BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 13 2006, pages 1585–1592
doi:10.1093/bioinformatics/btl130

Structural bioinformatics

Integrating multi-attribute similarity networks for robust representation of the protein space

Orhan Çamoğlu[1,*], Tolga Can[2] and Ambuj K. Singh[1]

[1]Department of Computer Science, University of California, Santa Barbara, CA 93106, USA and [2]Department of Computer Engineering, Middle East Technical University, 06531, Ankara, Turkey

Received on August 29, 2005; revised on March 22, 2006; accepted on March 31, 2006
Advance Access publication April 4, 2006
Associate Editor: Thomas Lengauer

ABSTRACT

Motivation: A global view of the protein space is essential for functional and evolutionary analysis of proteins. In order to achieve this, a similarity network can be built using pairwise relationships among proteins. However, existing similarity networks employ a single similarity measure and therefore their utility depends highly on the quality of the selected measure. A more robust representation of the protein space can be realized if multiple sources of information are used.

Results: We propose a novel approach for analyzing multi-attribute similarity networks by combining random walks on graphs with Bayesian theory. A multi-attribute network is created by combining sequence and structure based similarity measures. For each attribute of the similarity network, one can compute a measure of affinity from a given protein to every other protein in the network using random walks. This process makes use of the implicit clustering information of the similarity network, and we show that it is superior to naive, local ranking methods. We then combine the computed affinities using a Bayesian framework. In particular, when we train a Bayesian model for automated classification of a novel protein, we achieve high classification accuracy and outperform single attribute networks. In addition, we demonstrate the effectiveness of our technique by comparison with a competing kernel-based information integration approach.

Availability: Source code is available upon request from the primary author.

Contact: orhan@cs.ucsb.edu

Supplementary Information: Supplementary data are available on *Bioinformatic* online.

1 INTRODUCTION

Understanding the overall organization of the protein universe is one of the most important problems in computational biology. Such a global view serves as a key component for the functional analysis of proteins, and reveals insights into the evolutionary processes. Being able to categorize proteins based on their sequence and structural characteristics is an important first step towards achieving this goal.

There have been a number of studies for analyzing the protein universe. Liu and Rost (2003) present a review of recent automated methods for clustering protein space. Many of these studies focus on clustering protein sequences because of the relative abundance of sequence data compared with structural data. Yona *et al*. (1999) present a large-scale analysis of protein space by building a global map (ProtoMap) based on similarity of sequences. As one application of such a global view, Portugaly *et al*. (2002) estimate the probability of a protein to have a new fold, by superimposing the known folds onto the protein sequence similarity network. In similar studies, Yona and Levitt (2000) and Pandit *et al*. (2002) organize clusters of protein sequences based on known structural information.

The recent growth in the number of structures in the Protein Data Bank (PDB) (Berman *et al*., 2000) has enabled new analysis methods to uncover the global organization of the protein structural space (Holm and Sander, 1996; Hou *et al*., 2003). Holm and Sander (1996) present a classification of the protein structure space based on the Dali (Holm and Sander, 1993) structure similarity scores. Hou *et al*. (2003) built a global three-dimensional map of the protein fold space in which structurally related folds are represented by spatially adjacent points. As more protein structures are solved, more accurate predictions about the layout of the protein universe will be possible. Moreover, a new view of the protein universe created by combining sequence and structure information will reveal the complex correlations between sequence and structure homology, and the process of evolution.

The advantages of integrating multiple data sources for predicting protein function has been demonstrated by a number of recent studies (Deng *et al*., 2004; Lanckriet *et al*., 2004; Pavlidis *et al*., 2001; Yamanishi *et al*., 2004). Deng *et al*. (2004) use Markov random field (MRF) for assigning function based on information on protein interactions, gene expression profiles, protein complex membership, and domain presence. The integration is manually designed, e.g. complex membership data are integrated into the model as prior probabilities while co-expression is a part of the interaction network model, therefore limiting the automated integration of new data sources. Moreover, because the model is learned for a given function, n separate models are required for an n-class classification problem. Kernel-based integration approaches (Lanckriet *et al*., 2004; Schoelkopf *et al*., 2004; Yamanishi *et al*., 2004) provide a more extensible way of integrating multiple data sources, since basic operations (e.g. addition, multiplication by a constant) can be used on kernels of single data sources to produce valid combined kernels. A number of kernels such as string kernels (Haussler, 1999; Lodhi *et al*., 2000; Saigo *et al*., 2004) have been especially developed for biological data (Schoelkopf *et al*., 2004).

In this paper, we propose a novel framework for analyzing multi-attribute similarity networks built using multiple pairwise

*To whom correspondence should be addressed.

