Global alignment of multiple protein interaction networks with application to functional orthology detection

Rohit Singh*, Jinbo Xu†, and Bonnie Berger*‡§

*Computer Science and Artificial Intelligence Laboratory and ‡Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139; and †Toyota Technological Institute at Chicago, Chicago, IL 60637

Communicated by Silvio Micali, Massachusetts Institute of Technology, Cambridge, MA, July 16, 2008 (received for review March 12, 2008)

Protein–protein interactions (PPIs) and their networks play a central role in all biological processes. Akin to the complete sequencing of genomes and their comparative analysis, complete descriptions of interactomes and their comparative analysis is fundamental to a deeper understanding of biological processes. A first step in such an analysis is to align two or more PPI networks. Here, we introduce an algorithm, IsoRank, for global alignment of multiple PPI networks. The guiding intuition here is that a protein in one PPI network is a good match for a protein in another network if their respective sequences and neighborhood topologies are a good match. We encode this intuition as an eigenvalue problem in a manner analogous to Google's PageRank method. Using IsoRank, we compute a global alignment of the *Saccharomyces cerevisiae*, *Drosophila melanogaster*, *Caenorhabditis elegans*, *Mus musculus*, and *Homo sapiens* PPI networks. We demonstrate that incorporating PPI data in ortholog prediction results in improvements over existing sequence-only approaches and over predictions from local alignments of the yeast and fly networks. Previous methods have been effective at identifying conserved, localized network patterns across pairs of networks. This work takes the further step of performing a global alignment of multiple PPI networks. It simultaneously uses sequence similarity and network data and, unlike previous approaches, explicitly models the tradeoff inherent in combining them. We expect IsoRank—with its simultaneous handling of node similarity and network similarity—to be applicable across many scientific domains.

biological networks | graph isomorphism | network alignment | protein–protein interactions | functional coherence

A fundamental goal of biology is to understand the cell as a system of interacting components. In particular, the discovery and understanding of interactions between proteins has received significant attention in recent years. Toward this goal, high-throughput experimental techniques [e.g., yeast two-hybrid (1, 2) and coimmunoprecipitation (3)] have been invented to discover protein–protein interactions (PPIs) . The data from these techniques, which are still being perfected, are being supplemented by high-confidence computational predictions and analyses of PPIs (4–6). A powerful way of representing and analyzing this vast corpus of data is the PPI network: A network where each node corresponds to a protein and an edge indicates a direct physical interaction between the proteins.

As the size of PPI datasets for various species rapidly increases, comparative analysis of PPI networks across species is proving to be a valuable tool. Such analysis is similar in spirit to traditional sequence-based comparative genomic analyses; it also promises commensurate insights. As a phylogenetic tool, it offers a function-oriented perspective that complements traditional sequence-based methods. Comparative network analysis also enables us to identify conserved functional components across species (7) and perform high-quality ortholog prediction (i.e., identifying genes in different species derived from the same ancestral region). Solving these problems is crucial for transferring insights and information across species, allowing us to perform experiments in (say) yeast or fly and apply those insights toward understanding mechanisms of human diseases (8). Indeed, Bandyopadhyay *et al.* (9) have demonstrated that the use of PPI networks in computing orthologs produces orthology mappings that better conserve protein function across species (i.e., *functional orthologs*).

Previous work on PPI network alignment has almost exclusively focused on the local network alignment problem (see *Global vs. Local Network Alignment*) and has thus far targeted only pairwise alignments. The pioneering work of Kelley *et al.* (10, 11) described how BLAST similarity scores and PPI network information could be used to identify conserved functional motifs. Koyuturk *et al.* (12) proposed another method, motivated by biological models of duplication and deletion. Recently, Flannick *et al.* (7) proposed a new efficient approach, using modules of proteins to infer the alignment. Berg and Lassig (13) have proposed a Bayesian approach to this problem. Many of these methods limit the set of possible node-pairings based on sequence-based similarity scores or orthology predictions, and then add in network data to infer the alignment. This approach helps reduce the problem complexity, but lacks the flexibility of producing node-pairings that diverge from sequence-only predictions.

We note here that the graph alignment problem has also been studied in other domains. For example, in computer vision, the problem of matching a query image to an existing image in the database has often been formulated as a graph-matching problem, each image represented as a graph. Some of the solutions proposed in that domain use spectral techniques, i.e., they use eigenvalues computed based on each graph (14, 15). Our approach, which also constructs an eigenvalue problem (although, not for individual graphs) may be relevant in this domain as well.

In this article, we introduce an approach to comparative analysis of PPI networks to address the problem of finding the optimal *global* alignment between two or more PPI networks, aiming to find a correspondence between nodes and edges of the input networks that maximizes the overall "match" between the networks. We propose the IsoRank algorithm for multiple network alignment. The IsoRank algorithm simultaneously uses both PPI network data and sequence similarity data in an eigenvalue-based framework to

Preliminary versions of this work were presented at the RECOMB 2007, PSB 2008, and SODA 2008 Conferences.

