the FEBS Journal

MINIREVIEW

Identifying remote protein homologs by network propagation

William S. Noble[1], Rui Kuang[2], Christina Leslie[3] and Jason Weston[4]

1 Department of Genome Sciences Department of Computer Science and Engineering University of Washington Seattle, WA, USA
2 Department of Computer Science, Columbia University, New York, NY, USA
3 Center for Computational Learning Systems, Columbia University, New York, NY, USA
4 NEC Laboratories America, Princeton, NJ, USA

Keywords
network diffusion; protein homology; protein networks; sequence comparison

Correspondence
W. S. Noble, Department of Genome Sciences Department of Computer Science and Engineering University of Washington Seattle, WA, USA
Fax: +1 206 685 7301
Tel: +1 206 543 8930
E-mail: noble@gs.washington.edu

(Received 25 May 2005, revised 19 August 2005, accepted 30 August 2005)

doi:10.1111/j.1742-4658.2005.04947.x

Perhaps the most widely used applications of bioinformatics are tools such as PSI-BLAST for searching sequence databases. We describe a recently developed protein database search algorithm called RANKPROP. RANKPROP relies upon a precomputed network of pairwise protein similarities. The algorithm performs a diffusion operation from a specified query protein across the protein similarity network. The resulting activation scores, assigned to each database protein, encode information about the global structure of the protein similarity network. This type of algorithm has a rich history in associationist psychology, artificial intelligence and web search. We describe the RANKPROP algorithm and its relatives, and we provide evidence that the algorithm successfully improves upon the rankings produced by PSI-BLAST.

Introduction

Networks abound in the scientific literature these days. Some of these networks (gene regulatory networks, metabolic networks, protein–protein interaction networks) represent real biological phenomena. Other networks are useful abstractions that allow for formal reasoning to occur.

Recently, we described a network-based algorithm for detecting subtle protein sequence similarities [1]. This algorithm, called RANKPROP, performs a diffusion operation on a network of pairwise protein similarity relationships. The network itself is an abstraction, in which edges are defined using a protein sequence comparison algorithm such as SMITH–WATERMAN [2], BLAST [3], FASTA [4] or PSI-BLAST [5]. In our work, we use PSI-BLAST to define the network. Given a query sequence, RANKPROP produces a ranking of all the proteins in the network. Thus, RANKPROP's output is similar to the output of PSI-BLAST. However, RANKPROP's ranking relies not only upon the similarities identified by PSI-BLAST, but also upon the global network topology. Exactly how this is accomplished will be made clear below. In a cross-validated test of structural classification of proteins (SCOP) superfamily recognition, RANKPROP consistently produces better rankings than PSI-BLAST. This result indicates that the network topology provides significant value in identifying false positive and false negative relationships in the underlying protein similarity network.

In this minireview, we situate the RANKPROP algorithm with respect to the bioinformatics and network inference literatures. We also describe the algorithm itself in some detail, attempting to provide some intuitions for how

Abbreviations
HMM, hidden Markov model; MCL, Markov cluster; PYP, photoactive yellow protein; ROC, receiver operating characteristic; SCOP, structural classification of proteins.

the diffusion adds value to the existing network. Rankings produced by the RANKPROP algorithm are now available through the UC Santa Cruz Gene Sorter, http://genome.ucsc.edu.

Protein database search

Over the past 25 years, researchers have developed a battery of successively more powerful methods for detecting protein sequence similarities. Here we focus on algorithms that take as input a single query sequence and a protein database, and produce as output a ranking of that database with respect to the query. Although the protein similarity network is an abstraction defined for the RANKPROP algorithm, we can relate previous database search methods to this network.

Early algorithms did not exploit the structure of the protein similarity network at all, but focused instead on accurately defining the individual edges of the network. The scores assigned to these edges induce the output ranking. The NEEDLEMAN–WUNSCH [6] and SMITH–WATERMAN [2] dynamic programming algorithms find a provably optimal pairwise alignment between a user-provided query sequence and a target sequence from a database. However, optimality is only guaranteed with respect to a very simple model of evolution. Furthermore, in practice, these dynamic programming algorithms are slow, especially when run on computers of the early 1980s. Hence, the increasing size of GenBank necessitated the development of approximation algorithms like BLAST [3] and FASTA [4]. These algorithms run much more quickly, but at the expense of possibly missing some significant alignments.

Various approaches have been suggested for performing local search through the protein similarity network defined by algorithms such as BLAST. These methods search for short paths in the network [7], or use average- or single-linkage scoring of inbound edges [8,9]. The average-linkage approach was developed in the context of the ProtoMap project, which was one of the first to explicitly represent protein similarities as a network.

Profiles [10] and hidden Markov models (HMMs) [11,12] provide a more principled means of performing local network search. These methods use statistical models based upon multiple alignments to model the local structure of the network. The resulting model can then be compared to a target sequence. Because the model contains more information than the original query sequence, this comparison can yield statistically significant results that would be missed by a purely pairwise approach. Published results suggest that, for a given false positive rate, these family based methods allow the computational biologist to infer nearly three times as many homologies as a simple pairwise alignment algorithm [13]. Profiles and HMMs cannot directly solve the single-query search problem because they require multiple sequences for training; however, these models have been used successfully in the context of iterative search.

Iterative search algorithms traverse the protein similarity network. This approach was suggested early on [14] and was popularized by the SAM-T98 HMM software [15] and, to a greater degree, by PSI-BLAST [5]. These methods build an alignment-based statistical model of a local region of the protein similarity network and then iteratively collect additional sequences from the database to be added to the alignment. Note, however, that the search procedure is local and relies upon the ability to multiply align all of the modeled sequences with respect to the query. The RANKPROP algorithm does not rely upon a multiple alignment, and makes use of the entire protein similarity network.