© The Author 2006. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oxfordjournals.org

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

similarity measures. Our approach is based on random walks on graphs and exploits the global structure of the similarity network to infer protein relationships. The use of random walks over graphs for classification and clustering received considerable attention in recent years (Pan *et al*., 2004; Szummer and Jaakkola, 2001). The idea of random walks can be related to starting an energy diffusion process originating from a designated node of the similarity network and observing the final energies received by other nodes in the network. A similar approach has been used by Weston *et al*. (2004) to analyze the similarity network of protein sequences created using PSI-Blast (Altschul and Koonin, 1998) scores. The main difference in our approach is that we provide a Bayesian framework for merging random walk results of multiple similarity networks built using different similarity measures. Another difference is that, in our framework, nodes representing the categories, e.g. superfamilies, are part of the similarity network (Section 2.2). An obvious application of this enhancement is the computation of inter-category relationships (e.g. fold to fold) based on sequence/structure similarities of the proteins in the graph.

Our proposed framework is general enough to be applicable in different biological applications such as structural classification, functional classification, automated annotation and clustering. As a particular application for demonstrating the utility of our framework, we train a Bayesian model for automated structural classification of new proteins. Specifically, we use two sequence and three structure comparison methods to create five similarity networks, which are combined to create a multi-attribute similarity network. Our results show that when the affinities from different networks are combined using the Bayesian framework, higher accuracies are achieved compared with single-attribute networks. We also compare our proposed framework with a kernel-based integration approach by combining individual kernels created for different similarity measures using the SVM-Pairwise kernel (Liao and Noble, 2003), which has been shown to yield high performance for recognizing remote homologs. Experimental results show that our technique outperforms the kernel based method.

The main contributions of the paper are

- construction of a multi-attribute similarity network for proteins using sequence/structure-based measures,
- use of random walks on graphs for mining multi-attribute similarity networks,
- development of a general Bayesian framework for combining the results of random walks and
- application of the framework for automated structural classification of novel protein structures.

The rest of the paper is organized as follows. In Section 2, we present the idea of random walks on similarity networks. In Section 3, we present a Bayesian framework for combining a multi-attribute similarity network for the classification of new proteins. We present experimental results in Section 4 and conclude with a brief discussion in Section 5.

2 MULTI-ATTRIBUTE SIMILARITY NETWORKS

We define a similarity network on a database of proteins as a graph whose nodes represent individual proteins. Edges are established between proteins that show significant similarity, and the weight of an edge indicates the degree of similarity. The main advantage of such a similarity network over ranked immediate neighbor lists (as in traditional sequence/structure searches) is that it encodes the global structure of similarity relations among proteins (Weston *et al*., 2004).

When multiple information sources are used to establish the similarity network, the number of available edges provided by each source is not equal. In order to normalize the number of edges contributed from each information source, either a score-threshold or fixed-number-of-neighbors, i.e. *top-k*, approach should be adapted. The methodology used for establishing the edges is a parameter that is critical to the quality of the generated similarity network.

We adopt the *top-k* approach. The motivation is to overcome the possible non-uniformity of similarity scores, where larger proteins tend to get higher scores compared with small proteins. The differences between these two approaches are revisited in Section 4, where we evaluate the performance of various approaches: the score threshold approach and *top-k* approaches for $k \in \{1, 2, 5, 10\}$.

We create a separate similarity network for each information source/similarity measure. A similarity network for an information source is created by inserting protein–protein edges for similar proteins into the graph and normalizing the weights on these edges to the range [0, 1] such that the smallest similarity score corresponds to 0 and the highest similarity score corresponds to 1. The resulting set of similarity networks form a multi-attribute similarity network containing different types of edges with each type representing one similarity measure. In this paper, we use two sequence and three structure comparison methods to demonstrate that combining sequence and structure information provides a more accurate representation of the protein universe compared to using a single sequence/structure measure alone. For sequence based components of the similarity network, we choose two widely used sequence comparison methods, PSI-Blast (Altschul and Koonin, 1998), and HMMer (Eddy, 1998) on the SUPERFAMILY database (Gough, 2002) (HMMer is referred to as HMM throughout the paper). For measuring structure similarity, we use three existing structural alignment methods: Dali (Holm and Sander, 1993), Vast (Madej *et al*., 1995) and CE (Shindyalov and Bourne, 1998), which identify the homologs of a given structure. Each sequence- and structure-comparison method described above assigns a score for a pair of proteins, that indicates the statistical significance of the similarity between them. In particular, we have used the *Z*-scores reported by CE and Dali, *p*-values reported by Vast, and *E*-values ($-\log(E\text{-value})$) reported by HMM and PSI-Blast as similarity scores.

Next, we discuss how random walks on similarity networks are performed.

2.1 Random walks on similarity networks

Random walks provide an efficient mechanism for capturing the global structure of a similarity network. The idea is to start a random walker at a specified start node q, and at every time step to advance the walker to a random neighboring node (based on the weights of the adjacent edges) or to return to q with a restart probability c. The result of this process can be captured in an affinity vector $\mathbf{x_q}$ that measures the affinity (or the steady state residence probability) of the nodes in the graph to the start node q (Lovasz, 1996;

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

Input: similarity network $G = (V, E)$;
query node q;
restart probability c;
Output: affinity vector $\mathbf{x_q}$;

Let $\mathbf{s_q}$ be the restart vector with 0 for all its entries except a 1 for the entry denoted by node q;
Let $\mathbf{P}$ be the column normalized adjacency (transition) matrix defined by G;
Initialize $\mathbf{x_q} := \mathbf{s_q}$;
while ($\mathbf{x_q}$ has not converged)
$\mathbf{x_q} := (1-c)\mathbf{P}\mathbf{x_q} + c\mathbf{s_q}$;

Fig. 1. Iterative algorithm for computing the affinity vector $\mathbf{x_q}$.