Author contributions: R.S. and B.B. designed research; R.S. and B.B. performed research; R.S. and J.X. contributed new reagents/analytic tools; R.S. analyzed data; and R.S., J.X., and B.B. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

§To whom correspondence should be addressed. E-mail: bab@mit.edu.

This article contains supporting information online at www.pnas.org/cgi/content/full/0806627105/DCSupplemental.

compute network alignments, the relative weight of the two data sources being a free parameter.

We use IsoRank to simultaneously align the PPI networks of *Saccharomyces cerevisiae*, *Drosophila melanogaster*, *Caenorhabditis elegans*, *Mus musculus*, and *Homo sapiens*, the species that make up the bulk of available PPI data. The conserved subgraphs in this alignment are larger and more varied than those produced by previous methods, which performed pairwise network alignments. We also use the alignment results to predict functional orthologs across species and demonstrate that incorporating PPI data in ortholog prediction results in improvements over existing sequence-only approaches such as Homologene (www.ncbi.nlm.nih.gov/sites/entrez?db=homologene) and Inparanoid (16); moreover, we find our results compare favorably with those from local alignments on the yeast and fly networks (9). To test the biological quality of our predictions, we introduce a direct, automated method for scoring the quality of an ortholog list.

Background

Global vs. Local Network Alignment. In general, the goal in a network alignment problem is to find a common subgraph (i.e., a set of conserved edges) across the input networks. Corresponding to these conserved edges, there exists a mapping between the nodes of the networks. For example, when protein a_1 from network G_1 is mapped to proteins a_2 from G_2 and a_3 from G_3, then a_1, a_2, and a_3 refer to the same node in the set of conserved edges. What makes the problem difficult is the tradeoff involved: Maximizing the overlap between the networks (i.e., the number of conserved edges), while ensuring that the proteins mapped to each other are, as far as possible, evolutionarily related. In most existing approaches, and in this article, sequence similarity is used as a measure of evolutionary relationship, albeit an approximate one. However, more sophisticated measures are certainly possible; e.g., those that incorporate gene order (synteny).

The network alignment problem can be formulated in various ways, depending on the kind of input (pairwise vs. multiple alignments) and the scope of node mapping desired. Here, we draw an analogy from the sequence alignment problem to distinguish between local and global network alignment, the latter being the focus of this article.

Local Network Alignment (LNA). The goal in LNA is to find multiple, unrelated regions of isomorphism (i.e., same graph structure) between the input networks, each region implying a mapping independently of others. Many independent, high-scoring local alignments are usually possible between two input networks; in fact, the corresponding local alignments need not even be mutually consistent (i.e., a protein might be mapped differently under each alignment). The motivations behind local sequence alignment and local network alignment are similar—the former is often used to search for a conserved motif in the target species; the latter would be used to search for a known functional component (e.g., pathways, complexes, etc.) in a new species.

Global Network Alignment (GNA). The aim in GNA is to find the best overall alignment between the input networks. The mapping in a GNA should cover all of the input nodes: Each node in an input network is either matched to one or more nodes in the other network(s) or explicitly marked as a gap node (i.e., with no match in another network). In contrast, a LNA algorithm is essentially intended for finding similar motifs/patterns between two networks, and the mappings corresponding to different motifs may be mutually inconsistent. In GNA, however, our goal is to find a single consistent mapping covering all nodes across all input graphs. Furthermore, it must be transitive: If a_1 in G_1 is mapped to a_2 in G_2 and a_2 is mapped to nodes a_3, a_3' in G_3, then a_1 should also be mapped to a_3, a_3'. The global scope of GNA enables species-level comparisons. Analogous to global sequence alignment, which is often used for comparing genomic sequences to understand variations between species (17), GNA may be used to compare interactomes and for understanding cross-species variations. Also, the GNA problem is related to the detection of functional orthologs, as we discuss in *Results*.

The focus of this article is on the global network alignment problem, which has previously received little attention in the literature. One can imagine using LNA to estimate GNA: Use LNA methods to compute possible matches for each protein; then select the mapping best supported overall by the alignment results. A similar approach has been used for functional ortholog detection (9). Unfortunately, this approach is somewhat complex, and more importantly, ignores inconsistencies across local alignments so that the node matches in the final alignment might not even be mutually consistent. Instead, we propose a simpler, yet powerful algorithm.

IsoRank Algorithm

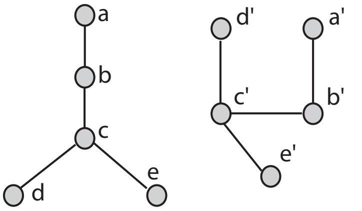
To start with, we consider the simple case of pairwise GNA. Here, the input consists of two PPI networks G_1 and G_2 (recall that the nodes of these networks correspond to proteins). Each edge e may have an associated edge weight $w(e)$ $(0 \leq w(e) \leq 1)$.

