Pairwise Global Alignment of Protein Interaction Networks By Matching Neighborhood Topology

Rohit Singh[1], Jinbo Xu[2], and Bonnie Berger[1]**

[1] Computer Science and AI Lab., Massachusetts Institute of Technology
rsingh@mit.edu, bab@mit.edu
[2] Toyota Technological Institute, Chicago, USA
j3xu@tti-c.org

Abstract. We describe an algorithm, IsoRank, for global alignment of two protein-protein interaction (PPI) networks. IsoRank aims to maximize the overall match between the two networks; in contrast, much of previous work has focused on the local alignment problem— identifying many possible alignments, each corresponding to a local region of similarity. IsoRank is guided by the intuition that a protein should be matched with a protein in the other network if and only if the neighbors of the two proteins can also be well matched. We encode this intuition as an eigenvalue problem, in a manner analogous to Google's PageRank method. We use IsoRank to compute the first known global alignment between the *S. cerevisiae* and *D. melanogaster* PPI networks. The common subgraph has 1420 edges and describes conserved functional components between the two species. Comparisons of our results with those of a well-known algorithm for local network alignment indicate that the globally optimized alignment resolves ambiguity introduced by multiple local alignments. Finally, we interpret the results of global alignment to identify functional orthologs between yeast and fly; our functional ortholog prediction method is much simpler than a recently proposed approach and yet provides results that are more comprehensive.

1 Introduction

A fundamental goal of biology is to understand the cell as a system of interacting components and, in particular, how proteins in the cell interact with each other. Towards this goal, high-throughput experimental techniques (e.g., yeast two-hybrid [12, 14] and co-immunoprecipitation [11]) to discover protein-protein interactions (PPIs) are being used. These techniques have also been supplemented by promising new computational approaches [27, 24, 23, 26, 17, 29, 9] to PPI prediction, resulting in an explosive growth in available PPI data. A powerful way of representing and analyzing all this data is the PPI network: a network where each node corresponds to a protein and an edge indicates a direct physical

** Corresponding Author. Also with the Department of Mathematics, MIT

interaction between the proteins. Computational analyses of these networks has already yielded valuable insights: the scale-free character of these networks and the disproportionate importance of "hub" proteins [30]; the combination of these networks with gene expression data to discern some of the dynamic character of the cell [8]; the use of PPI networks for inferring biological function [20], etc.

As more PPI data becomes available, 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. Such an analysis can identify conserved functional components across species [15]. As a phylogenetic tool, it offers a function-oriented perspective that complements traditional sequence-based methods. It also facilitates annotation transfer between species. Indeed, Bandyopadhyay *et al.* [3] have demonstrated that the use of PPI networks in computing orthologs produces orthology mappings that better conserve protein function across species.

In this paper, we explore a new approach to comparative analysis of PPI networks. Specifically, we consider the problem of finding the optimal *global* alignment between two PPI networks, aiming to find a correspondence between nodes and edges of the input networks that maximizes the overall match between the two networks. For this problem, we propose a novel pairwise global alignment algorithm, IsoRank.

1.1 Contributions

In this paper, we draw attention to the global network alignment problem and its biological importance (as distinct from local network alignment, see Sec. 1.2). We propose IsoRank— an algorithm for pairwise global network alignment of PPI networks; to the best of our knowledge, it is the first such algorithm of its kind. It simultaneously uses both PPI network data and sequence similarity data to compute the alignment, the relative weights of the two data sources being a free parameter (existing *local* network alignment algorithms have typically not provided such direct control over the relative weights). The algorithm is intuitive: a node i in G_1 is mapped to a node j in G_2 if the neighborhood topologies of i and j are similar, i.e., the neighbors of i can be well-mapped to the neighbors of j. This approach has parallels to Google's PageRank technique; like the latter, we formalize our intuition as an eigenvalue problem (see Sec. 3). IsoRank is, by design, tolerant to errors in the input (e.g., missing or spurious edges) and takes advantage of edge confidence scores as well as other biological signals (e.g. sequence similarity scores), when available. We use the algorithm to compute a global alignment of the *S. cerevisiae* and *D. melanogaster* PPI networks and describe the conserved subgraph (possibly disconnected) between them. The conserved subgraph immediately suggests functions for some hitherto unannotated proteins. It also suggests sets of functional orthologs between the two species; these predictions are consistent with those of Bandyopadhyay *et al.* [3], and, in some cases, are more precise and accurate.