The RANKPROP algorithm

The RANKPROP algorithm is surprisingly simple. Furthermore, although it can be computationally quite expensive, most of the computation occurs in the generation of the protein similarity network, before the user issues a query. The query stage is very fast.

In a protein similarity network, the edges represent similarities between pairs of proteins in the database. We use PSI-BLAST to define this network, though in theory the network could be computed using any pairwise sequence comparison algorithm. Associated with each edge in the network is a weight that quantities the degree of similarity between the proteins. This weight, w, is derived from the PSI-BLAST E-value, E, via the following transformation: $w = e^{-E/\sigma}$, where σ is a parameter of the algorithm. How the value of σ is set is described below. The weights associated with edges leading into a given node are then normalized to a sum of 1. Thus, one can think of the network as defining probabilistic transitions between proteins. Given a starting protein, we can successively choose random numbers and probabilistically travel through the protein similarity network according to the transition probabilities on the edges.

Querying the network consists of two steps. First, assuming that the query is not already in the network, PSI-BLAST is run to connect the query to the rest of the network. Second, an activation score of 1.0 is assigned to the query node, and this score is ‘pumped’ through the entire protein similarity network. This pumping, or diffusion, operation is iterative, with the activation score at node y_i at time $t + 1$ defined as the sum of two terms: the initial score from the query, and the weighted sum of all scores coming from the neighbors of y_i:

$$y_i(t+1) \leftarrow K_{1i} + \alpha \sum_{j=2}^{m} K_{ji} y_j(t)$$

where K_{ji} is the weight associated with the edge connecting the node *i* to node *j*, and node 1 is the query node. The term α controls the rate of diffusion of activation scores through the network. The RANKPROP algorithm essentially performs a probabilistic traversal of the network across all paths leading away from the query node. The output of the algorithm is the list of all nodes (proteins) in the network, ranked by activation score. A protein's rank reflects the number, length and strength of edges along the paths connecting the query to that protein.

To understand intuitively how RANKPROP successfully re-ranks proteins, consider the toy example shown in Fig. 1. This simple network contains two groups of homologous proteins (represented by gray and white nodes) that are not related to one another. We assume that the pairwise comparison algorithm has correctly identified all the homology relationships with two exceptions: one gray protein has not been linked to the query (false negative) and one white protein has been incorrectly linked to the query (false positive). RANKPROP successfully identifies these errors by examining the rest of the network. The relationships among the gray nodes allows a high level of activation to reach the false negative node. Conversely, the lack of connections from the query to the other white nodes allows the activation score initially assigned to the false positive query to diffuse through the white nodes.

A more realistic example is shown in Fig. 2. In order to illustrate how RANKPROP diffusion improves upon the rankings induced by the underlying protein similarity network, we focus on a particular query domain, photoactive yellow protein (PYP) from *Ectothiorhodospira halophila* which, in previously reported results [1], yields good performance from RANKPROP but not from PSI-BLAST. This protein is a member of the PYP-like sensor domain SCOP superfamily [16], which in our experiment contains five protein domains. Our initial experiment used a database of over 100 000 proteins, including protein domain sequences of known structure from SCOP as well as protein sequences from SWISS-PROT. Because visualizing such a large network is difficult, here we extract a relevant subnetwork by considering only paths from the query domain to three members of the PYP-like sensor domain superfamily and three false positives. The false positives are SCOP domains from other superfamilies which are ranked highly by PSI-BLAST or RANKPROP. The one remaining superfamily member, histidine kinase FixL heme domain from *Rhizobium meliloti* (D1EW0A), is linked to the query domain with a densely connected subnetwork, which is too large to include for the purposes of visualization. Furthermore, we display only proteins on paths that are shorter than five edges, and for