Furthermore, the input also consists of a similarity measure between the nodes of the two networks. These scores may be defined only for some node-pairs (i.e., protein-pairs). In this article, we use BLAST similarity scores, but additional measures (e.g., synteny-based scoring, functional similarity) can be incorporated. The desired output is a mapping between the nodes of the two networks that maximizes a convex combination[‖] of the following objective functions: (1) the size of the common graph implied by the mapping, and (2) the aggregate sequence similarity between nodes mapped to each other. Given the inputs, we construct an eigenvalue problem whose solution leads to a mapping between the nodes. From this mapping, the set of conserved edges can be easily computed.

Our algorithm works in two stages. It first associates a functional similarity score with each possible match between nodes of the two networks. Let R_{ij} be this score for the protein pair (i, j) where i is from network G_1 and j is from network G_2. Given network and sequence data, we construct an eigenvalue problem and solve it to compute R (the vector of all R_{ij}). The eigenvalue problem explicitly models the tradeoff between the twin objectives of high network overlap and high sequence similarity between mapped node-pairs. The second stage constructs the mapping for the GNA by extracting a set of high-scoring, mutually consistent matches from R.

Computing *R* (Setting Up the Constraints). To compute the functional similarity score R_{ij}, we pursue the intuition that (i, j) is a good match if the i and j sequences align and their respective neighbors are a good match with each other. For ease of explanation, let us first focus on the network-only data case. The intuition is to set up a system of constraints, where we compute the neighborhood scores in a recursive fashion. More precisely, we require Eq. **1** (see below) to hold for all possible pairs (i, j). There, $N(a)$ is the set of neighbors of node a; $|N(a)|$ is the size of this set; and V_1 and V_2 are the sets of nodes in networks G_1 and G_2, respectively. These equations require that the score R_{ij} for any match (i, j) be equal to the total support provided to it by each of the $|N(i)||N(j)|$ possible matches between the neighbors of i and j. In return, each node-pair (u, v) must distribute back its entire score R_{uv} equally among the

[¶] In the absence of any sequence similarity information, the optimum mapping will correspond to a maximum common subgraph (MCS) between G_1 and G_2 (i.e., the largest graph that is isomorphic to subgraphs of both) and the corresponding node-mapping such that each node is mapped to at most one node in the other network. Nodes not mapped to any other node are referred to as gap nodes. MCS is an NP-complete problem and thus approximate solutions, especially for the large-sized PPI networks, are essential. Also, when incorporating sequence data, the global alignment problem is no longer a pure MCS problem.



R

	a'	b'	c'	d'	e'
a	0.0312		0.0937		
b		0.1250		0.0625	0.0625
c	0.0937		0.2812		
d		0.0625		0.0312	0.0312
e		0.0625		0.0312	0.0312

$$R_{aa'} = \frac{1}{4} R_{bb'}$$

$$R_{bb'} = \frac{1}{3} R_{ac'} + \frac{1}{3} R_{a'c} + R_{aa'} + \frac{1}{9} R_{cc'}$$

$$R_{dd'} = \frac{1}{9} R_{cc'}$$

$$R_{cc'} = \frac{1}{4} R_{bb'} + \frac{1}{2} R_{be'} + \frac{1}{2} R_{bd'} + \frac{1}{2} R_{eb'} + \frac{1}{2} R_{db'} + R_{ee'} + R_{ed'} + R_{de'} + R_{dd'}$$

Fig. 1. Intuition behind the algorithm: Here we show, for a pair of small, isomorphic graphs how the vector of pairwise scores *R* is computed. For each possible pairing (*i,j*) between nodes of the two graphs, we compute the score R_{ij}. The scores are constrained to depend on the scores from the neighborhood as described by Eq. **1**. Only a partial set of constraints is shown here. The scores R_{ij} are computed by starting with random values for R_{ij} and by using the recursive methods described below to find values that satisfy these constraints; here we show the vector *R* reshaped as a table for ease of viewing (empty cells indicate a value of zero). The second stage of our algorithm uses *R* to extract likely matches. One strategy could be: choose the highest-scoring pair, output it, remove the corresponding row and column from the table, and repeat. This strategy will return the correct mapping {(*a*, *a*′), (*b*, *b*′), (*c*, *c*′), (*d*, *d*′), (*e*, *e*′)}. The {*d*, *e*} → {*d*, *e*} mapping is ambiguous; using sequence information, such ambiguities can be resolved.

$|N(u)||N(v)|$ possible matches between *its* neighbors. We note that these equations also capture nonlocal influences on R_{ij}: The score R_{ij} depends on the score of neighbors of *i* and *j* and the latter, in turn, depend on the neighbors of the neighbors and so on. The extension to the weighted-graph case is intuitive: The support offered to neighbors is then in proportion to the edge weights (Eq. **2**). Clearly, Eq. **1** is a special case of Eq. **2** when all of the edge weights are 1.