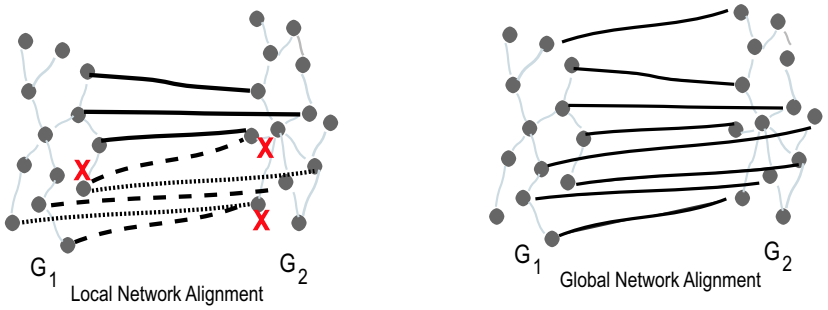


Fig. 1: **Cartoon comparing global and local network alignments**: The local network alignment between G_1 and G_2 specifies three different alignments; the mappings for each are marked by a different kind of line (solid, dashed, dotted). Each alignment describes a small common subgraph. Local alignments need not be consistent in their mapping— the points marked with 'X' each have ambiguous/inconsistent mappings under different alignments. In global network alignment, the maximum common subgraph is desired and it is required that the mapping for a node be unambiguous. In both cases, there are 'gap' nodes for which no mappings could be predicted (here, the nodes with no incident black edges are such nodes).

1.2 Related Work: the Distinction Between Local and Global Alignment

The network alignment problem has been formulated previously [6, 18, 15], with some variations. To place our work in that context, we first distinguish between global and local network alignment.

Each input network can be represented as an undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. Furthermore, G may be a weighted graph, i.e., a confidence measure $w(e)$ may be associated with each edge e in E. In this paper, we consider graphs of arbitrary structure; when graphs have specific structures (e.g., trees) other efficient methods are available [13, 28]. The goal in network alignment is to identify one or multiple possible mappings between the nodes of the input networks and, for each mapping, the corresponding set of conserved edges. Mappings may be partial, i.e., they need not be defined for all the nodes in the networks. Each mapping implies a common subgraph between the two networks: when protein a_1 from network G_1 is mapped to protein a_2 from network G_2, then a_1 and a_2 refer to the same node in the common subgraph; the edges in the common subgraph correspond to the conserved edges. Based on the kind of mapping(s) sought, we distinguish between the local and global network alignment (in analogy with sequence alignment).

Local Network Alignment (LNA): The goal in LNA is to find local 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, see Fig 1). This may not be undesirable (e.g., it may indicate gene duplication); however, in some cases LNA algorithms offer implausibly numerous matches for a single protein. The motivations behind local sequence alignment and local network alignment are analogous— the former is often used to find conserved sequence motifs; the latter for finding conserved functional components (e.g., pathways, complexes, etc.).

Previous work on PPI network alignment has almost exclusively focused on this problem: the pioneering work of Kelley *et al.* [6] described how BLAST similarity scores and PPI network information could be used to identify conserved functional motifs. Koyuturk *et al.* [18] proposed another method, motivated by biological models of duplication and deletion. Recently, Flannick *et al.* [15] proposed a new approach, using modules of proteins to infer the alignment. The approach is efficient and is the first LNA method to align multiple species simultaneously. In contrast to these methods, our work targets the global network alignment problem (see Footnote 3).

Global Network Alignment (GNA): The aim in GNA is to find the best overall alignment between the input networks. A GNA algorithm must define a single mapping across all parts of the input (see Fig 1), even if it were locally suboptimal in some regions of the networks. In contrast, an LNA algorithm has the freedom to choose the locally optimal mapping for each local region of similarity, even if this results in overlapping — and mutually inconsistent — local alignment. We avoid this in GNA by requiring that for any global alignment to be valid the corresponding mapping be *comprehensive*: each node in an input network is either matched to some node in the other network or explicitly marked as a gap node (i.e., with no match in the other network). Our goal in GNA then is to find a comprehensive mapping such that the size of the corresponding common subgraph is maximized. The motivations behind global sequence alignment and GNA are again analogous: the former is often used for comparing genomic sequences to understand variations between species; the latter may be used to compare interactomes, and to understand cross-species variations. Also, the GNA problem is related to the detection of functional orthologs, as we discuss in Sec. 4.