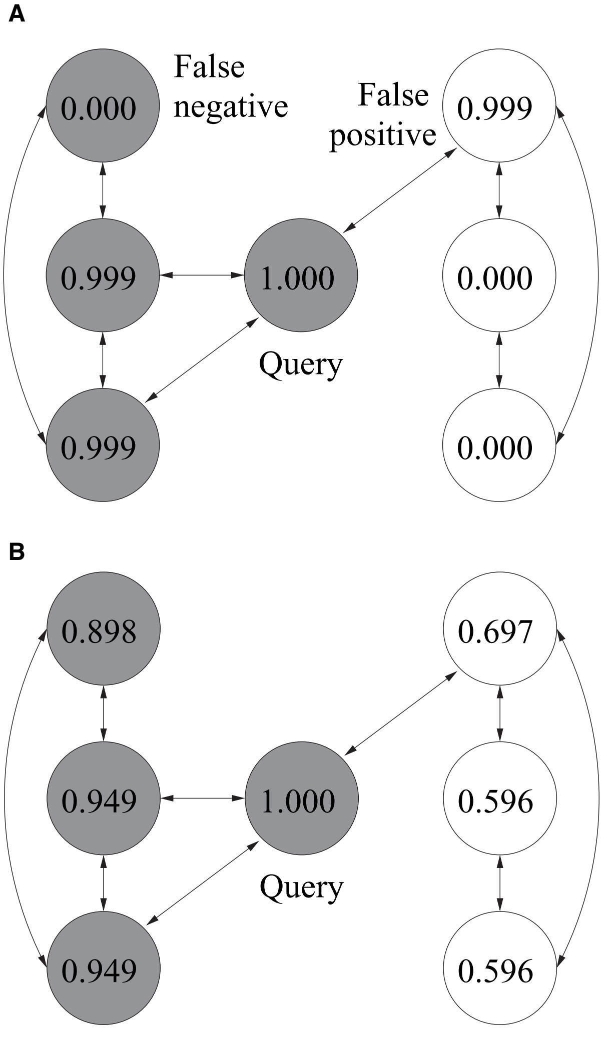
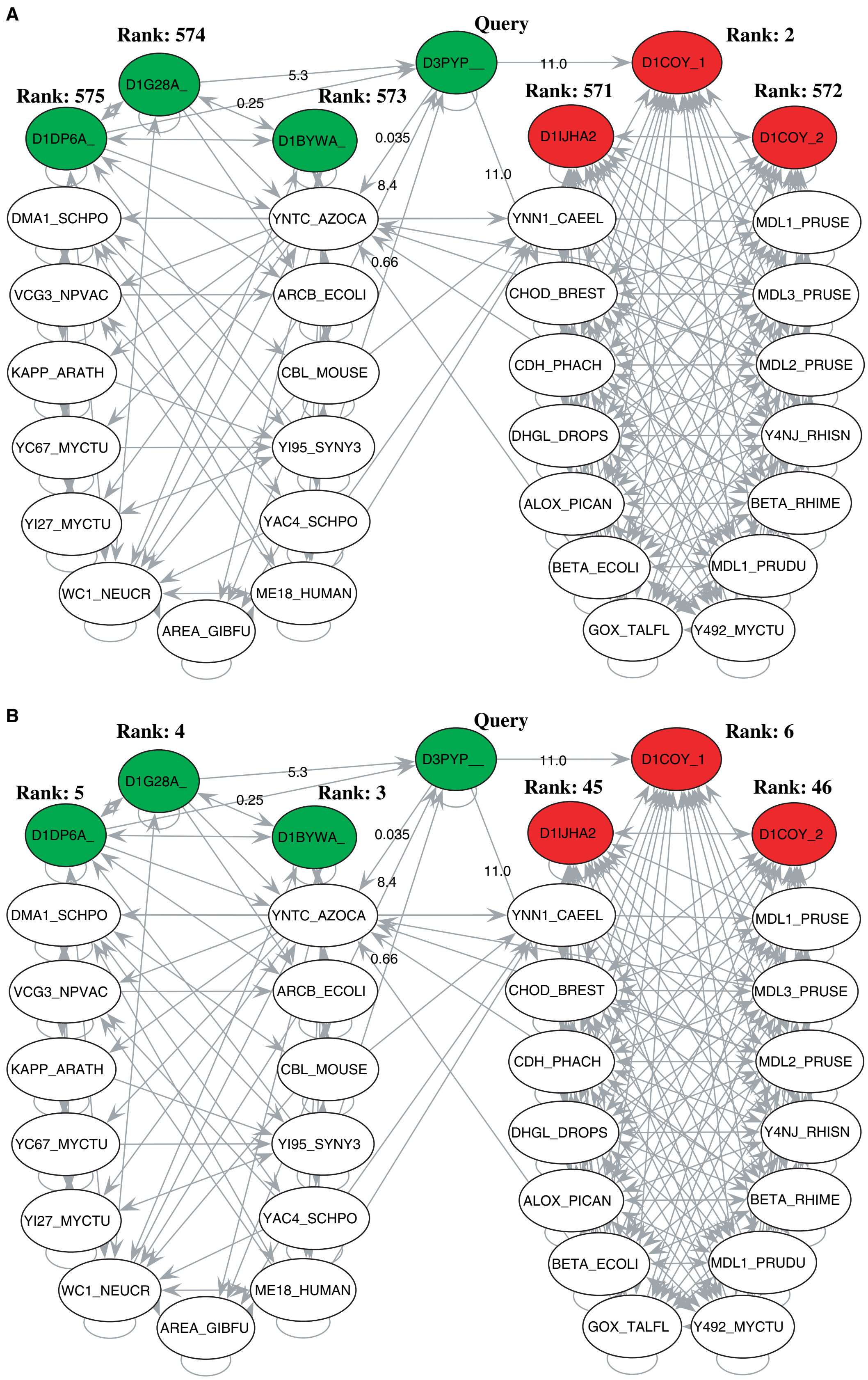


Fig. 1. RANKPROP uses network topology to re-rank proteins. (A) The figure shows a seven protein network. We assume that all gray nodes represent proteins that are homologous to one another, and that the white nodes represent a separate class of proteins that are homologous to one another but not to the proteins represented by the gray nodes. The pairwise comparison algorithm has assigned edges nearly correctly: the only mistakes are the missing edge between the query and the protein labeled 'false negative' and the extra edge between the protein labeled 'false positive'. Each node is labeled with its initial activation score, computed assuming that each edge has an E-value of 0.1. (B) After running the RANKPROP algorithm, the nodes receive activation scores that correctly re-rank the false positive and the false negative.