Pan *et al*., 2004). Formally, the affinity of a node v to node q, x_{qv}, is defined as:

DEFINITION 2.1. *x_{qv} is the steady state probability that a random walk starting at node q visits node v.*

The restart probability c enforces a restriction on how far we want the random walker to get away from the start node q. In other words, if c is close to 1, the affinity vector reflects the local structure around q, and as c gets close to 0, a more global view is observed.

The steady state affinity vector can be computed efficiently by iterative matrix multiplication. Figure 1 shows the algorithm. A proof of convergence of the above algorithm follows from existing literature (Bolch *et al*., 1998; Weston *et al*., 2004) and is also given in an accompanying supplement. The number of iterations for convergence is closely related to the restart probability c. As c gets smaller, the diameter of the observed neighborhood increases, thus the number of iterations to converge gets larger. The convergence criterion is that the L_1-norm between two consecutive $\mathbf{x_q}$s is less than a small threshold, e.g. 10^{-9}. In our experiments, for $c = 0.30$, the average number of iterations to converge is $\sim$55.

2.2 Integration of known classifications

The information of known category associations can be integrated into an existing similarity network of proteins for classification purposes. In this paper, we integrate the SCOP (Structural Classification of Proteins) classification (Murzin *et al*., 1995) into the similarity network. A node for each SCOP category is added to the similarity network, enhancing the set of nodes V as $V = V_{\text{protein}} \cup V_{\text{family}} \cup V_{\text{superfamily}} \cup V_{\text{fold}}$. A detailed description of this category extension process is given in the accompanying Supplementary Material.

To classify a new protein q, we compute the affinity of category nodes to q for each level of the taxonomy, namely the $\mathbf{x_q}(V_{\text{family}})$, $\mathbf{x_q}(V_{\text{superfamily}})$ and $\mathbf{x_q}(V_{\text{fold}})$ vectors. Each similarity network creates its own set of affinity vectors, and affinity vectors from different graphs are merged using a Bayesian approach which is discussed next.

3 AUTOMATED CLASSIFICATION USING SIMILARITY NETWORKS

In this section, we propose a Bayesian framework that uses multi-attribute similarity networks for automated classification. We choose to use a Bayesian classifier (Mitchell, 1997) for a number of reasons: it supports multi-class classifications, it can use the initial distribution of data to minimize training, it inherently supports multiple sources of information, it has minimal number of parameters, and it provides probabilistic decisions. The affinity vectors computed from the similarity network are used as values of random variables that represent category affinities. The similarity network of training proteins is used to compute prior probabilities and Bayesian correction is applied to the cases without adequate information. A Bayesian classifier is then constructed and used to classify query proteins at a given taxonomy level.

Figure 2 shows an overall view of the process of merging affinity vectors using the Bayesian approach. In the remainder of this section, we present the theoretical details of our Bayesian model for automated classification and show how the model is trained using known taxonomies such as SCOP (Murzin *et al*., 1995) and CATH (Orengo *et al*., 1997).

3.1 Bayesian network model

A Bayesian network graphically represents the conditional dependencies between the components of a probabilistic model. The joint probability of the model is then computed in a systematic way by using the conditional probabilities. The Bayesian model we propose in this section uses the affinities computed on the similarity network as values of the random variables. The random variables in the network presented in Figure 3 are as follows:

- Q: category of the query protein,
- H_i: the category with the highest affinity to the query protein based on similarity measure corresponding to tool T_i, i.e. $\texttt{arg max}\ \mathbf{x_q}(V)_i$ and
- A_i: affinity of category H_i for the query protein, i.e. $\texttt{max}\ \mathbf{x_q}(V)_i$.

As shown in Figure 3, we assume that data sources are conditionally independent from each other given the class of the query protein.

The category of the query protein depends on the categories with the highest affinities and their affinity values (Fig. 3). The probability that a query protein belongs to a category c_j given a set of m similarity measures (maximal affinity categories and their affinities) is written as follows. (Terms beginning with an upper case letter represent variables and the terms beginning with a lower case letter represent instances.)

$$P(Q = c_j \mid \ldots, H_i = h_i, A_i = a_i, \ldots). \tag{1}$$

We simplify the computation of the Equation (1) by assuming the independence of (H_i, A_i) sets of variables given a specific category c_j (standard naive Bayesian assumption):

$$\begin{aligned}
&P(Q = c_j \mid \ldots H_i = h_i, A_i = a_i \ldots) \\
&\quad = \frac{P(\ldots H_i = h_i, A_i = a_i \ldots \mid Q = c_j) P(Q = c_j)}{P(\ldots H_i = h_i, A_i = a_i \ldots)} \\
&\quad = \frac{P(Q = c_j)}{P(\ldots H_i = h_i, A_i = a_i \ldots)} \prod_{k=1}^{m} P(H_k = h_k, A_k = a_k \mid Q = c_j) \\
&\quad = \frac{P(Q = c_j)}{P(\ldots H_i = h_i, A_i = a_i \ldots)} \\
&\qquad \times \prod_{k=1}^{m} \frac{P(Q = c_j \mid H_k = h_k, A_k = a_k) P(H_k = h_k, A_k = a_k)}{P(Q = c_j)}
\end{aligned}$$