The GNA problem, as we describe it here, is the focus of this paper. It has previously received little attention in the literature; much of existing work has focused on the LNA problem[3]. One can imagine using results of an LNA to estimate a global alignment: use LNA methods to compute possible matches for

[3] We note that in some previous works on network alignment, the distinction between "global" and "local" network alignment has centered on the relative input sizes for each. There, the term "global network alignment" is used when the input consists of roughly equal-sized networks (e.g., two species-wide networks) while "local network alignment" is used when one input is a small query network and the other is a large species-wide network. In both instances, however, the output consists of multiple local subgraphs (and corresponding local alignments). As such, we believe that both these instances are best characterized as local network alignments, regardless of input sizes.

each protein. Then, for each protein select the mapping best supported overall by the alignment results. Banydopadhyay *et al.* have used a similar approach for functional ortholog detection. 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.

2 Problem Formulation

The input to the algorithm consists of two PPI networks G_1 and G_2. Each edge e may have an associated edge weight $w(e)$ $(0 < w(e) \leq 1)$. In addition, other measures of similarity between the nodes may be available. In this paper, we use BLAST similarity scores, but additional measures (e.g., synteny-based scoring, functional similarity) can be incorporated.

The desired output, given only PPI network data, is the 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. To address these issues, we formulate an eigenvalue problem that approximates the desired objective.

The "at most one match per node" constraint is motivated by analogy with two-way global *sequence* alignment where any position in a sequence can be matched to at most one position in the other sequence. When performing LNA, Kelley *et al.* [6] have imposed a similar constraint. The benefits of imposing this constraint are: (1) we simplify the alignment problem, and (2) we can *unambiguously* identify the closest functional equivalent of a protein in the other species; this is related to the discovery of functional orthologs (see Sec. 4). On the other hand, in instances of gene duplication across species this constraint requires that a protein cannot be matched to multiple proteins in another species. In future work, we plan to relax this constraint.

3 Algorithm: IsoRank

The key problem that our algorithm (IsoRank) targets is identifying the node mappings between the input networks; given such a mapping, the set of conserved edges can be easily computed. The algorithm works in two stages. It first associates a score with each possible match between nodes of the two networks. Let R_{ij} be the 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}s). The second stage constructs the mapping for the GNA by extracting from R high-scoring, pairwise, mutually-consistent matches.

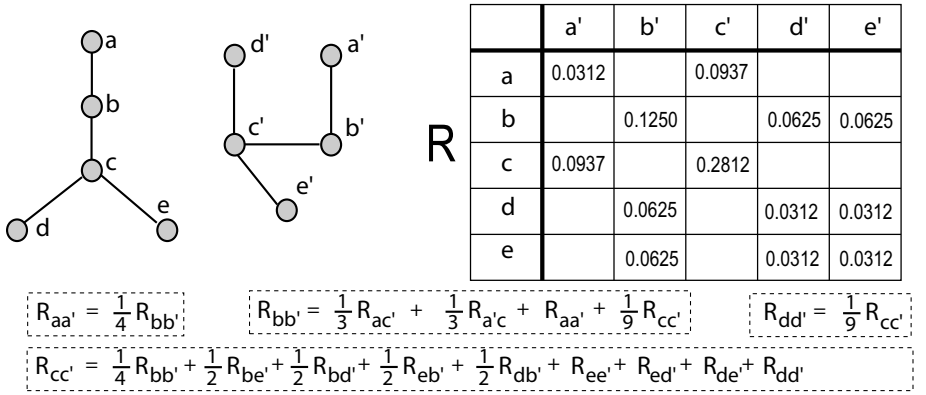


Fig. 2: **Intuition behind the algorithm**: Here we show, for a pair of small, isomorphic graphs how the vector of pairwise scores (R) is computed. For each possible pairing (i, j) between nodes of the two graphs, we compute the score R_{ij}. The scores are constrained to depend on the scores from the neighborhood as described by Eqn. 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 using the 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: 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: $\{(c, c'), (b, b'), (a, a'), (d, d'), (e, e')\}$. The $\{d, e\} \rightarrow \{d', e'\}$ mapping is ambiguous; using sequence information, such ambiguities can be resolved.