In Eq. **3**, we rewrite Eq. **1** in matrix form. Here, *A* is a $|V_1||V_2| \times |V_1||V_2|$ matrix and $A[i, j][u, v]$ refers to the entry at the row (i, j) and column (u, v) (the row and column are doubly indexed). Eq. **2** can be similarly rewritten.

$$R = \Sigma R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv} \quad i \in V_1, j \in V_2, \qquad [1]$$

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i, u)w(j, v)}{\sum_{r \in N(u)} w(r, u) \sum_{q \in N(v)} w(q, v)} R_{uv}$$

$$i \in V_1, j \in V_2, \qquad [2]$$

$$R = AR, \quad \text{where}$$
$$A[i, j][u, v] = \begin{cases} \dfrac{1}{|N(u)||N(v)|} & \text{if } (i, u) \in E_1, (j, v) \in E_2. \\ 0 & \text{otherwise} \end{cases} \qquad [3]$$

The vector *R* is determined by finding a nontrivial solution to these equations (a trivial solution is to set all R_{ij}'s to zero). In Fig. 1, we illustrate, on a pair of small graphs, how the equations capture the graph topology; their solution also confirms our intuition: node pairs that match well have higher R_{ij} scores.

Computing *R* (Solving the Constraints). In general, to solve the above equations, we observe that these equations describe an eigenvalue problem (see Eq. **3**). The value of *R* we are interested in is the principal eigenvector of *A*. Note that *A* is a stochastic matrix (i.e., each of its columns sums to 1) so that the principal eigenvalue is 1. In the case of biological networks, *A* is typically a very large matrix ($\approx 10^8 \times 10^8$ for fly-vs.-yeast GNA); however, *A* and *R* are both very sparse, so *R* can be efficiently computed by iterative techniques. We use the *power method*, an iterative technique often used for large eigenvalue problems. The power method repeatedly updates *R* as per the update rule:

$$R(k + 1) \leftarrow AR(k)/|AR(k)|, \qquad [4]$$

where $R(k)$ is the value of the vector *R* in the *k*-th iteration and has unit norm. In case of a stochastic matrix (like *A*), the power method will probably converge to the principal eigenvector.

The incorporation of other information, e.g., BLAST scores, into this model is straightforward. Let B_{ij} denote the score between *i* and *j*; for instance, B_{ij} can be the Bit-Score of the BLAST alignment between sequences *i* and *j*. B_{ij} need not even be numeric—they can be binary. Let *B* be the vector of B_{ij}. We first normalize *B*: $E = B/|B|$ so that all sequence similarity scores sum to 1. The eigenvalue equation is then modified to a convex combination of network and sequence similarity scores:

$$\begin{aligned} R &= \alpha AR + (1 - \alpha)E, \qquad 0 \le \alpha \le 1, \text{ or} \\ R &= (\alpha A + (1 - \alpha)E1^{\mathrm{T}})R. \end{aligned} \qquad [5]$$

Eq. **5** also describes an eigenvalue problem and is solved by similar techniques as Eq. **3** (here, we use $|R|_1 = 1$). In this computation, α controls the weight of the network data (relative to sequence data), e.g., $\alpha = 0$ implies no network data will be used, whereas $\alpha = 1$ indicates only network data will be used. Tuning α allows us to analyze the relative importance of PPI data in finding the optimal alignment. The parameter α also controls the speed of convergence of this stage, with the algorithm converging in $O(\log(1/1\text{-}\alpha))$ iterations.

Multiple GNA. When the input consists of more than two networks, we repeat the above process for every pair of input networks, i.e., we compute the functional similarity scores *R* for every pair of input networks.

Extracting Node Mappings from *R*. At this stage in the algorithm, we have a score R_{ij} for every pair of nodes not from the same network; typically, for more than 99% of node-pairs, this score is zero. This score indicates how good a match *i* and *j* are for each other when considering both network and sequence data. To extract a node mapping from these scores, we need to identify pairs of nodes that have high R_{ij} scores, at the same time ensuring that the mapping obeys transitive closure; i.e., if it contains the pairs (a, b) and (b, c), then it also contains (a, c). The node mappings can be done in two ways.

One-to-one Mappings. Here, we require that any node be mapped to *at most* one other node per species. Biologically, the single match for a node can be interpreted as the closest functional ortholog of the corresponding protein in the other species. Computationally, this constraint has the advantage of being solvable efficiently and without requiring any free parameters (thus reducing the risk of overfitting). The disadvantage is that this formulation ignores issues like gene duplication because more than one match per species is possible. Here, we use this mapping only for a case study of the yeast-fly GNA (the two species with the most amount of available PPI data), using it to identify pairs of closest functional orthologs across the two species.

Many-to-many Mappings. The more general case is when a node can be mapped to more than one node in another species. The mapping produced here is of the same form as Clusters of Orthologous Genes (COGs) (18, 19): The entire set of nodes across all networks is partitioned, each partition corresponding to a set of nodes mapped to each other. Each set may contain zero, one or many nodes from each species. The intuition here is that the proteins in a single set are functional orthologs of each other, i.e., are evolutionarily related and perform the same function in their respective species.

To construct such a partition of genes from the set of scores R computed in the previous approach, we design an algorithm that searches for sets of genes such that each set obeys the following requirements: each gene in the set has high pairwise R scores with most other genes in the set; there are no genes outside each set with this property; and there are a limited number of genes from each species. This limit varies from species to species: more genes from *H. sapiens* are allowed in the set than from *S. cerevisiae*, reflecting the intuition that there is greater gene duplication in the former.