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

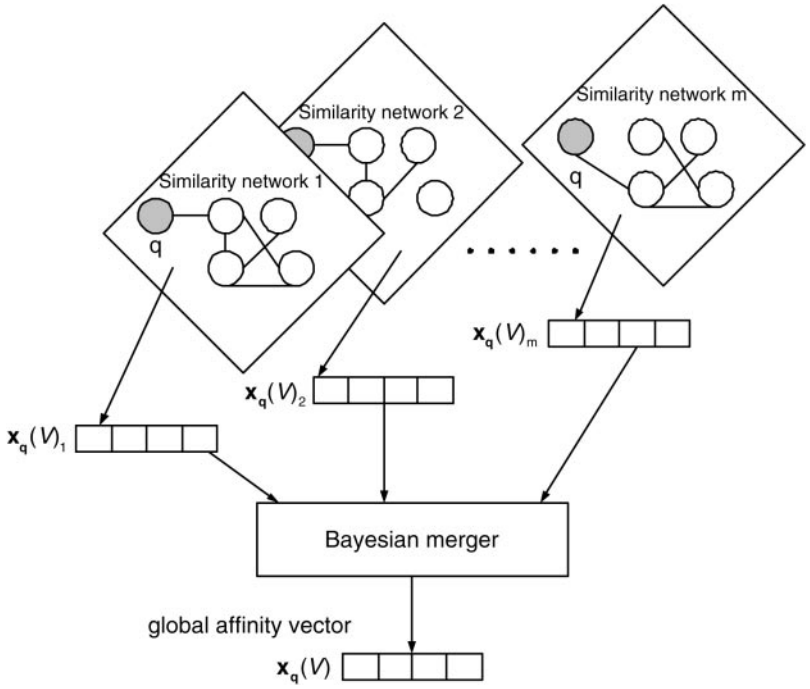


Fig. 2. Merging affinity vectors using a Bayesian approach.

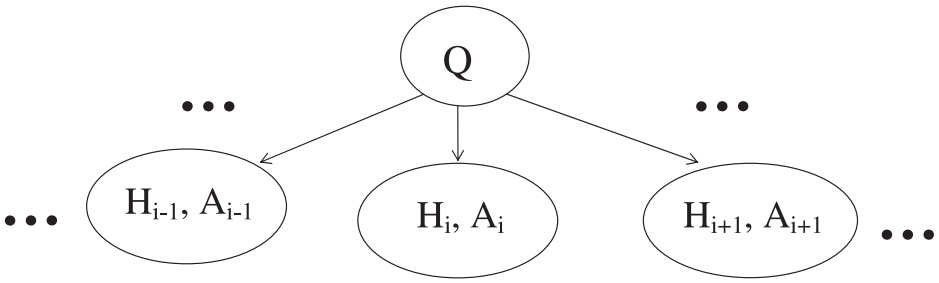


Fig. 3. Bayesian network of classifier.

$$= \frac{\prod_{k=1}^{m} P(H_k = h_k, A_k = a_k)}{P(\ldots H_i = h_i, A_i = a_i \ldots)} \times P(Q = c_j) \prod_{k=1}^{m} \frac{P(Q = c_j \,|\, H_k = h_k, A_k = a_k)}{P(Q = c_j)}. \quad (2)$$

Since the first term is a constant over all categories, the classification can be carried out knowing the values of $P(Q = c_j)$ and $P(Q = c_j | H_k = h_k, A_k = a_k)$. The exact probabilities, if needed, can be computed by normalization. Our decision to use the highest affinity from a similarity measure instead of the entire affinity vector simplifies the training task as detailed next. Note that we use a non-standard parameterization in Equation (3) instead of a standard parameterization [Equation (2)]. The non-standard approach provides a more reliable global prior as described next.

3.2 Training of Bayesian classifier

Training of the Bayesian classifier involves computation of the prior and posterior probabilities. These probabilities are computed based on the similarity network of training proteins (datasets are described in detail in Section 4). The prior probabilities, $P(Q = c_j)$, are computed based on the distribution of the training proteins over the different categories.