Computing R (setting up the constraints): To compute R_{ij} we pursue the intuition that (i, j) is a good match if i and j's respective neighbors also match well with each other. More precisely, we require the following equality to hold for all possible pairs (i, j):

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

where $N(a)$ 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 match (u, v) must distribute back its entire score R_{uv} equally among the $|N(u)||N(v)|$ possible matches between its neighbors. We note that these equations also capture non-local 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 now in

proportion to the edge weights:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i,u)w(j,v)}{\sum_{r \in N(u)} w(r,u) \sum_{q \in N(v)} w(q,v)} R_{uv} \quad i \in V_1,\ j \in V_2 \quad (2)$$

Clearly, Eqn. 1 is a special case of Eqn. 2 when all the edge weights are 1. We can rewrite Eqn. 1 in matrix form (Eqn. 2 can be similarly rewritten):

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

where 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).

Another interpretation of the above equations is that they describe a random walk on the product graph of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. We define $G^* = (V^*, E^*)$ where $V^* = V_1 \times V_2$ and $E^* = \{(\,(i,j),(u,v)\,) \mid (i,u) \in E_1,\ (j,v) \in E_2\}$. Also, if G_1 and G_2 are weighted, so is G^*: $w(\,(i,j),(u,v)\,) = w(i,u)w(j,v)$. We now specify a random walk among the nodes of G^*: from any node we can move to one of its neighbors, with a probability proportional to the edge weight:

$$P(s_t = (i,j) \mid s_{t-1} = (u,v)) = \frac{w(i,u)w(j,v)}{\sum_{r \in N(u)} w(r,u) \sum_{q \in N(v)} w(q,v)} \quad (4)$$

where s_t is the node occupied at time t. Eqns. 1, 2 and 3 can now be interpreted as defining R to be the stationary distribution of this random walk (its transition matrix is A). Thus, a high R_{ij} implies that the node (i,j) of G^* has a high probability of being occupied in the stationary distribution.

The vector R is determined by finding a non-trivial solution to these equations (a trivial solution is to set all R_{ij}s to zero). In Fig 3, 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 Eqn. 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. Also, for numerical stability purposes we require that R be normalized, i.e., $|R|_1 = 1$. In the case of biological networks, A is typically a very large matrix (about $10^8 \times 10^8$ 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* [16], an iterative technique often used for large eigenvalue problems. The power method repeatedly updates R as per the update rule: $R(k+1) \leftarrow AR(k)/|AR(k)|$, 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 provably converge to the principal eigenvector; the convergence can be sped up significantly by a judicious choice of the initial value $R(0)$ [16]. As we describe shortly, a good initial value $R(0)$ is often available in our case.

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}s need not even be numeric— they can be binary. Let B be the vector of B_{ij}s. We first normalize B: $E = B/|B|$. The eigenvalue equation is then modified to

$$R = \alpha AR + (1 - \alpha)E \quad \text{where} \quad 0 \leq \alpha \leq 1. \tag{5}$$

Eqn. 5 is solved by similar techniques as Eqn. 3. Also, node matches based purely on sequence similarity are an approximation to the node mappings desired; hence, the vector E is a good choice for the initial value $R(0)$ in the power method. We emphasize that this choice of starting value does not change the final value of R— it just speeds up the computation.

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, while $\alpha = 1$ indicates only network data will be used. Tuning α allows us to analyze the relative importance of PPI data in finding the optimal alignment.

Extracting the mapping from R: Once R has been computed, we extract the node mappings from it. An appealing approach is to extract the set of mutually-consistent, pairwise matches (p, q) such that the sum of their scores is maximized. The optimal solution can thus be found efficiently by interpreting R as encoding a bipartite graph and finding the maximum-weight bipartite matching [22] for this graph. Each side of the bipartite graph contains all the nodes from one network. The weight of the edge (i, j) is then set to R_{ij}. We compute the maximum-weight matching in this bipartite graph and output the paired nodes. Any remaining unpaired nodes are designated as gap nodes. This algorithm guarantees the set of matches that satisfy our criterion.

While this principled algorithm does give good results, in practice we found that the following greedy algorithm sometimes performs even better: identify the highest score R_{pq} and output the pairing (p, q). Then, remove all scores involving p or q. We then repeat this process until the list is empty. In the bipartite graph, this strategy corresponds to removing, at each step, the maximum weight edge and the incident nodes. In future work, we plan to investigate whether this heuristic's better performance is related to the structure of R.