Our algorithm computes each set of orthologous proteins by identifying a seed pair of match nodes and extending it by using a modified greedy algorithm. We first construct a k-partite graph H from the scores R. Each of its k parts contains nodes from one of the input networks. Edges are only allowed between nodes from different parts. The presence of an edge e_{ij} implies that node i (from G_1) can potentially be mapped to j (from G_2), i.e., $R_{ij} > 0$; the edge-weight R_{ij} indicates the strength of the potential match.

While the k-partite graph H has any edges remaining:

1. Select the edge (i, j) with the highest score (let i be from G_1 and j from G_2). Initialize a new match-set with i and j as its initial members.

2. In every other species (G_3,. . . ,G_k), if a node l exists such that (*i*) R_{il} and R_{jl} are the highest scores between l and any node in G_1 and G_2, respectively and, (*ii*) the scores $R_{il} \geq \beta_1 R_{ij}$, and $R_{jl} \geq \beta_1 R_{ij}$, add it to the set. This set of nodes forms the primary match-set; it has at most one node from each species.

3. Add upto r-1 nodes from different parts of the graph to the primary match-set. Suppose u (from G_x) is in the primary match-set. Then, a node v (from G_x) is added to the set if $R_{vw} \geq \beta_2 R_{uw}$ for each node w (w≠u) in the primary set.

4. Remove from H all of the nodes in this match-set and their edges.

Here, the parameters r, β_1, β_2 are user-defined ($0 < \beta_1, \beta_2 < 1$); we chose their values such that the functional coherence (defined in next section) of the resulting sets of matched nodes was maximized. Note that Step 2A alone gives a maximum k-partite matching.

Given a mapping between the nodes of the input networks, the corresponding common subgraph in the GNA can be identified relatively easily. For example, if a_1 is aligned to a_2, and b_1 is aligned to b_2, the output subgraph should contain an edge between the corresponding nodes if and only if both the input networks contain supporting edges.

Results

Global Alignment of Yeast, Fly, Worm, Human, and Mouse Networks. We analyzed the common subgraph implied by the multiple alignments of the following five species: *S. cerevisiae*, *D. melanogaster*, *C. elegans*, *M. musculus*, and *H. sapiens*. The common subgraph corresponding to the global alignment has 1,663 edges that are supported by edges in at least two PPI networks and 157 edges that are supported by at least three networks. There are very few edges with support from four or more species; however, this is not surprising because the worm and mouse networks are very small. The size of the common subgraph is relatively small (only ≈ 5% of the human PPI network). One reason for the small overlap between the PPI networks could be that the current PPI data are both incomplete and noisy. As the quality and quantity of data improves, this overlap should increase further. Even with this incomplete data, we believe that the currently computed (partial) set of node-pairings is robust. In supporting information (SI) Fig. S2, we describe experiments that suggest that the eigenvalue formulation is robust to errors in PPI data, especially when sequence data are provided.

A naive approach to multiple network alignment would use current sequence-based orthology predictions to perform the mapping; in contrast, by incorporating both sequence and network data, our algorithm performs significantly better. The common subgraph implied by Homologene's sequence-only mapping contains only 509 edges with support in two or more species and 40 edges with support in three or more species. Thus, the addition of network topology in computing the mappings increases the size of the common subgraph by over threefold (from 509 to 1,663 and 40 to 157, respectively). A direct comparison cannot be performed against Inparanoid orthology lists because the Inparanoid's pairwise orthology lists cannot be used for multiple network alignment. Instead, we evaluated the total number of conserved edges implied by Inparanoid in 10 $(=\binom{5}{2})$ pairwise network alignments. Even though this final number, 1,172, likely over-counts some conserved edges, it is significantly less than the number of conserved edges implied by our algorithm.

The common subgraph in the global alignment consists of multiple components, many of which are significantly larger than those from local alignment methods. Also, unlike the latter, these subgraphs correspond to a variety of topologies: Linear, complex-like, tree-shaped, etc. Some of the subgraphs are also enriched in proteins involved in a specific function (see Fig. 2; also see *SI Text* for more details).

Here, we present a set of functional orthologs (FO) across five species: Yeast, fly, worm, mouse, and human. We note here that our approach complements, rather than replaces, existing orthology prediction approaches. For example, whereas IsoRank predicts orthologs from a functional perspective, Homologene makes predictions from an evolutionary perspective. Our FO mapping is simply the node mapping computed by our algorithm (see *SI Text* for the list of FOs). Of the 86,932 proteins from the five species, 59,539 (68.5%) of the proteins in our list were matched to at least one protein in another species (i.e., had at least one FO). In contrast, Homologene has lower coverage, predicting at least one ortholog for only 33,434 (38.5%) proteins. Also, as we describe next, we believe that the biological quality of our method's ortholog predictions compare favorably with Homologene and Inparanoid.

