Detection of Gene Orthology Based On Protein-Protein Interaction Networks

Fadi Towfic, M. Heather West Greenlee, Vasant Honavar
Bioinformatics and Computational Biology Graduate Program, Iowa State University, Ames, IA
{ftowfic,mheather,honavar}@iastate.edu

***Abstract*—Ortholog detection methods present a powerful approach for finding genes that participate in similar biological processes across different organisms, extending our understanding of interactions between genes across different pathways, and understanding the evolution of gene families. We exploit features derived from the alignment of protein-protein interaction networks to reconstruct KEGG orthologs for *Drosophila melanogaster*, *Saccharomyces cerevisiae*, *Mus musculus* and *Homo sapiens* protein-protein interaction networks extracted from the DIP repository for protein-protein interaction data using the decision tree, Naive-Bayes and Support Vector Machine classification algorithms. The performance of our classifiers in reconstructing KEGG orthologs is compared against a basic reciprocal BLAST hit approach. We provide implementations of the resulting algorithms as part of BiNA, an open source biomolecular network alignment toolkit.**

I. Introduction

With the advent of fast and relatively inexpensive sequencing technology, it has become possible to access and compare genomes from a wide range of organisms including many eukaryotes as well as bacteria and archea through databases such as GenBank [4], Ensembl [17], PlantGDB [13] and others [9], [8], [5]. The availability of genomes from such a wide range of organisms has enabled the comparison and analysis of evolutionary relationships among genes across organisms through the reconstruction of phylogenies [39], common pathways [23], [27], and comparing gene functions [33], [15]. Of particular interest in this context is the problem of finding genes originating from a single gene from a common ancestor of the compared genomes (orthologs) [25]. Ortholog detection methods present a powerful approach for finding genes that participate in similar biological processes across different organisms, extending our understanding of interactions between genes across different pathways, and understanding the evolution of gene families.

Several sequence-based approaches currently exist for finding orthologous genes among a set of genomes. For instance, one of the simplest methods is to utilize reciprocal best BLAST hits [1] across a set of species to identify orthologs [21]. The COGs (Clusters of Orthologous Groups) approach [36], for example, defines orthologs as sets of proteins that are reciprocal best BLAST hits across a minimum of three species. Another possible approach utilized by databases such as InParanoid [30] and OrthoMCL [26] consists of an iterative BLAST search to construct the reciprocal BLAST hits, and a second step that clusters the reciprocal hits to achieve greater sensitivity. InParanoid uses a pre-defined set of rules to construct its clusters, while OrthoMCL utilizes a sequence-based Markov clustering algorithm for clustering its proteins/genes into ortholog groups. Other approaches, such as PhyOP [19], RAP [14] and others [33], [23], [39], [15] identify orthologous genes/proteins by utilizing phylogenetic analysis to explicitly exploit the evolutionary rates across the species being compared. Such approaches account for the different mutation rates accumulated by the various species being compared, thus allowing greater sensitivity in detecting the pairs of genes/proteins to be classified as orthologous. Methods such as those utilized by Fu et al. consider gene order and rearrangements in detecting orthologs [18]. Recently, with the availability of large-scale analysis of protein-protein interactions, protein-protein interaction networks have also been considered in detecting orthologous genes. Ogata et al. utilized a graph comparison algorithm to compare protein-protein interaction networks and determined orthologs by matching the nodes in the protein-protein interaction graphs [31]. Bandyopadhyay et al. utilized the PathBLAST pathway alignment algorithm to detect orthologs [3]. Another method utilized by databases such as KEGG is to manually construct orthology groups based on a combination of features such as sequence similarity, pathway interactions, and phylogenetic analysis [27], [23].

Against this background, we explore a set of graph features that may be utilized in detecting orthologs based on sequence similarity as well as the similarity of their neighborhoods in protein-protein interaction networks. Furthermore, we construct a set of classifiers that utilize the above features and compare the classifiers to the reciprocal BLAST hits approached for the reconstruction of KEGG orthologs [23]. The basic idea behind our approach is to align a pair of protein-protein interaction networks and scan the alignment for all possible matches that a node (protein) from one network can pair with in the other network. We then train decision tree [42], Naive-Bayes [29], Support Vector Machine [10], and an ensemble classifier [12] that utilize features from the alignment algorithm to identify KEGG orthologs and we compare the performance of the classifiers to the reciprocal BLAST hit method.