For the computation of the posterior probabilities, $P(Q = c_j | H_i = h_i, A_i = a_i)$, we can use a histogram-based estimation technique. The affinities are discretized into bins and the posterior probabilities are computed for each category and bin. However, this approach fails for small training sets. For example, for 1308 families in SCOP, if 10 bins are used, there would be 17 108 640 $P(Q = c_j | H_i = h_i, A_i = a_i)$ terms. Since there are only 32 149 protein structures in PDB and 2 131 063 protein sequences in UniProt (as of August 2005), there will not be adequate training data for many of the terms. To address this issue, we use Bayesian correction (Jensen, 2001): a global prior is computed over all the bins, and later this is modified by the training data and normalized. For each (h_i, a_i) pair, we compute a histogram whose bins range over the categories. Each bin c_j in the histogram is initialized to the prior probability $P(Q = c_j)$. This value is then incremented by one for each training protein that falls into it. Thus, the value in bin c_j will finally equal the initial value plus the

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

number of training proteins for which $H_i = h_i$, $A_i = a_i$, and category $= c_j$. The histogram for the (h_i, a_i) pair is then normalized to yield the posterior probabilities $P(Q = c_j | H_i = h_i, A_i = a_i)$ for each bin.

Next, we illustrate the computation of prior and posterior probabilities using an example. Assume that there are three categories c_1, c_2, and c_3. If 20% of the training proteins belong to category c_1, 30% belong to c_2, and the remaining 50% belong to c_3, then the prior probabilities are computed as $P(Q = c_1) = 0.2$, $P(Q = c_2) = 0.3$, and $P(Q = c_3) = 0.5$. Now, consider any pair (h, a) for a similarity measure corresponding to a tool T_i. In order to compute the posterior probabilities for this pair, the bins of the histogram are initialized to (0.2, 0.3, 0.5). Suppose that two proteins in the training set belong to the pair (h, a) for tool T_i. Further, suppose that their categories are c_1 and c_2. Then, the histogram's final value is (1.2, 1.3, 0.5). After normalization, we obtain (0.4, 0.43, 0.17). This leads to the posterior probabilities $P(Q = c_1 | H_i = h, A_i = a) = 0.4$, $P(Q = c_2 | H_i = h, A_i = a) = 0.43$, and $P(Q = c_3 | H_i = h, A_i = a) = 0.17$.

Our decision to use the highest affinity from each similarity measure instead of the entire affinity vector can be justified by examining the number of terms that need to be computed during training. For example, at the family level, an affinity vector has 1308 entries. Assuming 10 bins are used, there will be 10^{1308} different combinations to evaluate during training. This renders the approach of using the entire affinity vector impractical.

4 EXPERIMENTS

We conducted a number of experiments to study the parameters that influence our approach based on similarity networks, random walks and Bayesian classification. We also validated the proposed framework by comparison with a number of alternative approaches. We used the SCOP database (Murzin *et al*., 1995) version 1.63 for validation. SCOP is a hierarchical classification of protein structures and is used as a benchmark in many studies. We removed redundant sequences from this set using the ASTRAL compendium for sequence and structure analysis (Chandonia *et al*., 2004). We selected protein domains with <95% sequence identity, reducing the set to 8720 domains. This was further restricted to a dataset of size 4087 by considering only single domain protein chains for which pre-computed sequence/structure similarity measures are provided. Of these proteins, 3315 are members of the four major SCOP classes: all α, all β, $\alpha + \beta$ and α/β. We performed 3-fold cross validation on this set. In other words, we partition 3315 chains into three sets. We use two of these partitions (2210 chains), in turn, for training, and the remaining partition (1105 chains) for testing. Note that no training protein is used for testing. The accuracy values reported are the average values. We name the training sets DS and the query sets QS.

We created a multi-attribute similarity network on *DS* using the sequence/structure similarity measures described in Section 2. The resulting graph contains ~3700 nodes on the average for the three training datasets ($|V_{\text{protein}}| = 2210$, $|V_{\text{family}}| \simeq 745$, $|V_{\text{superfamily}}| \simeq 445$ and $|V_{\text{fold}}| \simeq 305$) and ~40 000 edges per similarity measure for the score threshold approach. We then conducted random walks for each protein q in *QS* by inserting them (one at a time) into the similarity network via edges that represent sequence/structure relationships to the proteins in DS. Thus, we computed $\mathbf{x_q}(V_{\text{family}})$, $\mathbf{x_q}(V_{\text{superfamily}})$ and $\mathbf{x_q}(V_{\text{fold}})$ vectors for each similarity measure. We employed the strategy proposed by Lindahl and Elofsson (2000) to have a complete separation of different levels in the classification: we discarded edges (i.e. similarities) to family members of q when computing $\mathbf{x_q}(V_{\text{superfamily}})$ and ignored both family/superfamily level edges when computing $\mathbf{x_q}(V_{\text{fold}})$. The classification is then carried out by the Bayesian classifier trained on DS as described in Section 3.2. In our experiments, we tested the classification performance using the SCOP classification as the ground truth. The performance is measured by the percentage of correct classifications. As for the running time, the classification of a query protein is performed in the order of seconds on a typical desktop computer.

The first set of experiments analyzes the parameters that affect our approach. The second set compares the proposed framework with alternative approaches. Additional experimental results that assess the certainty of classification assignments are given in the accompanying supplement.