A
Query
Rank: 574
Rank: 575
Rank: 573
Rank: 2
Rank: 571
Rank: 572
D3PYP__
D1G28A_
D1DP6A_
D1BYWA_
D1COY_1
D1IJHA2
D1COY_2
5.3
0.25
0.035
8.4
11.0
11.0
0.66
DMA1_SCHPO
VCG3_NPVAC
KAPP_ARATH
YC67_MYCTU
YI27_MYCTU
WC1_NEUCR
AREA_GIBFU
ME18_HUMAN
YAC4_SCHPO
YI95_SYNY3
CBL_MOUSE
ARCB_ECOLI
YNTC_AZOCA
YNN1_CAEEL
CHOD_BREST
CDH_PHACH
DHGL_DROPS
ALOX_PICAN
BETA_ECOLI
GOX_TALFL
Y492_MYCTU
MDL1_PRUDU
BETA_RHIME
Y4NJ_RHISN
MDL2_PRUSE
MDL3_PRUSE
MDL1_PRUSE
B
Query
Rank: 4
Rank: 5
Rank: 3
Rank: 6
Rank: 45
Rank: 46
D3PYP__
D1G28A_
D1DP6A_
D1BYWA_
D1COY_1
D1IJHA2
D1COY_2
5.3
0.25
0.035
8.4
11.0
11.0
0.66
DMA1_SCHPO
VCG3_NPVAC
KAPP_ARATH
YC67_MYCTU
YI27_MYCTU
WC1_NEUCR
AREA_GIBFU
ME18_HUMAN
YAC4_SCHPO
YI95_SYNY3
CBL_MOUSE
ARCB_ECOLI
YNTC_AZOCA
YNN1_CAEEL
CHOD_BREST
CDH_PHACH
DHGL_DROPS
ALOX_PICAN
BETA_ECOLI
GOX_TALFL
Y492_MYCTU
MDL1_PRUDU
BETA_RHIME
Y4NJ_RHISN
MDL2_PRUSE
MDL3_PRUSE
MDL1_PRUSE

which each edge on the path has an E-value no larger than 0.1. The resulting network contains 34 proteins and is shown in Fig. 2A. In the initial ranking produced by PSI-BLAST (Fig. 2A), three PYP-like sensor domains are ranked very low, while a false positive, cholesterol oxidase of the glucose-methanol-choline (GMC) family from *Brevibacterium sterolicum* (D1COY_1), is ranked higher. Although there is no edge directly from the query to the three other PYP-like sensor domains, all four are linked to a set of strongly connected proteins from SWISS-PROT, some of which are connected to the query. On the other hand, the false positive D1COY_1 has fewer supporting connections from the query in this network. Thus, after running RANKPROP, all the true superfamily members are ranked correctly above nonsuperfamily members.

Other network-based propagation algorithms for homology tasks

Other recent work has also proposed diffusion algorithms defined on different kinds of protein networks for homology-related tasks. The MARKOV CLUSTER (MCL) algorithm [17], designed for clustering nodes in a graph by simulating stochastic flow, has been used to detect protein families in large sequence databases [18]. In this task, the MCL algorithm performs multiple rounds of random walks on a similarity network of proteins and then decomposes the network into components, each of which represents a candidate protein family. Similar to RANKPROP, the MCL algorithm uses a similarity network defined by a symmetric connectivity matrix between proteins weighted by their sequence similarity and normalized to be stochastic. The MCL algorithm makes random walks by alternately taking expansion and inflation operations to update the connectivity matrix K as follows:

$$\text{Expansion}: K = K^{\text{n}}$$

$$\text{Inflation}: K_{\text{ij}} = (K_{\text{ij}})^r / \sum_{q=1}^{m} (K_{\text{qj}})^r$$

where K^{n} is the matrix product of K for n times, m is the row dimension of K, and r is a real number larger than 1. The expansion step boosts the probabilities between nodes in the same cluster, because random walks connect members of the same cluster more frequently than between members of different clusters. On the other hand, the inflation step re-scales the transition probabilities by favoring links with higher scores. As in RANKPROP, the MCL algorithm captures global cluster structure in graphs but uses a two-step bootstrapping procedure. This bootstrapping procedure provably converges to an equilibrium state, separating the graph into isolated subgraphs with no flow between them (i.e., edges between these subgraphs have zero weight in the limit). The MCL algorithm has also been successfully applied in many other problem domains [19–21] besides protein family detection.

Another recent propagation algorithm is MOTIFPROP [22], which like RANKPROP is applied to the protein remote homology detection problem. Instead of relying on a pairwise similarity score between proteins, the MOTIFPROP algorithm assumes that shared sequence motifs are capable of capturing the cluster structure among proteins. A protein-motif similarity network, a bipartite graph defined by a connectivity matrix between proteins and motifs, is constructed for this purpose. Starting with the connectivity matrix H and initial activation values on protein nodes and motif nodes, MOTIFPROP takes a two-step diffusion operation to update activation scores of protein nodes and motifs by

$$P^{t+1} = \alpha \tilde{H} F^t + (1 - \alpha) P^0$$

$$F^{t+1} = \alpha \tilde{H}' P^t + (1 - \alpha) F^0$$

where parameter $\alpha \in (0,1)$ balances between the diffusion information and initial activation scores, $\tilde{H}$ is obtained from H by normalizing so that entries in each row sum to 1 and $\tilde{H}'$ is a similarly row-normalized version of the transpose of H. F^0 is the vector of initial motif activation values, and P^0 is the vector of initial activation values from the base ranking algorithm, each normalized so that entries sum to 1. The vector P^0 can be initialized in the same way as in RANKPROP, and the components of F^0 can be estimated based on some statistical measures for different motif sets [22]. By inducing a ranking of motifs along with the ranking of database sequences, MOTIFPROP provides additional information useful for discovering common structural components between remote homologies and also improves the sensitivity of remote homology detection.

Fig. 2. RANKPROP improves the recognition of the PYP-like sensor domain superfamily. (A) The figure shows the protein similarity network. Green nodes are members of the PYP-like sensor domain superfamily. White nodes are Swiss-Prot sequences with no known structure, and red nodes are SCOP proteins from a different SCOP fold. Each node is labeled with the protein ID and rank before the first iteration of RANKPROP. Edges to/from the query domain are labeled with *E*-values. (B) This network is the same as the one in (A), except that the ranks have been computed after 20 iterations of RANKPROP. In both networks, only edges with *E*-values less than 0.1 are displayed.