We utilize the alignment algorithms available as part of the BiNA (Biomolecular Network Alignment) toolkit [38] as well as graph features extracted from the aligned protein-protein interaction networks such as degree distribution, BaryCenter [41], betweenness [40] and HITS (Hubs and Authorities) [24] centrality measures. Our experiments with the fly, yeast, mouse and human protein-protein interaction networks extracted from DIP (Database of Interacting Proteins) [34] demonstrate the feasibility of the proposed approach for detecting KEGG

978-0-7695-3885-3/09 $26.00 © 2009 IEEE
DOI 10.1109/BIBM.2009.85

orthologs.
The rest of the paper is organized as follows: Section 2 introduces the dataset and methods for aligning two biomolecular networks and describes our approach for exploiting the neighborhood similarity measures for detecting orthology. Section 3 describes the experimental setup and experimental results. Section 4 concludes with a summary of the main contributions of the paper in the broader context of related literature and a brief outline of some directions for further research.

II. Materials and Methods

A. Dataset

The yeast, fly, mouse and human protein-protein interaction networks were obtained from the Database of Interacting Proteins (DIP) release 1/26/2009 [34]. The sequences for each dataset were obtained from uniprot release 14 [2]. The DIP sequence ids were matched against their uniprot counterparts using a mapping table provided on the DIP website. All proteins from DIP that had obsolete uniprot IDs or were otherwise not available in release 14 of the uniprot database were removed from the dataset. The fly, yeast, mouse and human protein-protein interaction networks consisted of $6,645$, $4,953$, 424 and $1,321$ nodes and $20,010$, $17,590$, 384 and $1,716$ edges, respectively. The protein sequences for each dataset were downloaded from uniprot [2]. BLASTp [1] with a cutoff of 1×10^{-10} was used to match protein sequences across species. The KEGG (Kyoto Encyclopedia of Genes and Genomes) [23] orthology and uniprot annotations for all species were downloaded from the KEGG website and matched against the uniprot id's for the proteins in the datasets.

B. Graph Representation of BLAST Orthologs

The proteins in the DIP protein-protein interaction networks for mouse, human, yeast, and fly were matched using BLAST as shown in figure 1. As can be seen from the figure, protein-protein interaction networks are represented as two labeled graphs (graphs 1 and 2) with weighted edges connecting sequence-homologous nodes across the two graphs. The BLAST similarity scores are taken into account when comparing the neighborhoods around each of the vertices in the graphs to reconstruct the KEGG orthologs. This graph representation is similar to the representations used by NetworkBLAST [22], HopeMap [37], and Graemlin 2.0 [16]. A k-hop neighborhood-based approach to alignment uses the notion of k-hop neighborhood. The k-hop neighborhood of a vertex $v_x^1 \in V_1$ of the graph $G_1(V_1, E_1)$ is simply a subgraph of G_1 that connects v_x^1 with the vertices in V_1 that are reachable in k hops from v_x^1 using the edges in E_1. Given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, a mapping matrix $\mathbf{P}$ that associates each vertex in V_1 with zero or more vertices in V_2 (the matrix $\mathbf{P}$ can be constructed based on BLAST matches) and a user-specified parameter k, we construct for each vertex $v_x^1 \in V_1$ its corresponding k-hop neighborhood C_x in G_1. We then use the mapping matrix $\mathbf{P}$ to obtain the set of matches for vertex v_x^1 among the vertices in V_2; and construct the k-hop neighborhood Z_y for each matching vertex v_y^2 in G_2 and $P_{v_x^1 v_y^2} = 1$. Let $S(v_x^1, G_2)$ be the resulting collection of k-hop neighborhoods in G_2 associated with the vertex v_x^1 in G_1. We compare each k-hop subgraph C_x in G_1 with each member of the corresponding collection $S(v_x^1, G_2)$ to identify the k-hop subgraph of G_2 that is the best match for C_x (based on a chosen similarity measure). Figure 1 illustrates this process.

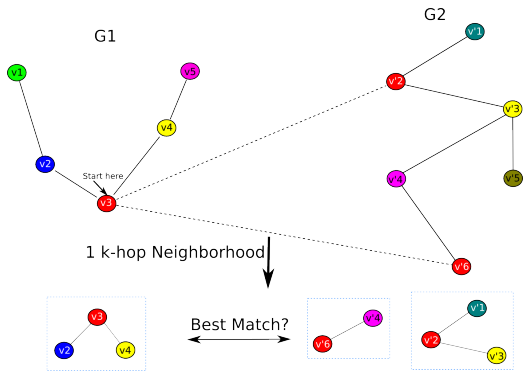

Figure 1. A schematic of the graph representation of the BLAST orthologs based on the DIP protein-protein interaction networks. The protein-protein interaction networks are represented as two labeled graphs (G1 and G2) with corresponding relationships among their nodes (similarly colored nodes are sequence homologous according to a BLAST search). Nodes from G1 (e.g., v3) are compared to their sequence-homologous counterparts in G2 (e.g., v'2 and v'6) based on the topology of their neighborhood and sequence homology of the neighbors. In the figure, v'2 has the same number of neighbors of v3 and one of the neighbors of v'2 (i.e., v'3) is sequence-homologous to v4. Thus, v'2 is scored higher (more likely to be an ortholog to v3) compared to v'6.