4.1 Analysis of parameters

To study the effect of restart probability (the only parameter of the iterative random walk algorithm), we conducted experiments on a single attribute network built using VAST similarity scores. Figure 4 shows the effect of changing the restart probability on the percentage of correct classifications. The category of the query protein is given by the category of its nearest neighbor, i.e. 1-NN classification. We see that decreasing c (i.e. increasing the diameter of random walks) decreases the family level performance a little bit, because of added noise by other families. On the whole, the results are not very sensitive to the choice of c. We adopt $c = 0.30$ in the rest of the experimental evaluation.

The purpose of the next set of experiments is to analyze the effect of the number of neighbors, k, in the *top-k* approach to network construction. The choice of k changes the structure of a similarity network by adding different number of edges. Theoretically, an increase in k captures distant similarities. But, beyond a certain point, it forces noise into the affinity vectors, thus having a negative effect on the ability of the network to capture the protein space. Figure 5 shows the classification performance of Bayesian classifiers that are based on graphs built with varying number of closest neighbors. The graph validates our theory about the effect of k on the performance at different levels. Classification performance at the family level decreases rapidly when the number of top k neighbors used increases. At the superfamily and fold levels, though, the performance does not change as much with respect to the number of neighbors considered. We choose $k = 2$ in the rest of the experiments.

Next, we investigated how the classifier performance is affected by using score thresholds instead of *top-k* in similarity network construction. The classification results for classifiers built using the top-2 approach and score thresholding on single-attribute and multi-attribute similarity networks are displayed in Table 1. An integrated classifier that uses multi-attribute similarity networks performs better than the classifiers that use single-attribute similarity networks regardless of the graph creation methodology. Additionally, top-2 approach is more successful than the score thresholding approach. The reason behind this is the nonuniformity of the scores assigned by the comparison methods. A match between two small proteins is statistically less significant than a match between two large proteins, resulting in a higher score for the latter pair. Experimentally, we observed that for a fixed

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

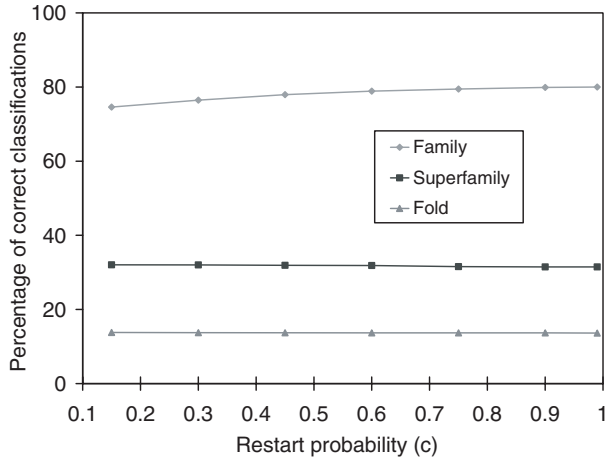


Fig. 4. The effect of changing the restart probability on classification using VAST similarity measure. The query protein is assigned to the category of the highest affinity protein.

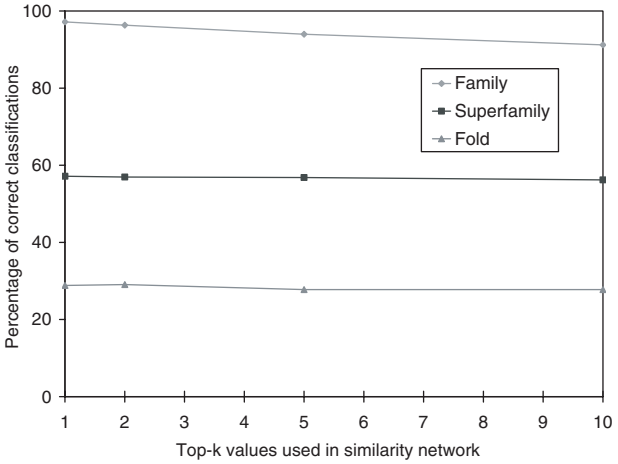


Fig. 5. Performance analysis of classifiers on graphs built on varying number of *top-k* neighbors.

Table 1. The percentage of correct classifications on similarity networks based on score thresholds and top-2 neighbors

	Family		Superfamily		Fold	
	Score	Top-2	Score	Top-2	Score	Top-2
VAST	76.69	91.20	33.02	46.78	13.96	24.23
DALI	81.80	92.63	47.33	51.37	16.92	22.66
CE	70.39	89.69	32.75	45.98	9.83	20.91
HMM	81.88	90.25	41.96	40.94	8.17	8.20
BLAST	86.18	89.58	12.75	13.41	2.96	3.06
Integrated	92.81	96.32	54.24	56.94	22.66	29.06

score threshold, large proteins tend to have hundreds of neighbors, whereas small proteins have only a few. Because of this inherent property of similarity scores, a score threshold is not suitable for defining the similarity cutoff. The performance improves significantly for classifiers based on single-attribute similarity networks for top-2 compared with score thresholding. For the integrated classifier, though, the change in the performance is lower. This indicates that the non-uniformity originating from the similarity networks is corrected when multiple sources are merged to make a combined decision. Combining similarity networks using multiple similarity measures seems to make the final decisions more robust.