In other related work, a procedure to enforce symmetry, applied to a large binary connectivity matrix, has proved helpful for detection of multidomain protein sequences during protein clustering and reduction of false positives due to transitive domains [23]. This kind of algorithm does not use a diffusion operation but does take advantage of an implicit protein similarity network through processing of a connectivity matrix.

Ranking in other domains

The protein homology detection task can be usefully compared to many other ranking tasks, such as searching the web or ranking images. In a protein database search, the input is a user query (the amino acid sequence of a protein) and a given database of proteins, and the output is a ranking of the given database. In a web search, the input is a query term (text from part of a web page) and a database of web pages, and again the output is a ranking of the database. In several other such domains, algorithms similar to RANKPROP have been very successful.

For example, one of the best performing web search algorithms is PAGERANK [24], which drives the popular Google website. 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. PAGERANK works by making the assumption that when one page links to another page, it is effectively casting a (weighted) vote for that other page. The more votes that are cast for a page, the more important the page must be. Moreover, the importance of the page that is casting the vote determines how important the vote itself is. These ranking scores are calculated through a so-called spreading activation network: each page propagates its score to its neighbors via its outbound links and alters its score based upon the received scores from its inbound links, according to the formula

$$y_{\mathrm{j}}(t+1) = (1-\alpha) + \alpha \sum_{i} \frac{K_{\mathrm{ij}} y_{\mathrm{i}}(t)}{C_{\mathrm{i}}}$$

where y_{j} denotes the page rank of web page j, and $K_{\mathrm{ij}} = 1$ if page i links to page j, and 0 otherwise. $C_{\mathrm{i}} = \sum_p K_{\mathrm{ip}}$ is the number of outbound links of page i, and α is a damping factor (usually set to 0.85). In practice, the propagation is usually iterated a small number of times, e.g. up to $t = 40$ time steps. (PAGERANK corresponds to computing the principal eigenvector of the normalized link matrix of the web, and can hence be computed in closed form, rather than by iteration, but at greater computational expense.) 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.

The idea of spreading activation, however, dates back further than PAGERANK. In [25], spreading activation is defined as a class of algorithms that propagate numerical values (activation levels) in a network for the purpose of selecting the nodes that are most closely related to the source of the activation. As such, the model is related to associationist models of thought, traceable to Freud and Pavlov and, ultimately, to Aristotle [26].

Spreading activation was first described as a computational process by Quillian [27], who showed how it can be used to search a semantic network, comparing and contrasting word-senses in a network structured dictionary database. The original idea was to spread activation not from all nodes concurrently (as in PAGERANK) but from a set of nodes, or a single node query:

$$y_{\mathrm{j}}(t+1) = C_{\mathrm{j}}(t) + \gamma y_{\mathrm{j}}(t) + \alpha \sum_{i} K_{\mathrm{ij}} y_{\mathrm{i}}(t)$$

where $C_{\mathrm{j}}(t)$ is the external input for node j at time step t and γ is the relaxation rate, chosen between 0 and 1. In a typical application, some nodes (the sources) are activated by external inputs and these in turn cause others to become active with varying intensities. Such algorithms have been used in various artificial intelligence systems [27,28] and as a component of computational models of memory in cognitive psychology [26,29,30].

More recently, in [31], the convergence of a similar algorithm to (1) is shown, and a closed form expression is given. The propagation approach is shown to outperform a local distance measure approach in the problems of image ranking (given a query image) and text document ranking (given a query text document). Finally, most recently, because the success of the RANKPROP algorithm, the authors of [32] have also applied the RANKPROP algorithm to content based image retrieval with iterative feedback, with state of the art results.

Validation of the RANKPROP algorithm

The RANKPROP algorithm has been validated using a gold standard derived from protein structure. SCOP [16] is a hierarchical organization of protein domains into classes based upon structural characteristics. Each group, defined at the superfamily level of the hierarchy, contains protein domains that are presumed to be homologous to one another, whereas protein