C. Shortest Path Graph Kernel Score

The shortest path graph kernel was first described by Borgwardt and Kriegel [6]. As the name implies, the kernel compares the length of the shortest paths between any two nodes in a graph based on a pre-computed shortest-path distance. The shortest path distances for each graph may be computed using the Floyd-Warshall algorithm as implemented in the CDK (Chemistry Development Kit) package [35]. We modified the Shortest-Path Graph Kernel to take into account the sequence homology of nodes being compared as computed by BLAST [1]. The shortest path graph kernel for subgraphs Z_{G_1} and Z_{G_2} (e.g., k-hop subgraphs, bicomponent clusters extracted from G_1 and G_2 respectively) is given by:

$$S = \sum_{v_i^1, v_j^1 \in Z_{G_1}} \sum_{v_k^2, v_p^2 \in Z_{G_2}} \delta(v_i^1, v_k^2) \times \delta(v_j^1, v_p^2) \times d(v_i^1, v_j^1) \times d(v_k^2, v_p^2)$$
$$K(Z_{G_1}, Z_{G_2}) = \log[S]$$

where $\delta(v_x^1, v_y^2) = \frac{BlastScore(v_x^1, v_y^2) + BlastScore(v_y^2, v_x^1)}{2}$. $d(v_i^1, v_j^1)$ and $d(v_k^2, v_p^2)$ are the lengths of the shortest paths between v_i^1,v_j^1 and v_k^2,v_p^2 computed by the Floyd-Warshall algorithm. The runtime of the Floyd-Warshall Algorithm is $O(n^3)$. The shortest path graph kernel has a runtime of $O(n^4)$ (where n is the maximum number of nodes in larger of the two graphs being compared). Please see figure 2 for a general outline of the comparison technique used by the shortest-path graph kernel.

D. Random Walk Graph Kernel Score

The random walk graph kernel [7] has been previously utilized by Borgwardt et al. [7] to compare protein-protein

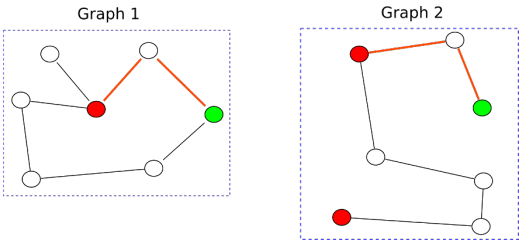

Figure 2. An example of the graph matching conducted by the shortest path graph kernel. Similarly colored nodes are sequence homologous according to a BLAST search. As can be seen from the figure, the graph kernel compares the lengths of the shortest paths around homologous vertices across the two graphs. The red edges show the matching shortest path in both graphs as computed by the graph kernel. The shortest path distance graph kernel takes into account the sequence homology score for the matching vertices across the two graphs as well as the distances between the two matched vertices within the graphs.

interaction networks. The random walk graph kernel for subgraphs Z_{G_1} and Z_{G_2} (e.g., k-hop subgraphs, bicomponent clusters extracted from G_1 and G_2 respectively) is given by:

$$K(Z_{G_1}, Z_{G_2}) = p \times (\mathbf{I} - \lambda K_x)^{-1} \times q \quad (1)$$

where $\mathbf{I}$ is the identity matrix, λ is a user-specified variable controlling the length of the random walks (a value of 0.01 was used for the experiments in this paper), K_x is an $nm \times nm$ matrix (where n is the number of vertices in Z_{G_1} and m is the number of vertices in Z_{G_2} resulting from the Kronecker product $K_x = Z_{G_1} \otimes Z_{G_2}$, specifically,

$$K_{\alpha\beta} = \delta(Z_{G_{1_{ij}}}, Z_{G_{2_{kl}}}), \alpha \equiv m(i-1)+k, \beta \equiv m(j-1)+l \quad (2)$$

Where $\delta(Z_{G_{1_{ij}}}, Z_{G_{2_{kl}}}) = \frac{BlastScore(Z_{G_{1_{ij}}}, Z_{G_{2_{kl}}}) + BlastScore(Z_{G_{2_{kl}}}, Z_{G_{1_{ij}}})}{2}$; p and q are $1 \times nm$ and $nm \times 1$ vectors used to obtain the sum of all the entries of the inverse expression $((\mathbf{I} - \lambda K_x)^{-1})$.