Once a comprehensive alignment has been computed, the corresponding 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 (a_1, a_2) and (b_1, b_2) if and only if both the input networks contain supporting edges (i.e., (a_1, b_1) in G_1 and (a_2, b_2) in G_2). When edges also have associated weights, formalizing the intuition depends on how the edge weights are being interpreted; for example, we could require that the combined weight be higher than a threshold or that the minimum of the two be greater than a threshold.

4 Results: GNA of Yeast and Fly PPI Networks

We now describe the results of two-way global alignment of the *S. cerevisiae* and *D. melanogaster* PPI networks, the two species with the most available network

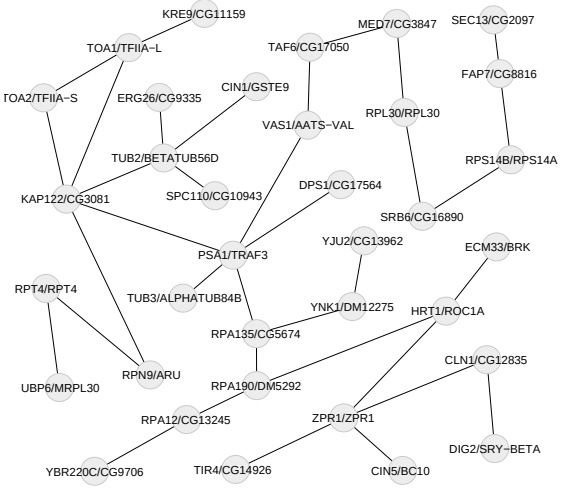
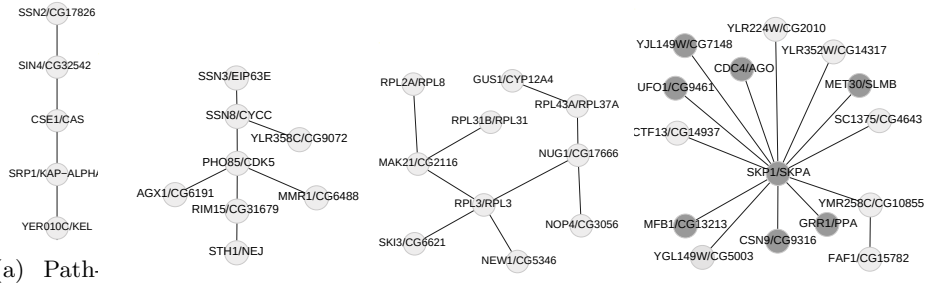


Fig. 3: **Largest connected component of the yeast-fly Global Network Alignment:** The node labels indicate the corresponding "yeast/fly" proteins (the two separated by a "/"). The proteins in this graph span a variety of functions: metabolic, signaling, transcription etc. For a discussion of this subgraph's size, see text.

data. The PPI network data for the species was retrieved from the GRID [4] and DIP [7] databases, and the sequence data was retrieved from Ensembl [2]. The edges in the PPI networks did not have associated weights. We applied IsoRank to this pair of networks, using it to identify the common subgraph.

The common subgraph corresponding to the global alignment between the yeast and fly PPI networks has 1420 edges (where $\alpha = 0.6$; the criterion for choosing α is described later in this section). While this indicates a relatively low overlap between the yeast and fly networks (both the networks have more than 25000 edges each), it is not surprising: firstly, currently available PPI data is known to contain many false-positives, and the number of true interactions in the current networks is expected to be significantly lower [27, 25]. Secondly, current PPI data is far from comprehensive; e.g., the fly network has no known PPIs for about 6500 proteins (almost 50% of the genome). As these issues get resolved, we expect the size of the global alignment to grow substantially. Nevertheless, the current global alignment already provides many valuable insights.

The alignment subgraph consists of many disconnected components, with the largest component having 35 edges (Fig. 3). The component's size may seem low but is directly related to the poor connectivity of the alignment subgraph. The poor connectivity is, we believe, because of the poor quality and coverage of current PPI networks; as the datasets improve, so will the connectivity. Even now, however, the subgraph in Fig. 3 is significantly larger than any common subgraph we could identify using Pathblast [6], a LNA method. The longest pathway-like component identified by the latter had 4 nodes, and the largest complex-like component had 16 nodes. Also, the components of the global alignment span various topologies, from linear pathways (Fig. 4(a)) to components