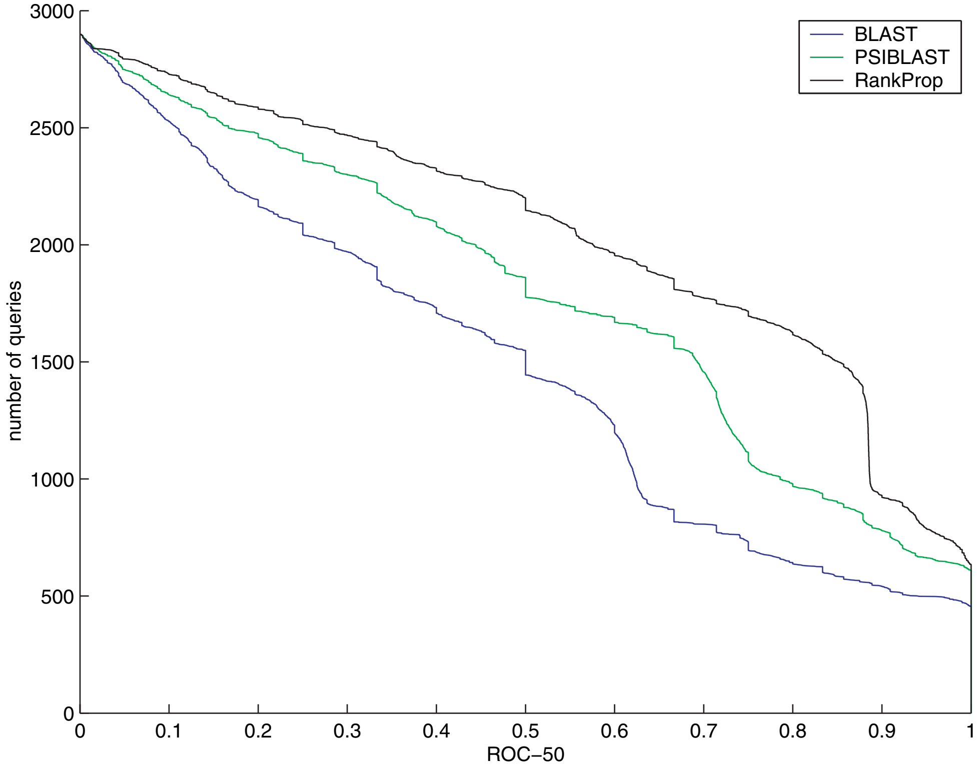


Fig. 3. Comparison of RANKPROP performance with BLAST AND PSI-BLAST. The figure plots the percentage of queries (out of 2899) for which a given protein ranking algorithm achieves a specified ROC_{50} score. The three series correspond to the RANKPROP algorithm, PSI-BLAST using the default inclusion threshold of 0.005 and a maximum of six iterations, and BLAST. More details are provided in [1].

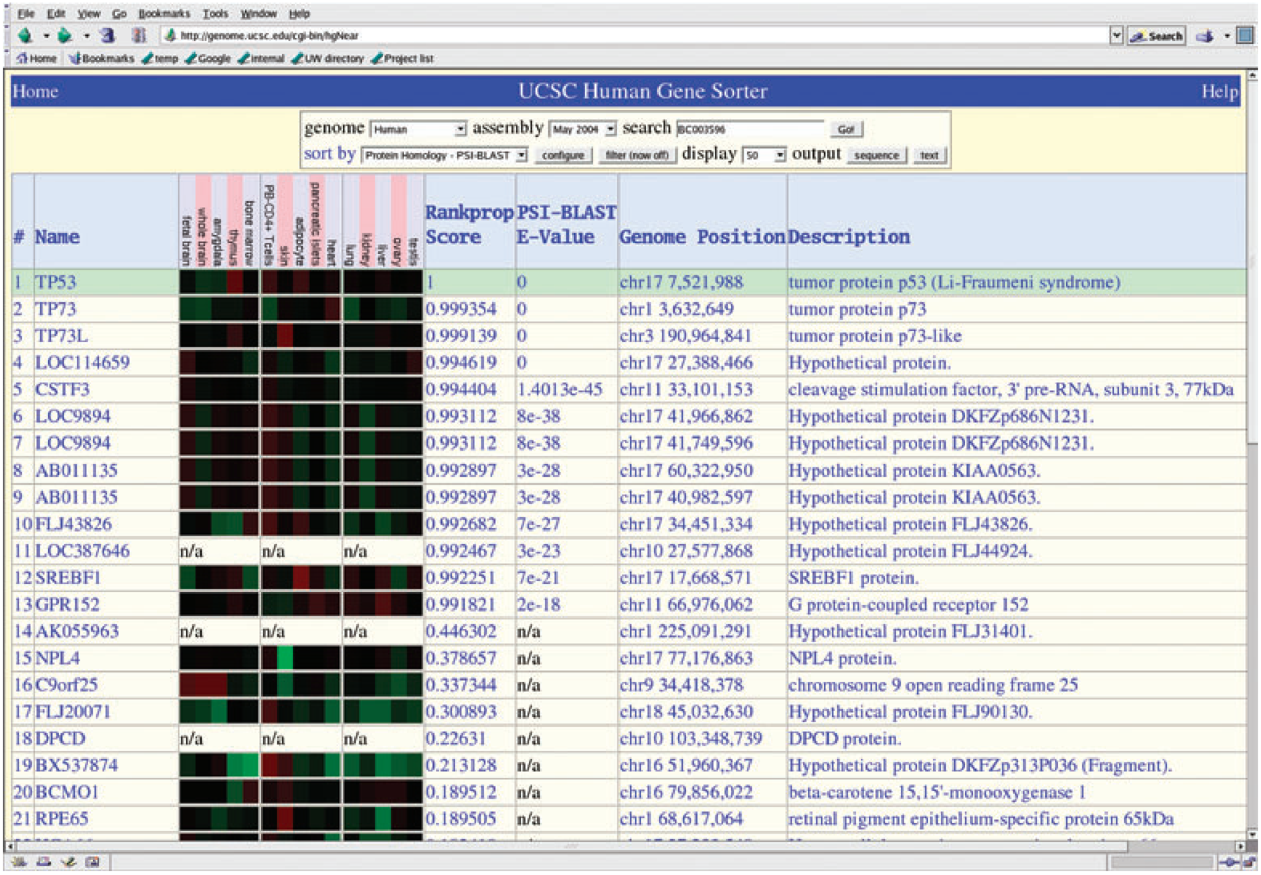
domains within one fold group share structural similarity but may not be homologous. Following the design used in other experiments (e.g. [33]), we consider a pair of domains to be homologous if they are in the same superfamily, and unrelated if they are in different folds. Protein pairs that are in the same fold but different superfamilies have an uncertain relationship and hence are not used in the validation.

Figure 3 compares the performance of RANKPROP to BLAST and PSI-BLAST. The database consists of 108 931 proteins, which includes 7329 SCOP domains and 101 602 complete proteins from Swiss-Prot. For each SCOP domain in a predefined test set of 2899 proteins, we rank the entire database, extract the SCOP domains, and label each one as 'true' if it is in the same superfamily as the query, 'false' if it is in a different fold, and 'unknown' if it is in the same fold as the query but a different superfamily. To evaluate the quality of a ranking, we compute receiver operating characteristic (ROC) scores [34] with respect to the ranked list of 'true' and 'false' labels. More specifically, the ROC score is the normalized area under a ROC curve, which plots true positives as a function of false positives at different thresholds. By putting all true positives ahead of true negatives, a perfect ranking algorithm will have a ROC score of 1 while a random ranking algorithm will receive a ROC score of 0 5. For this particular task, because we are interested in the quality of the top of the ranking, we compute the ROC_{50} score [35]; i.e., the area under the ROC curve up to the first 50 false positives. The figure shows a dramatic improvement in the quality of the rankings induced by RANKPROP.