We adapted the random walk graph kernel to align protein-protein interaction networks by taking advantage of the reciprocal BLAST hits (RBH) among the proteins in the networks from different species [21]. Naive implementation of our modified random-walk graph kernel, like the original random-walk graph kernel [7], has a runtime complexity of $O(r^6)$ (where $r = max(n, m)$). This is due to the fact that the product graph's adjacency matrix is $nm \times nm$, and the matrix inverse operation takes $O(h^3)$ time, where h is the number of rows in the matrix being inverted (thus, the total runtime is $O((rm)^3)$ or $O(r^6)$ where $r = max(n, m)$). However, runtime complexity of the random walk graph kernel (and hence our modified random walk graph kernel) can be improved to $O(r^3)$ by making use of the Sylvester equations as proposed by Borgwardt et al. [7]. Figure 3 illustrates the computation of the random walk graph kernel.

E. BaryCenter Score

The BaryCenter score is calculated based on the total shortest path of the node. The shortest path distances for each node in a graph is calculated and the score is assigned to the node based the sum of the lengths of all the shortest paths that pass through the node [41]. More central nodes in a connected component will have smaller overall shortest

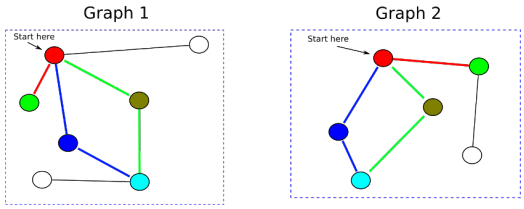

Figure 3. An example of the graph matching conducted by the random walk graph kernel. Similarly colored vertices are sequence homologous according to a BLAST search. As can be seen from the figure, the graph kernel compares the neighborhood around the starting vertices in each graph using random walks. Colored edges indicate matching random walks across the two graphs of up to length 2. The random walk graph kernel takes into account the sequence homology of the vertices visited in the random walks across the two graphs as well as the general topology of the neighborhood around the starting vertex.

paths, and 'peripheral' nodes on the network will have larger overall shortest paths.

F. Betweenness Score

Betweenness is a centrality measure of a vertex within a graph. Vertices that occur on many shortest paths between other vertices have a higher betweenness score than nodes that do not occur on many paths [40]. For a graph $G_1(V_1, E_1)$, the betweenness score for vertex $v_x^1 \in V_1$is defined as:

$$B(v_x^1) = \sum_{v_i^1 \neq v_x^1, v_j^1 \neq v_x^1, v_i^1 \neq v_j^1, v_{x,i,j}^1 \in V_1} \frac{\delta_{v_i^1 v_j^1}(v_x^1)}{\delta_{v_i^1 v_j^1}}$$

Where $\delta_{v_i^1 v_j^1}$ is the number of the shortest paths from v_i^1 to v_j^1 and $\delta_{v_i^1 v_j^1}(v_x^1)$ is the number of shortest paths from v_i^1 to v_j^1 that pass through vertex v_x^1.

G. Degree Distribution Score

A simple node importance ranker based on the degree of the node. Nodes with a high number of connections will get a high score while nodes with a smaller number of connections will receive a lower score.

H. HITS Score

The HITS score represents the "hubs-and-authorities" importance measures for each node in a graph [24]. The score is computed iteratively based on the degree connectivity of the nodes in the graph and the "authoritativeness" of the neighbors around each node. For a graph $G_1(V_1, E_1)$, each node v_x^1 is assigned two scores: $\alpha(v_x^1)$ and $\gamma(v_x^1)$. Vertices that are connected to many vertices are marked as hubs, and thus their $\alpha(v_x^1)$ scores are large. On the other hand, a vertex that points to highly connected vertices is referred to as an authority and is assigned a high $\gamma(v_x^1)$ score. Some nodes can be both highly connected (have high $\alpha(v_x^1)$ score) and most of their neighbors can also be highly connected (thus, have a high $\gamma(v_x^1)$); such nodes would have a high HITS score.

I. Classification of Orthologs Based on Sequence and Network Similarity

In order to establish orthologs between fly, yeast, human and mouse, the 1 hop and 2 hop shortest path and random walk scores, BLAST score, BaryCenter score, betweenness score,

degree distribution score and HITS score were computed for each pair of homologs detected by BLAST (total of 9 features). The BaryCenter, betweenness, degree distribution and HITS scores were combined using Milenkoviæ et al.'s [28] formula for averaging node-based scores in a graph:

$$S(u_x^1, v_y^2) = \frac{|\log(S(u_x^1)+1) - \log(S(v_y^2)+1)|}{\log(\max(S(u_x^1), S(v_y^2))+2)}$$

Where $S(u_x^1)$ and $S(v_y^1)$ are the scores for the nodes from $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, where $u_x^1 \in V_1$ and $v_y^2 \in V_2$. The above formula produces a normalized score for each node-based feature (BaryCenter, betweenness, degree distribution, and HITS scores) for each pair of homologs while adjusting for any bias in magnitude differences in the scores for the graphs (e.g, G_1 may have much more nodes than G_2, thus the node-based scores for G_1 may be more likely to be greater than the node-based scores for G_2).