Functional Coherence: Evaluating Orthology Predictions. In this article, we also propose a direct, automated method for scoring the quality of an ortholog list. The method is motivated by the lack of automated, direct measures of ortholog-list quality. Currently, the most common approach is a manual case-by-case analysis of a few protein–pairs grouped differently under the two lists; however, this anecdotal approach cannot be easily extended to a comprehensive evaluation. Recently, Chen *et al.* (20) have described an *indirect*

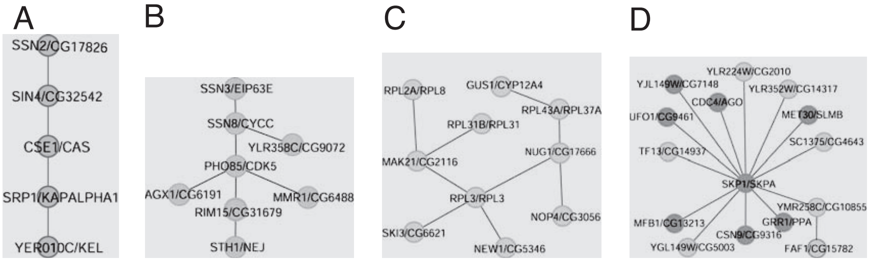


Fig. 2. Selected subgraphs of the yeast-fly GNA: The node labels indicate the corresponding yeast/fly proteins (the two separated by a '/'). The subgraphs span a variety of topologies and are often enriched in specific functions. For example, in *D*, the nodes for which at least one of the corresponding proteins is known to be involved in ubiquitin ligase activity are shaded.

automated approach where they compare many ortholog lists to identify the list(s) with the best overall agreement with the remaining ones. However, this approach only measures mutual agreement between the orthology lists, not if they each make predictions which are biologically plausible.

Our direct, automated measure of ortholog quality is based on using functional information. The intuition here is simple: Given an ortholog list, we select those sets of orthologous proteins for which functional information is available for many members of the set; this is measured by the presence/absence of GO annotation. A combination of GO annotation and PPI data has been explored before, for example, in predicting functions of unannotated proteins (21). Usually, the number of such selected sets is large enough to generate robust statistics (see *SI Text*). For each selected set, we collect all of the Gene Ontology (GO) terms corresponding to proteins in it. We excluded GO terms describing cellular compartment or location; we believe that GO terms describing molecular function or biological process are appropriate for capturing the protein's functional role. We evaluated whether these GO terms describe similar functions, computing a *coherence* score for the set. Higher scores imply higher coherence, indicating that the genes in the set all perform similar functions. Finally, an aggregate score (across all sets) is computed (see *SI Text* for details).

The functional coherence of our predicted functional orthologs is comparable with that of Homologene and Inparanoid predictions. The functional coherence scores are: 0.220 (our predictions), 0.223 (Homologene), and 0.206 (mean score across Inparanoid's pairwise ortholog sets). Homologene's slightly better score may partly be due to its use of data from many species (>5). At the same time, our predicted FOs do not deviate drastically from sequence-only predictions: 66% of protein-pairs grouped together by Inparanoid are also grouped together by our approach.

Case-study: Functional Orthologs between Yeast and Fly. In relative terms, the *S. cerevisiae* and *D. melanogaster* networks are two of the largest PPI networks currently available. A pairwise comparison of these networks is interesting because the impact of including PPI data may be more apparent here (recall that in the 5-species GNA described earlier, some of the species had relatively small PPI datasets). In performing this alignment, we focused on extracting one-to-one mappings between yeast and fly proteins; i.e., for each protein we searched for the one protein in the other species most similar to it.

Although, this approach does not adequately address the issues of gene duplication, the discovery of the single, closest functional ortholog between the species is of practical value (e.g., when replicating experiments done in yeast, in fly, and vice versa).

To find this mapping, we computed functional similarity scores R and then used a bipartite matching algorithm to find the one-to-one mapping (see *SI Text* for more details). The common subgraph corresponding to the global alignment between the yeast and fly PPI networks has 1,420 edges (where $\alpha = 0.6$; the criterion for choosing α is described in *SI Text*). Although, this still represents a small fraction of the individual network sizes, it is relatively large when compared with the size of 5-species GNA.

When we interpret the mapping between the two species as functional orthologs, the IsoRank results compare favorably with Bandyopadhyay *et al.*'s results. The latter method was the first to systematically compute functional orthologs using PPI data; it uses sequence matches and then local network alignment scores to give probabilistic scores to node pairs. Our method has the advantage that it guarantees the predicted sets of FOs will be mutually consistent and achieves higher genome coverage—PathBlast's yeast-vs.-fly local alignments cover only 20.56% of the genes covered by our global alignment. In many cases the FO predictions between the two methods are partially or fully consistent (see Table S2), i.e., FOs predicted by our method are also the likely FOs predicted by their method. In a few other cases, predictions of the two methods differ. At least in some such cases, our method's predictions are better supported by evidence. For example, our method predicts *Bic* (in fly) as the FO of *Egd* (in yeast). The method of Bandyopadhyay *et al.* (9) is ambiguous here, because *Bcd*, its predicted FO of *Egd*, is also predicted as a FO of *Btt1*. Furthermore, there is experimental evidence that both *Egd* and *Bic* are components of the Nascent Polypeptide-Associated Complex (NAC) in their respective species, lending support to our prediction; in contrast, *Bcd* does not seem to be involved in NAC.