The RANKPROP algorithm has two parameters that can be set by the user: the diffusion constant α and the σ parameter used in converting E-values to edge weights. For the SCOP experiments, we set these parameters using a separate set of queries, choosing the parameter values ($\alpha = 0.95$ and $\sigma = 100$) that yield optimal performance.

RANKPROP on the UCSC Gene Sorter

Although the RANKPROP algorithm is quite simple and the source code is publicly available (http://www.kyb.tuebingen.mpg.de/bs/people/weston/rankprot/supplement.html), computing a protein similarity network can be very computationally expensive. We have


File Edit View Go Bookmarks Tools Window Help
http://genome.ucsc.edu/cgi-bin/hgNear
Search
Home Bookmarks temp Google internal UW directory Project list
Home
UCSC Human Gene Sorter
Help

genome Human assembly May 2004 search BC003596 Go!
sort by Protein Homology - PSI-BLAST configure filter (now off) display 50 output sequence text

#	Name	fetal brain whole brain amygdala thymus bone marrow PB-CD4+ Tcells skin adipocyte pancreatic islets heart lung kidney liver ovary testis	Rankprop Score	PSI-BLAST E-Value	Genome Position	Description
1	TP53		1	0	chr17 7,521,988	tumor protein p53 (Li-Fraumeni syndrome)
2	TP73		0.999354	0	chr1 3,632,649	tumor protein p73
3	TP73L		0.999139	0	chr3 190,964,841	tumor protein p73-like
4	LOC114659		0.994619	0	chr17 27,388,466	Hypothetical protein.
5	CSTF3		0.994404	1.4013e-45	chr11 33,101,153	cleavage stimulation factor, 3' pre-RNA, subunit 3, 77kDa
6	LOC9894		0.993112	8e-38	chr17 41,966,862	Hypothetical protein DKFZp686N1231.
7	LOC9894		0.993112	8e-38	chr17 41,749,596	Hypothetical protein DKFZp686N1231.
8	AB011135		0.992897	3e-28	chr17 60,322,950	Hypothetical protein KIAA0563.
9	AB011135		0.992897	3e-28	chr17 40,982,597	Hypothetical protein KIAA0563.
10	FLJ43826		0.992682	7e-27	chr17 34,451,334	Hypothetical protein FLJ43826.
11	LOC387646	n/a n/a n/a	0.992467	3e-23	chr10 27,577,868	Hypothetical protein FLJ44924.
12	SREBF1		0.992251	7e-21	chr17 17,668,571	SREBF1 protein.
13	GPR152		0.991821	2e-18	chr11 66,976,062	G protein-coupled receptor 152
14	AK055963	n/a n/a n/a	0.446302	n/a	chr1 225,091,291	Hypothetical protein FLJ31401.
15	NPL4		0.378657	n/a	chr17 77,176,863	NPL4 protein.
16	C9orf25		0.337344	n/a	chr9 34,418,378	chromosome 9 open reading frame 25
17	FLJ20071		0.300893	n/a	chr18 45,032,630	Hypothetical protein FLJ90130.
18	DPCD	n/a n/a n/a	0.22631	n/a	chr10 103,348,739	DPCD protein.
19	BX537874		0.213128	n/a	chr16 51,960,367	Hypothetical protein DKFZp313P036 (Fragment).
20	BCMO1		0.189512	n/a	chr16 79,856,022	beta-carotene 15,15'-monooxygenase 1
21	RPE65		0.189505	n/a	chr1 68,617,064	retinal pigment epithelium-specific protein 65kDa

Fig. 4. RANKPROP on the UC Santa Cruz Gene Sorter. The web interface allows the user to rank homologs of any protein in the human genome by RANKPROP activation score. The figure shows the ranking of proteins related to the *p53* tumor suppressor gene.

therefore made RANKPROP available via the UC Santa Cruz Gene Sorter at http://www.genome.ucsc.edu [36]. Figure 4 shows the browser interface. Here, homologs of the human *p53* gene have been ranked by RANKPROP activation score. These scores are computed in a network of all human proteins, with edges defined by PSI-BLAST. The Gene Sorter allows for ranking by BLAST E-value (symmetrized) PSI-BLAST E-value, or RANKPROP activation score, so the differences in rankings can be compared. In this particular case, RANKPROP suggests weak relationships with numerous proteins that PSI-BLAST did not identify.

Discussion

The RANKPROP algorithm provides a new, meta-level approach to the protein database search problem. The algorithm capitalizes on the decades of research that went into producing current, state of the art search algorithms such as PSI-BLAST; but RANKPROP also leverages information about the global topology of the protein similarity network. Our experiments indicate that the patterns of connectivity between the query and its neighbors and among the query's neighbors and their neighbors, etc., contain important information that allows RANKPROP to differentiate between correctly and incorrectly inferred homology relationships.