J. Performance evaluation

We compare the performance of a simple method for detecting orthologs based on reciprocal BLASTp hits with the decision tree [42], Naive-Bayes [29], Support Vector Machine [10], and ensemble classifier [12] trained using the graph-based features as well as features based on network neighborhood similarity (see above) with 10-fold cross-validation. Demsar's [11] non-parametric test can be used to compare machine learning algorithms. However, because the use of this test requires the number of data sets to be greater than 10 and the number of methods to be greater than 5 [11], it cannot be applied directly to our analysis (since we have only 6 datasets and 5 methods). Hence, following Demsar's recommendation [11], we use the average ranks for each classifier based on their observed performance on datasets to compare the overall performance of different classification methods.

III. Analysis and Results

A. Reconstructing KEGG Orthologs Using BLAST

The detection of orthologs based on network alignment was recently conducted by Bandyopadhyay et al. [3]. Although Bandyopadhyay et al. showed that network-based features may be used to detect orthologs that might be missed by InParanoid, their performance was not directly compared to a reciprocal BLASTp approach. In our analysis, we compare predictions based only on the BLASTp score as well as predictions based on the network features discussed in section 2. The results in table I show the performance of the reciprocal BLAST hits method (similar to that utilized by COGs [36]) in reconstructing the orthologs between the fly, yeast, human and mouse datasets from DIP [34]. The reciprocal BLAST hit method involved finding all reciprocal hits between proteins in each dataset (e.g., mouse and human) and iterating over all possible BLAST score cutoffs to detect which of the hits are orthologous. As can be seen from the table, this method performs fairly well in reconstructing the KEGG orthologs for each dataset. As noted by Bandyopadhyay et al. [3], this may be due to the fact that most ortholog detection schemes depend on sequence homology analysis for at least part of their methods. Although KEGG orthologs rely on additional information other than sequence homology (such as metabolic pathway comparison and manual curation) [23], sequence homology may still carry a very strong basis to the detection of KEGG orthologs. Table II shows the performance of classifiers using only the BLASTp scores to detect KEGG orthologs between fly, yeast, mouse and human. The logistic regression classifier in WEKA [42] has the best performance overall (according to the average rank score shown in table II), however, it does not outperform the reciprocal BLASTp hit method shown in table I. As noted above, this may be due to KEGG's reliance on sequence homology in identifying orthologs.

Datasets	AUC
Mouse-Human	90.39
Mouse-Fly	92.62
Mouse-Yeast	96.14
Human-Fly	88.89
Human-Yeast	85.63
Yeast-Fly	75.03

Table I
PERFORMANCE OF THE RECIPROCAL BLAST HIT METHOD ON THE FLY, YEAST, HUMAN AND MOUSE PROTEIN-PROTEIN INTERACTION DATASETS FROM DIP.

B. Reconstructing KEGG Orthologs Using Sequence and Protein-Protein Interaction Network Data

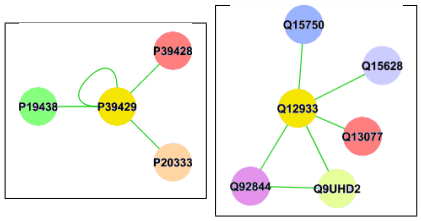

Figure 4. A sample 1 hop neighborhood around one of the matched orthologs (TNF receptor-associated factor 2 "P39429" in mouse and "Q12933" in human) according to the graph features (**LEFT:** 1 hop network around the "P39429" protein for mouse, **RIGHT:** 1 hop neighborhood around the "Q12933" protein for human). Similarly colored nodes are sequence homologous. The graph properties search for similar topology and sequence homology around the neighborhood of the nodes being compared.

