 measure, RANKPROP performs better on smaller superfamilies using a small σ, and *vice versa*. Therefore, a simple modification to the algorithm, in which the value of σ depends on the number of strong matches to the query sequence, once again yields strong performance relative to PSI-BLAST. In future work, we plan to investigate more thoroughly algorithms that choose σ dynamically based on the local density of the protein similarity network.

A valuable component of the PSI-BLAST algorithm is its method for estimating statistical confidence, in the form of *E* values. Currently, RANKPROP does not produce *E* values; however,

approximate E values may be derivable by means of interpolation and smoothing of the PSI-BLAST E values with respect to the RANKPROP ranking. Alternatively, it may be possible to fit a probability distribution to the output scores (24). This fitting will be the subject of future research.

The primary outcome of this work is not the RANKPROP algorithm *per se*, but the observation that exploiting the entire structure of the protein similarity network can lead to significantly improved recognition of pairwise protein sequence similarities. RANKPROP provides an efficient, powerful means of learning from the protein similarity network; however, other network-based algorithms may also yield similar improvements relative to the ranking induced by the underlying protein similarity network. Furthermore, this observation is applicable to a wide range of problem domains, including image and text ranking, as well as protein or gene ranking using different (or multiple) types of biological data.

This work is supported by National Science Foundation Awards EIA-0312706 and DBI-0078523, National Institutes of Health Grant LM07276–02, and an Award in Informatics from the Pharmaceutical Research and Manufacturers of America Foundation (to C.S.L.). W.S.N. is an Alfred P. Sloan Research Fellow.

1. Smith, T. & Waterman, M. (1981) *J. Mol. Biol.* **147,** 195–197.
2. Pearson, W. R. (1985) *Methods Enzymol.* **183,** 63–98.
3. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) *J. Mol. Biol.* **215,** 403–410.
4. Gribskov, M., Lüthy, R. & Eisenberg, D. (1990) *Methods Enzymol.* **183,** 146–159.
5. Krogh, A., Brown, M., Mian, I., Sjolander, K. & Haussler, D. (1994) *J. Mol. Biol.* **235,** 1501–1531.
6. Park, J., Teichmann, S. A., Hubbard, T. & Chothia, C. (1997) *J. Mol. Biol.* **273,** 1–6.
7. Grundy, W. N. (1998) *Proceedings of the Second International Conference on Computational Molecular Biology*, eds. Istrail, S., Pevzner, P. & Waterman, M. (ACM, New York), pp. 94–100.
8. Yona, G., Linial, N. & Linial, M. (1999) *Proteins Struct. Funct. Genet.* **37,** 360–678.
9. Tatusov, R. L., Altschul, S. F. & Koonin, E. V. (1994) *Proc. Natl. Acad. Sci. USA* **91,** 12091–12095.
10. Hughey, R. & Krogh, A. (1996) *Comput. Appl. Biosci.* **12,** 95–107.
11. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997) *Nucleic Acids Res.* **25,** 3389–3402.
12. Brin, S. & Page, L. (1998) *Comput. Networks ISDN Syst.* **30,** 107–117.
13. Roweis, S. T. & Saul, L. K. (2000) *Science* **290,** 2323–2326.
14. Tenenbaum, J. B., de Silva, V. & Langford, J. C. (2000) *Science* **290,** 2319–2323.
15. Zhu, X., Ghahramani, Z. & Lafferty, J. (2003) in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, eds. Fawcett, T. & Mishra, N. (AAAI Press, Menlo Park, CA), pp. 329–336.
16. Anderson, J. R. (1983) *The Architecture of Cognition* (Harvard Univ. Press, Cambridge, MA).
17. Shrager, J., Hogg, T. & Huberman, B. A. (1987) *Science* **236,** 1092–1094.
18. Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995) *J. Mol. Biol.* **247,** 536–540.
19. Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. & Chothia, C. (1998) *J. Mol. Biol.* **284,** 1201–1210.
20. Jaakkola, T., Diekhans, M. & Haussler, D. (1999) *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, eds. Lengauer, T., Schneider, R., Bork, B., Brutlag, D., Glasgow, J., Mewes, H.-W. & Zimmer, R. (AAAI Press, Menlo Park, CA), pp. 149–158.
21. Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V. & Altschul, S. F. (2001) *Nucleic Acids Res.* **29,** 2994–3005.
22. Hanley, J. A & McNeil, B. J. (1982) *Radiology (Easton, Pa.)* **143,** 29–36.
23. Gribskov, M & Robinson, N. L. (1996) *Comput. Chem.* **20,** 25–33.
24. Platt, J. C. (1999) in *Advances in Large Margin Classifiers*, eds. Smola, A., Bartlett, P., Schölkopf, B. & Schuurmans, D. (MIT Press, Cambridge, MA), pp. 61–74.