Discussion

Over the last few years, the corpus of PPI data has increased at an exponential size and the rapid pace of data accumulation continues. Taking advantage of these large PPI datasets will pose significant computational challenges. We believe that two particularly important classes of problems are likely to be: (1) understanding the structure of these networks, i.e., what general graph-theoretic characteristics do these graphs share and what are the biological implications of the commonalities; and (2) combining this data with other biological datasets to gain insights not accessible from individual datasets. Here, we have attempted to address certain aspects of both these problems.

The IsoRank algorithm presented in this article performs a simultaneous global alignment of multiple PPI networks. The global alignment allows the comparison of overall structure of various networks, allowing us to make inferences about what is conserved and what is not. The algorithm also provides for explicit modeling of sequence and network scores, by means of a single "weight" parameter. Such combination of networks and sequence data should improve our understanding of the functional correspondence between genes/proteins across species. Another benefit is that IsoRank is, by design, tolerant to errors in the input (e.g., missing or spurious edges) and takes advantage of edge confidence scores and other biological signals (e.g., functional information), when available (see *SI Text* for more details).

Our network alignment method is different from previous methods, which focus on local alignment, i.e., finding independent, localized regions of similarity between the networks. The latter are useful when one wants to find, say, the analogous pathway in fly for a given mouse pathway. However, our goal is an overall comparison to get general inferences about network structure. It is possible to coerce the multiple local alignments into something approaching a global alignment, e.g., in a way similar to that described by Bandyopadhyay *et al* (9). However, it soon gets quite complicated. Our method, in contrast, is simpler and requires fewer parameters, whereas providing results that compare favorably with their approach. It is also more easily extensible to multiple species. However, there may be cases when a local approach may be better-suited than a global approach; e.g., Murali *et al.* (22) have discussed such a scenario in the context of predicting protein function.

The results of the global alignment can be directly interpreted as describing functional orthologs (FOs) across species. Rather than relying excessively on sequence-score based heuristics, IsoRank uses functional information (from PPI networks) to predict FOs. The functional coherence scores suggest that our approach is a simpler and better way of capturing functional similarities between proteins. However, our approach has certain limitations. In the absence of sequence data, IsoRank cannot distinguish k-regular graphs. It does not make as detailed and fine-tuned a use of sequence data as some existing sequence-only methods do; this is both good and bad: Some fine-tuning may increase the number of true positives, whereas excessive fine-tuning might result in overfitting and more false positives. However, reliance on PPI data is hindered by the fact that for many proteins, no PPI data are available. In such cases, the algorithm's ability to identify functionally related sets of proteins may suffer. However, the expected increase in the availability of PPI data should help overcome this limitation. Also, previous work has explored an integrative approach to predicting protein function (23); the FO predictions made by IsoRank can be incorporated in such a framework.

Another contribution of this article is an automated, unbiased, direct measure of the quality of orthology lists. We have used it to see how IsoRank's putative orthologs compare to the ones produced by Homologene and Inparanoid. It is unclear how to apply the functional coherence measure to Bandyopadhyay *et al.*'s probabilistic node-pair assignments because, unlike the above methods, they do not clearly partition the nodes into disjoint ortholog sets. This measure is general enough to be used in any setting where orthology predictions are made as disjoint sets of nodes [i.e., in the same form as COG (18)].

One current goal is to further improve the extraction of node mappings from the computed functional similarity scores R. Also, there is clearly room in our approach to leverage sequence information in a more sophisticated way. The set of conserved edges across the various networks should be studied in greater detail to understand the kind of edges that are conserved.

Our approach has similarities to Google's PageRank algorithm for ranking webpages in order of relevance. In PageRank, a graph is constructed where each node corresponds to a webpage and an edge from node a to b indicates that the webpage a links to webpage b. To identify the most authoritative webpage, the algorithm pursues the intuition that an authoritative page is one that is pointed to by many other authoritative pages. This intuition is formalized by constructing equations that relate the authoritativeness scores of the various webpages; these scores are then found by an eigenvalue approach. The problem we address is quite different: Comparing sets of graphs to find correspondences between nodes, rather than ranking nodes of a single graph. The similarities in the approaches lies in the idea of looking at neighborhood topology to compute the final solution.

Methods

Datasets. We constructed PPI networks for five species: *S. cerevisiae*, *D. melanogaster*, *C. elegans*, *M. musculus*, and *H. sapiens*. These networks were constructed by combining data retrieved from the DIP, BioGRID and HPRD databases. The relative coverage of the PPI data varied heavily; the number of edges per species were: 36,387 (human), 31,899 (yeast), 25,831 (fly), 4,573 (worm), and 255 (mouse). Sequence data for the various proteins was retrieved from Ensembl and the BLAST Bit-values were used as the score of sequence similarity between input proteins. Even in species with relatively high PPI coverage (e.g., yeast), there were many proteins that did not occur in the PPI network. To ensure that these proteins were included in the functional ortholog lists, we added singleton (disconnected) nodes corresponding to each such protein in the respective PPI networks, thus using only sequence data.