4.2 Comparative analysis

To validate the effectiveness of the similarity networks, we analyzed the performance of Bayesian classifiers built on these networks and compared our results with a kernel based support vector machine (SVM) classifier, a nearest neighbor classifier, and an integrated sum classifier.

In the first technique, we built an integrated kernel by combining individual kernels representing each similarity measure. The kernel for a similarity measure is created using a technique similar to the SVM-Pairwise (Liao and Noble, 2003) method, in which a protein is represented as a vector of similarity scores, and the kernel between two proteins is defined using the Gaussian kernel with the optimum parameters found using 5-fold cross validation. Using this method, we created kernels for the sequence and structure similarity measures. We then normalized all kernels to 1 on the diagonal and centered in the feature space. There are a number of ways of integrating the individual kernels. Lanckriet *et al*. (2004) proposed a weighted sum approach to combine kernels where weights are determined using semi-definite programming. However, a uniform sum approach has been shown to produce satisfactory results (Pavlidis *et al*., 2001; Yamanishi *et al*., 2004), and we employ that approach in our experiments. We use the LIBSVM library (Chang and Lin, 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm) to train a multi-class soft margin support vector machine (C-SVM). We determined the best value for the penalty parameter in the error term using 5-fold cross validation. The multi-class strategy implemented by LIBSVM employs a one-against-one strategy, in which $k \cdot (k-1)/2$ classifiers are trained for k classes. The accuracy results we report in the paper are the results from average of multiple one-against-one SVMs. Alternatively, we could have used a one-against-all classification strategy for testing, as employed by previous work on protein classification using SVMs (Liao and Noble, 2003; Saigo *et al*., 2004). This strategy is also implemented in LIBSVM. Our experiments using one-against-all classification did not produce better results than the one-against-one strategy.

We also compared our technique with two other simpler integration and classification schemes:

- 1NN classifier: Here, we directly use the similarity scores of the first nearest neighbors (1NN) of query proteins as determined by each similarity measure. A query protein is then classified into the category indicated by the combined 1NN scores. Instead of a vector of random walk affinities, we now have a vector of similarity scores. In order to merge the similarity measures, we train a Bayesian classifier as discussed in Section 3.
- Sum classifier: Another simple scheme is to merge the individual results by summing the affinity vectors (as opposed to a Bayesian classifier). Then, the protein is assigned to the category that has the highest sum.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

Table 2. Performance of Bayesian classifiers based on various similarity networks (SN) compared with 1NN classifier, sum classifier and SVM classifier

	Family	Superfamily	Fold
SVM classifier	91.24	49.33	22.87
1NN classifier	89.47	46.78	19.12
Sum classifier	96.77	53.21	23.94
SN without category nodes	62.71	40.43	17.91
SN with score thresholding	92.81	54.24	22.66
SN with top-1	97.16	57.13	28.84
SN with top-2	96.31	56.94	29.06

The values represent the percentage of correct classifications. SN with Bayesian integration clearly outperform the competing techniques.

The comparison of Bayesian classifiers based on similarity networks and competing techniques is depicted in Table 2. This table clearly indicates that our integrated Bayesian approach outperforms all the competing techniques by achieving correct classification accuracies of 96.31–29.06% from family to fold levels. On the other hand, SVM classifier correctly classified 91.24, 49.33 and 22.87% of the proteins at the family, superfamily and fold levels respectively. The poor performance of the 1NN classifier compared with our integrated approach shows that the use of global similarity information is superior to naive, local ranking methods. Similarly, the lower accuracy of the sum classifier establishes that using a naive integration approach is not as effective as our Bayesian model. Also, the poor performance of the similarity network without category nodes approach signifies the benefit of a category node enhanced network for classification purposes. We conclude from these results that multi-attribute similarity networks are effective at capturing close and remote similarities, and that Bayesian networks provide a sound integration technique.

5 DISCUSSION

We proposed a novel framework for analysis of multi-attribute similarity networks based on random walks on graphs and Bayesian theory. Our framework is general enough to be applicable in different biological applications such as structural classification, functional classification, automated annotation and clustering. We showed that by using multiple similarity networks based on different similarity measures, one can obtain a better representation of the protein space and can make more accurate decisions. Moreover, our experiments showed that the proposed technique is more effective in capturing the properties of the protein space compared with a competing kernel-based information integration approach.

The similarity network and the classification scheme we propose are not limited to only sequence and structure sources, but are general enough to be incorporated with a wide range of information sources. The only constraint on the source is that it should be able to give a similarity measure between a pair of proteins. One can even use motif databases such as PROSITE (Sigrist *et al*., 2002) and Pfam (Bateman *et al*., 2004) as information sources (Kuang *et al*., 2005). In this case, the similarity between a pair of proteins is defined as the similarity of the sets of motifs each protein contains. Using a similar approach, pathway data and gene expression data can also be incorporated into our framework. As the diversity of information sources increases in the future, integrated approaches have the potential of capturing the protein space even more accurately.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. and Koonin,E.V. (1998) Iterated profile searches with PSI-BLAST–a tool for discovery in protein databases. *Trends Biochem. Sci*., **23**, 444–447.