Because RANKPROP does not rely upon multiple alignments to the query sequence, it runs the risk of introducing false positive associations via multidomain proteins. Theoretically, a single-domain protein A which is homologous to a multidomain protein AB could lead to a false inference of homology between A and a single-domain protein B. However, our experiments [1] indicate that multidomain proteins do not cause a serious problem for RANKPROP. In practice, the single-domain protein B will receive a relatively high rank, but RANKPROP will successfully rank it below the true homologs. Nevertheless, to address this issue directly, and also to allow RANKPROP to provide explanatory output in addition to its ranking, we are currently developing variants of the algorithm that cut proteins in the network into shorter segments based on pairwise alignments. We also plan to augment the ranking output with a probabilistic score, allowing users to set a score threshold a priori. With these modifications, we expect that RANKPROP will provide fast, high-quality, user-friendly protein sequence database search results.

Acknowledgements

The authors thank Mark Diekhans and Jim Kent for assistance in creating the UCSC Gene Sorter interface to RankProp. This work is supported by NSF awards IIS-0093302, DBI-0243257 and EIA-0312706. W.S.N. is an Alfred P. Sloan Foundation Research Fellow.

References

1 Weston J, Elisseef A, Zhou D, Leslie C & Noble WS (2004) Protein ranking: from local to global structure in the protein similarity network. *Proc Natl Acad Sci USA* **101**, 6559–6563.

2 Smith T & Waterman M (1981) Identification of common molecular subsequences. *J Mol Biol* **147**, 195–197.

3 Altschul SF, Gish W, Miller W, Myers EW & Lipman DJ (1990) A basic local alignment search tool. *J Mol Biol* **215**, 403–410.

4 Pearson WR (1985) Rapid and sensitive sequence comparisions with FASTP and FASTA. *Methods Enzymol* **183**, 63–98.

5 Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W & Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389–3402.

6 Needleman S & Wunsch C (1970) A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J Mol Biol* **48**, 443–453.

7 Park J, Teichmann SA, Hubbard T & Chothia C (1997) Intermediate sequences increase the detection of homology between sequences. *J Mol Biol* **273**, 1–6.

8 Grundy WN (1998) Family-based homology detection via pairwise sequence comparison. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology* (Istrail S, Pevzner P & Waterman M, eds), pp. 94–100. ACM Press, New York, NY, USA.

9 Yona G, Linial N & Linial M (1999) Protomap: Automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins: Struct Funct Genet* **37**, 360–678.

10 Gribskov M, Luthy R & Eisenberg D (1990) Profile analysis. *Methods Enzymol* **183**, 146–159.

11 Krogh A & Riis SK (1999) Hidden neural networks. *Neural Computation* **11**, 541–563.

12 Baldi P, Chauvin Y, Hunkapiller T & McClure MA (1994) Hidden Markov models of biological primary sequence information. *Proc Natl Acad Sci USA* **91**, 1059–1063.

13 Park J, Karplus K, Barrett C, Hughey R, Haussler D, Hubbard T & Chothia C (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol Biol* **284**, 1201–1210.

14 Tatusov RL, Altschul SF & Koonin EV (1994) Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. *Proc Natl Acad Sci USA* **91**, 12091–12095.

15 Karplus K, Barrett C & Hughey R (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics* **14** (10), 846–856.

16 Murzin AG, Brenner SE, Hubbard T & Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* **247**, 536–540.

17 Van Dongen S (2000) A new cluster algorithm for graphs. (INS-R0011). National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.

18 Enright AJ, Van Dongen S & Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res* **30** (7), 1575–1584.

19 Li L, Stoeckert CJ & Roos DS (2003) OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Res* **13**, 2178–2189.

20 Pereira-Leal JB, Enright AJ & Ouzounis CA (2004) Detection of functional modules from protein interaction networks. *Proteins Struct Funct Bioinformat* **54**, 49–57.

21 Watson JD (2003) Target selection and determination of function in structural genomics. *Int Union Biochem Mol Biol Life* **55**, 249–255.

22 Kuang R, Weston J, Noble WS & Leslie C (2005) Motif-based protein ranking by network propagation. *Bioinformatics* doi: 10.1093/bioinformatics/bti608.

23 Enright AJ & Ouzounis CA (2000) Generage: a robust algorithm for sequence clustering and domain detection. *Bioinformatics* **16**, 451–457.

24 Brin S & Page L (1998) The anatomy of a large scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, pp. 107–117.

25 Shrager J, Hogg T & Huberman BA (1987) Observation of phase transitions in spreading activation networks. *Science* **236**, 1092–1094.

26 Anderson JR (1983) *The Architecture of Cognition.* Harvard University Press, Cambridge, MA.

27 Quillian MR (1968) *Semantic Information Processing* (Minsky M, ed.), pp 216–270. MIT Press, Cambridge, MA, USA.

28 Cohen PR & Stanhope PM (1986) *Proceedings of the 6th International Workshop on Expert Systems and Their Applications*. Avignon, France.

29 Howe A (1984) In *Proceedings of the Canadian Society for Computational Studies of Intelligence*, pp. 25–27. London, Ontario.

30 Collins AM & Loftus EF (1975) Using spreading activation to identify relevant help. *Psychol Rev* **82**, 407.

31 Zhou D, Weston J, Gretton A, Bousquet O & Schoelkopf B (2003) Ranking on data manifolds. *Adv Neural Info Processing Systems* **16**, 169–176.

32 He J, Li M, Zhang H, Tong H & Zhang C (2004) Manifold-ranking based image retrieval. In *Proceedings of 12th ACM International Conference on Multimedia*. ACM Press, New York, NY, USA.

33 Jaakkola T, Diekhans M & Haussler D (1999) Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pp. 149–158. AAAI Press, Menlo Park, CA.

34 Hanley JA & McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**, 29–36.

35 Gribskov M & Robinson NL (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computers Chem* **20**, 25–33.

36 Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM & Haussler D (2002) Human genome browser at UCSC. *Genome Res* **12**, 996–1006.