Parameter Choices. When performing the alignment, we chose the following parameter settings: $\alpha = 0.6$, $r = 5$, $\beta_1 = 0.1$, $\beta_2 = 0.1$. These settings correspond to the node mapping with the best functional coherence (see *Results*).

Note Added in Proof. Since preliminary versions of this work appeared [PSB08], other approaches have been proposed for the multiple alignment problem.[‖][**]

ACKNOWLEDGMENTS. We thank Michael Baym and Michael Schnall-Levin for helpful comments, Leonid Chindelevitch for additionally proving the convergence bound, and the referees for valuable advice and suggestions.

[‖] Flannick J, *et al.*, Twelfth Annual International Conference on Research in Computational Molecular Biology, March 30–April 2, 2008, Singapore.

[**] Maxim Kalaev M, Bafna V, Sharan R, Twelfth Annual International Conference on Research in Computational Molecular Biology, March 30–April 2, 2008, Singapore.

1. Uetz P, *et al*. (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae. Nature* 403(6770):623–627.
2. Ito T, *et al.* (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci USA* 98:4569–4574.
3. Krogan NJ, *et al.* (2006) Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae. Nature* 440(7084):637–643.
4. Han JD, Bertin N, Hao T, Goldberg D, Berriz G, Zhang L, Dupuy D, Walhout A, Cusick M, Roth F, Vidal M (2004) Evidence for dynamically organized modularity in the yeast protein–protein interaction network. *Nature* 430(6995):88–93.
5. Nabieva E, Jim K, Agarwal A, Chazelle B, Singh M (2005) Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21 Suppl(1):i302–10.
6. Yook SH, Oltvai ZN, Barabasi AL (2004) Functional and topological characterization of protein interaction networks. *Proteomics* 4(4):928–942.
7. Flannick J, Novak A, Srinivasan BS, McAdams HH, Batzoglou S (2006) Graemlin: General and robust alignment of multiple large interaction networks. *Genome Res* 16(9):1169–1181.
8. Uetz P, *et al.* (2005) Herpesviral protein networks and their interaction with the human proteome. *Science* 311(5758):187–188.
9. Bandyopadhyay S, Sharan R, Ideker T (2006) Systematic identification of functional orthologs based on protein network comparison. *Genome Res* 16(3):428–435.
10. Kelley BP, *et al.* (2004) Pathblast:a tool for alignment of protein interaction networks. *Nucleic Acids Res* 32(Web Server issue):W83–88.
11. Kelley BP, *et al.* (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc Natl Acad Sci USA* 100:11394–11399.
12. Koyuturk M, Grama A, Szpankowski W (2005) Pairwise local alignment of protein interaction networks guided by models of evolution. *RECOMB* 3500:48–65.
13. Berg J, Lässig M (2006) Cross-species analysis of biological networks by Bayesian alignment. *Proc Natl Acad Sci USA* 103:10967–10972.
14. Leordeanu M, Hebert M (2005) A spectral technique for correspondence problems using pairwise constraints. *Tenth International Conference on Computer Vision* (IEEE, Washington, DC) Vol. 2, pp 1482–1489.
15. Cour T, Srinivasan P, Shi J (2006) Balanced spectral matching. *The Neural Information Processing System* (Neural Information Processing Systems Foundation, La Jolla), pp 313–320.
16. O'Brien KP, Remm M, Sonnhammer E.L (2005) Inparanoid: A comprehensive database of eukaryotic orthologs. *Nucleic Acids Res* 33(Database issue):D476–480.
17. Kellis M, Patterson N, Birren B, Berger B, Lander E (2004) Methods in comparative genomics:genome correspondence, gene identification and regulatory motif discovery. *J Comput Biol* 11(2–3):319–355.
18. Tatusov R.L, *et al.* (2003) The COG database: An updated version includes eukaryotes. *BMC Bioinformatics* 4:41. Available at www.biomedcentral.com/1471–2105/4/41.
19. Tatusov RL, Koonin EV, Lipman DJ (1997) A Genomic Perspective on Protein Families. *Science* 278(5338):631–637.
20. Chen F, Mackey AJ, Vermunt JK, Roos DS (2007) Assessing Performance of Orthology Detection Strategies Applied to Eukaryotic Genomes. *PLoS One* 2(4):e383. Available at www.plosone.org/article/fetchArticle.action?articleURI=info%3Adoi%2F10.1371%2Fjournal.pone.0000383.
21. Letovsky SK, S (2003) Prediction protein function from protein–protein interaction date:a probabilistic approach. *Bioinformatics* 19 (Suppl 1):i197–204.
22. Murali TM, Wu CJ, Kasif S (2006) The art of gene function prediction. *Nat Biotechnol* 24(12):1474–1475.
23. Karaoz U, *et al.* (2004) Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci USA* 101:2888–2893.