Bateman,A. *et al*. (2004) The Pfam protein families database. *Nucleic Acids Res*., **32**, D138–D141.

Berman,H.M. *et al*. (2000) The protein data bank. *Nucleic Acids Res*., **28**, 235–242.

Bolch,G., Grener,S., deMeer,H. and Shridhar bhai Trivedia,K. (1998) *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley-Interscience.

Chandonia,J.M *et al*. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res*., **32**, D189–D192.

Chang,C.C. and Lin,C.J. (2001) *LIBSVM: a library for support vector machines*.

Deng,M. *et al*. (2004) An integrated probabilistic model for functional prediction of proteins. *J. Comput. Biol*., **11**, 463–475.

Eddy,S.R. (1998) Profile hidden markov models. *Bioinformatics*, **14**, 755–763.

Gough,J. (2002) The SUPERFAMILY database in structural genomics. *Acta Crystallogr. D. Biol. Crystallogr*., **58**, 1897–1900.

Haussler,D. (1999) Convolution Kernels on Discrete Structures. *Technical Report UCSC-CLR-99-10*. University of California, Santa Cruz.

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol*., **233**, 123–138.

Holm,L. and Sander,C. (1996) Mapping the protein universe. *Science*, **273**, 595–602.

Hou,J. *et al*. (2003) A global representation of the protein fold space. *Proc. Natl Acad. Sci. USA*, **100**, 2386–2390.

Jensen,F.V. (2001) *Bayesian networks and decision graphs*, Springer.

Kuang,R. *et al*. (2005) Motif-based protein ranking by network propagation. *Bioinformatics*, **21**, 3711–3718.

Lanckriet,G.R. *et al*. (2004) Kernel-based data fusion and its application to protein function prediction in yeast. *Pac. Symp. Biocomput.,* 300–311.

Liao,L. and Noble,W.S. (2003) Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *J. Comput. Biol*., **10**, 857–868.

Lindahl,E. and Elofsson,A. (2000) Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol*., **295**, 613–625.

Liu,J. and Rost,B. (2003) Domains, motifs and clusters in the protein universe. *Curr. Opin. Chem. Biol*., **7**, 5–11.

Lodhi,H., Shawe-Taylor,J., Cristianini,N. and Watkins,C. (2000) Text classification using string kernels. In *Proceedings of Neural Information Processing Systems,* pp. 563–569.

Lovasz,L. (1996) Random walks on graphs: a survey. *Combinatorics, Paul Erdos is Eighty*, vol. **2**, pp. 353–398.

Madej,T. *et al*. (1995) Threading a database of protein cores. *Proteins*, **23**, 356–369.

Mitchell,T. (1997) *Machine Learning*, McGraw Hill.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*., **247**, 536–540.

Orengo,C.A. *et al*. (1997) CATH—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.

Pan,J.Y. Duygulu,P., Falontsos,C. and Hyung-Jeong,Y. (2004) GCap: graph-based automatic image captioning. In *Proceedings of Computer Vision and Pattern Recognition Workshop,* vol. **9**, pp. 146–154.

Pandit,S.B. *et al*. (2002) SUPFAM—a database of potential protein superfamily relationships derived by comparing sequence-based and structure-based families: implications for structural genomics and function annotation in genomes. *Nucleic Acids Res*., **30**, 289–293.

Pavlidis,P., Weston,J., Cai,J. and Grundy,W.N. (2001) Gene functional classification from heteregeneous data. In *Proceedings of Research in Computational Biology,* pp. 249–255.

Portugaly,E. *et al*. (2002) Selecting targets for structural determination by navigating in a graph of protein families. *Bioinformatics*, **18**, 899–907.

Saigo,H. *et al*. (2004) Protein homology detection using string alignment kernels. *Bioinformatics*, **20**, 1682–1689.

Schoelkopf,B., Tsuda,K. and Vert,J.-P. , (eds) (2004) *Kernel methods in computational biology*. MIT Press.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020

Shindyalov,I. and Bourne,P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*., **11**, 739–747.

Sigrist,C.J.A. *et al*. (2002) PROSITE: a documented database using patterns and profiles as motif descriptors. *Brief Bioinform*., **3**, 265–274.

Szummer,M. and Jaakkola,T. (2001) Partially labeled classification with markov random walks. In *Proceedings of Neural Information Processing Systems,* pp. 945–952.

Weston,J. *et al.* (2004) Protein ranking: from local to global structure in the protein similarity network. *Proc. Natl Acad. Sci. USA*, **101**, 6559–6563.

Yamanishi,Y. *et al*. (2004) Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, **20** (Suppl. 1), i363–i370.

Yona,G. and Levitt,M. (2000) Towards a complete map of the protein space based on a unified sequence and structure analysis of all known proteins. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.,* **8,** 395–406.

Yona,G. *et al*. (1999) ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins*, **37**, 360–378.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/22/13/1585/193368 by Middle East Technical University Library user on 23 July 2020