Table III shows a comparison of the classifiers trained on the 1 hop and 2 hop Random Walk graph kernel and Shortest Path graph kernel scores as well as the degree distribution, BaryCenter [41], betweenness [40] and HITS (Hubs and Authorities) [24] centrality measures described in section II. We utilized the approach of Hall et al. [20] as implemented in WEKA [42] to rank the features based on their contribution to the classification performance. We found that the random-walk and shortest-path graph kernel scores were the top two ranked features in terms of their predictive ability. As seen from Table III, most of the classification methods show some improvement over the classifiers trained only on the BLASTp scores shown in table II. Notably, the ensemble classifier on the mouse-human dataset substantially outperforms its BLASTp counterpart. Table IV shows a few representative orthologous pairs that are missed by a regression-based classifier trained on BLASTp scores but are detected by the

Datasets	Adaboost j48 AUC	NB AUC	SVM AUC	Log. Reg. AUC	Ensemble AUC
Mouse-Human	87.79 (4)	90.15 (3)	77.31 (5)	90.29 (2)	90.30 (1)
Mouse-Fly	87.58 (4)	88.47 (3)	70.17 (5)	92.01 (1)	88.89 (2)
Mouse-Yeast	89.85 (5)	91.89 (2)	90.78 (3)	95.46 (1)	91.45 (4)
Human-Fly	81.35 (4)	87.70 (2)	65.90 (5)	88.90 (1)	84.42 (3)
Human-Yeast	82.97 (3)	81.26 (4)	63.68 (5)	85.50 (1)	84.19 (2)
Yeast-Fly	73.02 (3)	72.49 (4)	56.80 (5)	74.86 (1)	74.48 (2)
Average Rank	3.83	3	4.67	1.17	2.33

Table II
PERFORMANCE OF THE RECIPROCAL BLAST HIT SCORE AS A FEATURE TO THE DECISION TREE (J48), NAIVE BAYES (NB), SUPPORT VECTOR MACHINE (SVM) AND ENSEMBLE CLASSIFIERS ON THE FLY, YEAST, HUMAN AND MOUSE PROTEIN-PROTEIN INTERACTION DATASETS FROM DIP. VALUES IN PARENTHESIS ARE THE RANKS FOR THE CLASSIFIERS ON THE SPECIFIED DATASET.

Datasets	Adaboost j48 AUC	NB AUC	SVM AUC	Log. Reg. AUC	Ensemble AUC
Mouse-Human	95.19 (2)	88.72 (5)	90.78 (3)	89.57 (4)	96.18 (1)
Mouse-Fly	90.31 (1)	85.81 (3)	81.28 (4)	80.67 (5)	88.94 (2)
Mouse-Yeast	92.04 (3)	85.50 (4)	79.63 (5)	95.60 (1)	95.50 (2)
Human-Fly	88.18 (1)	83.10 (4)	75.03 (5)	87.04 (3)	87.20 (2)
Human-Yeast	82.83 (2)	81.26 (4)	78.22 (5)	81.57 (3)	84.84 (1)
Yeast-Fly	74.52 (1)	69.36 (4)	64.57 (5)	74.33 (2)	72.78 (3)
Average Rank	1.67	4	4.5	3	1.83

Table III
PERFORMANCE OF ALL THE COMBINED FEATURES (RECIPROCAL BLAST HIT SCORE, 1 AND 2 HOP SHORTEST PATH GRAPH KERNEL SCORE, 1 AND 2 HOP RANDOM WALK GRAPH KERNEL SCORE, BARYCENTER, BETWEENNESS, DEGREE DISTRIBUTION AND HITS) AS INPUT TO THE DECISION TREE (J48), NAIVE BAYES (NB), SUPPORT VECTOR MACHINE (SVM) AND ENSEMBLE CLASSIFIERS ON THE FLY, YEAST, HUMAN AND MOUSE PROTEIN-PROTEIN INTERACTION DATASETS FROM DIP. VALUES IN PARENTHESIS ARE THE RANKS FOR THE CLASSIFIERS ON THE SPECIFIED DATASET.

Mouse Protein	Human Protein	BLASTp score	RW 1HOP	SP 1HOP	RW 2HOP	SP 2HOP	BaryCenter	betweenness	Degree	HITS
P05627	P05412	481	104	197.35	612	290.27	0.71	0.69	0.01	0.26
P36898	P36894	725	28.13	222.85	90.66	576.51	0.35	0.77	0.01	3.06E-10
P39429	Q12933	870	48	126.18	150.47	187.45	0.79	0.11	0.01	1.20E-4

Table IV
KEGG ORTHOLOGS DETECTED USING THE ENSEMBLE CLASSIFIER UTILIZING ALL NETWORK FEATURES. THE ORTHOLOGS SHOWN IN THE ABOVE TABLE WERE MISSED BY THE BLAST LOGISTIC REGRESSION CLASSIFIER.

ensemble classifier trained on the network features and figure 4 shows the network neighborhood for one of such pairs (the TNF receptor-associated factor 2). This suggests that the combination of sequence homology with network-derived features may present a more reliable approach than simply relying on reciprocal BLASTp hits in identifying orthologs.

IV. DISCUSSION AND FUTURE WORK

The availability of genomes from a wide range of organisms has enabled the comparison and analysis of evolutionary relationships among genes across organisms through the reconstruction of phylogenies [39], common pathways [23], [27], and comparing gene functions [33], [15]. Ortholog detection methods present a powerful approach for finding genes that participate in similar biological processes across different organisms, extending our understanding of interactions between genes across different pathways, and understanding the evolution of gene families. We have explored a set of graph-based features that may be utilized for the detection of orthologs among different genomes by combining sequence-based evidence (such as BLAST-based sequence homology) with the network alignment algorithms available as part of the BiNA (Biomolecular Network Alignment) toolkit [38] as well as graph features extracted from the aligned protein-protein interaction networks such as degree distribution, BaryCenter [41], betweenness [40] and HITS (Hubs and Authorities) [24] centrality measures. The features may be used to score orthologous nodes in large biomolecular networks by comparing the neighborhoods around each node and scoring the nodes based on the similarity of their neighborhoods in the corresponding protein-protein interaction networks. Classifiers can then be trained using the scores to generate predictions as to whether or not a given pair of nodes are orthologous. Our results suggest that the algorithms that rely on orthology detection methods (e.g., for genome comparison) can potentially benefit from this approach to detecting orthologs (e.g., in the case of the comparison between mouse and human). The proposed method can also help identify proteins that have strong sequence homology but differ with respect to their interacting partners in different species (i.e., proteins whose functions may have diverged after gene-duplication).

Our experiments with the fly, yeast, mouse and human datasets suggest that the accuracy of identification of orthologs using the proposed method is quite competitive with that of reciprocal BLASTp method for detecting orthologs. The improvements obtained using information about interacting partners in the case of the mouse-human data (96.18% for the network-based method as opposed to 90.31% AUC for the reciprocal BLASTp method) suggest that the proposed technique could be useful in settings that benefit from accurate identification of orthologs (e.g., genome comparison).

The network neighborhood-based homology detection algorithm is implemented in BiNA (http://www.cs.iastate.edu/~ftowfic), an open source Biomolecular Network Alignment toolkit. The current implementation includes variants of the shortest path and random walk graph kernels for computing orthologs between pairs of subnetworks and the computation of various graph-based features available in the Java Universal

Graph Framework library [32] such as the degree distribution, BaryCenter [41], betweenness [40] and HITS (Hubs and Authorities) [24] centrality measures. The modular design of BiNA allows the incorporation of alternative strategies for decomposing networks into subnetworks and alternative similarity measures (e.g., kernel functions) for computing the similarity between nodes. It would be interesting to explore variants of methods similar to those proposed in this paper for improving the accuracy of detection of orthologous genes or proteins using other sources of data (e.g., gene co-expression networks).

Acknowledgments: This research was supported in part by an Integrative Graduate Education and Research Training (IGERT) fellowship to Fadi Towfic, funded by the National Science Foundation (NSF) grant (DGE 0504304) to Iowa State University and a NSF Research Grant (IIS 0711356) to Vasant Honavar. The authors are grateful to the BIBM-09 anonymous referees for their helpful comments on the manuscript.

REFERENCES

[1] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3390, 1997.

[2] A. Bairoch, R. Apweiler, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al. The Universal Protein Resource (UniProt). *Nucleic Acids Research*, 33:D154, 2005.

[3] S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome research*, 16(3):428–435, 2006.

[4] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and D.L. Wheeler. GenBank: update. *Nucleic Acids Research*, 32(Database Issue):D23, 2004.

[5] J.A. Blake, J.E. Richardson, C.J. Bult, J.A. Kadin, and J.T. Eppig. MGD: the mouse genome database. *Nucleic acids research*, 31(1):193, 2003.

[6] K.M. Borgwardt and H.P. Kriegel. Shortest-Path Kernels on Graphs. *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81, 2005.

[7] K.M. Borgwardt, H.P. Kriegel, SVN Vishwanathan, and N.N. Schraudolph. Graph Kernels For Disease Outcome Prediction From Protein-Protein Interaction Networks. *Proceedings of the Pacific Symposium of Biocomputing*, 2007.

[8] M.C. Brandon, M.T. Lott, K.C. Nguyen, S. Spolim, S.B. Navathe, P. Baldi, and D.C. Wallace. MITOMAP: a human mitochondrial genome database–2004 update. *Nucleic acids research*, 33(Database Issue):D611, 2005.

[9] JM Cherry, C. Adler, C. Ball, SA Chervitz, SS Dwight, ET Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, et al. SGD: Saccharomyces genome database. *Nucleic Acids Research*, 26(1):73, 1998.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. Cambridge Univ Pr, 2000.

[11] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[12] T. G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[13] Q. Dong, S.D. Schlueter, and V. Brendel. PlantGDB, plant genome database and analysis tools. *Nucleic acids research*, 32(Database Issue):D354, 2004.

[14] J.F. Dufayard, L. Duret, S. Penel, M. Gouy, F. Rechenmann, and G. Perrière. Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases. *Bioinformatics*, 21(11):2596–2603, 2005.

[15] J.A. Eisen and M. Wu. Phylogenetic analysis and gene functional predictions: phylogenomics in action. *Theoretical population biology*, 61(4):481–488, 2002.

[16] J. Flannick, A. Novak, C.B. Do, B.S. Srinivasan, and S. Batzoglou. Automatic parameter learning for multiple network alignment. *Lecture Notes in Computer Science*, 4955:214–231, 2008.

[17] P. Flicek, BL Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, et al. Ensembl 2008. *Nucleic acids research*, 36(Database issue):D707, 2008.

[18] Z. Fu, X. Chen, V. Vacic, P. Nan, Y. Zhong, and T. Jiang. MSOAR: A high-throughput ortholog assignment system based on genome rearrangement. *Journal of Computational Biology*, 14(9):1160–1175, 2007.

[19] L. Goodstadt and C.P. Ponting. Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human. *PLoS Comput Biol*, 2(9):e133, 2006.

[20] M.A. Hall and L.A. Smith. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, pages 235–239, 1999.

[21] AE Hirsh and HB Fraser. Protein dispensability and rate of evolution. *Nature*, 411(6841):1046–9, 2001.

[22] M. Kalaev, V. Bafna, and R. Sharan. Fast and accurate alignment of multiple protein networks. *Lecture Notes in Computer Science*, 4955:246, 2008.

[23] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, et al. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36(Database issue):D480, 2008.

[24] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[25] E. Koonin. Orthologs, paralogs and evolutionary genomics. *Annu. Rev. Genet*, 39:309–38, 2005.

[26] L. Li, C.J. Stoeckert, and D.S. Roos. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–2189, 2003.

[27] X. Mao, T. Cai, J.G. Olyarchuk, and L. Wei. Automated genome annotation and pathway identification using the KEGG Orthology (KO) as a controlled vocabulary. *Bioinformatics*, 21(19):3787–3793, 2005.

[28] T. Milenkoviæ and N. Pržulj. Uncovering Biological Network Function via Graphlet Degree Signatures. *Cancer Informatics*, 6:257, 2008.

[29] T. Mitchell. *Machine Learning*. McGraw-Hill, Boston, MA, 1997.

[30] K.P. O'Brien, M. Remm, and E.L.L. Sonnhammer. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic acids research*, 33(Database Issue):D476, 2005.

[31] H. Ogata, W. Fujibuchi, S. Goto, and M. Kanehisa. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic acids research*, 28(20):4021, 2000.

[32] J. O'Madadhain, D. Fisher, S. White, and Y. Boey. The JUNG (Java Universal Network/Graph) Framework. *University of California, Irvine, California*, 2003.

[33] M. Remm, C.E.V. Storm, and E.L.L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of molecular biology*, 314(5):1041–1052, 2001.

[34] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(Database Issue):D449, 2004.

[35] C. Steinbeck, C. Hoppe, S. Kuhn, M. Floris, R. Guha, and E.L. Willighagen. Recent Developments of the Chemistry Development Kit (CDK)-An Open-Source Java Library for Chemo-and Bioinformatics. *Current Pharmaceutical Design*, 12(17):2111–2120, 2006.

[36] R.L. Tatusov, M.Y. Galperin, D.A. Natale, and E.V. Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):33, 2000.

[37] Wenhong Tian and Nagiza F. Samatova. Pairwise alignment of interaction networks by fast identification of maximal conserved patterns. *Proc. of the Pacific Symposium on Biocomputing*, 2009.

[38] Fadi Towfic, M. Heather-West Greenlee, and Vasant Honavar. Aligning biomolecular networks using modular graph kernels. In *Lecture Notes in Bioinformatics*, 2009. To appear.

[39] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev. Automatic genome-wide reconstruction of phylogenetic gene trees. *Bioinformatics*, 23(13):i549, 2007.

[40] D.R. White and S.P. Borgatti. Betweenness centrality measures for directed graphs. *Social Networks*, 16(4):335–346, 1994.

[41] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–275. ACM New York, NY, USA, 2003.

[42] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, second edition, 2005.