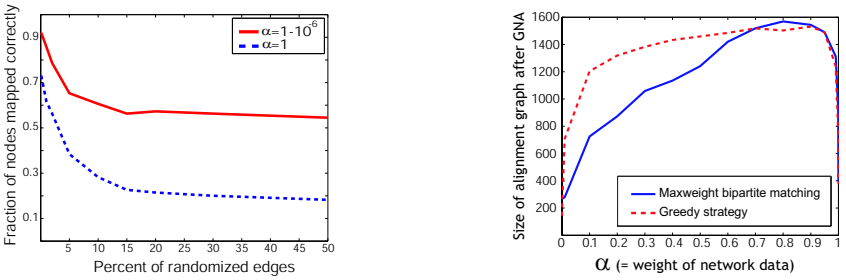
(a) Pathway (b) Kinases (c) Ribosome Complex (d) Ubiquitin Ligase

Fig. 4: **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 (c) and (d). In (d), the nodes for which at least one of the corresponding proteins is known to be involved in ubiquitin ligase activity are shaded.

corresponding to protein complexes (Fig 4(d)); in contrast, some of the local network alignment methods [6, 18] are tailored to search only for specific topologies. We emphasize that our components were discovered simultaneously— they are just subgraphs of the larger alignment graph. Many of our discovered components are de-facto *functional modules* (though not in the sense Flannick *et al.* [15] use the term): they are enriched in proteins involved in a single biological process (e.g., see Fig 4(d)). These functions range from various signaling cascades (Fig. 4(b)) to core cellular functions like ribosomal synthesis and function (Fig. 4(c)), DNA transcription and translation, cell division etc. The preponderance of core cellular functions in the conserved subgraph is not too surprising— it is exactly these mechanisms that are likely to be highly conserved across species.

The global alignment may be used to predict protein function. For example, Fig 4(d) shows a subgraph of the global alignment, most of the proteins in which are involved in SCF ubiquitin ligase activity. Hence, we predict the function of two hitherto-unannotated fly proteins CG7148 and CG13213 as being involved in ubiquitin protein ligase activity. In support of this, we note that the FlyBase database [5] indicates that the involvement of these proteins in ubiquitin ligase activity has been postulated before in the literature. Of course, more sophisticated methods to transfer annotation may perform even better at elucidating function of such proteins [20].

Evaluating the algorithm's error tolerance: Our simulations indicate that the algorithm is tolerant to error in the input (Fig 5(a)); this is valuable since PPI networks have high false positive and false negative rates. To evaluate the algorithm's error-tolerance, we first extracted a 200-node subgraph of the yeast PPI network. We then randomized a fraction p of its edges using the Maslov-Sneppen trick that preserves node degrees [19]: we randomly choose two edges (a, b) and (c, d), remove them, and introduce new edges (a, d) and (c, b). We



(a) Error-Tolerance (b) α

Fig. 5: **(a) Effect of error on algorithm's performance:** We believe the solid (red) curve slightly overestimates the algorithm's performance, while the dashed (blue) curve grossly underestimates it. See the discussion in text below. **(b) Impact of α on the size of the alignment graph.**

then computed a GNA between these two graphs, with $\alpha = 1$ and $\alpha = 1 - 10^{-6}$. For each choice of p, we created 5 such randomized graphs and computed the average fraction of nodes that are mapped to themselves in the original graph after a GNA. Using $\alpha = 1$ results in a significant underestimate because there often are multiple possible isomorphism-preserving mappings between two isomorphic graphs (e.g., see Fig 3) and our algorithm— even if working correctly— might choose a mapping that does not preserve node labels. Adding a very small amount of sequence information ($\alpha = 1 - 10^{-6}$) helps avoid this, but also results in a slight overestimate. We believe the true curve (for Fig 5(a)) is closer to the top curve than the bottom one. Clearly, the algorithm makes very few mistakes when the error rate p is low and even for fairly high error rates (20-50%), its performance degrades smoothly and very slowly. When computing the yeast-fly GNA, we assigned a significant weight to sequence information ($\alpha = 0.6$); these simulations suggest our results are quite robust to errors in PPI data.

Evaluating the influence of α: As α increases, so does the importance of network data in the alignment process, for both the greedy strategy and the maximum weight bipartite matching strategy (Fig 5(b)). In line with our expectations, the size of the common subgraph depends on this parameter: $\alpha = 0$ results in a graph with 266 edges, while $\alpha = 0.9$ results in 1544 edges (for the greedy strategy). Intriguingly, as α gets very close to 1, the common graph's size *decreases*. We believe that this discrepancy is an artifact of the current PPI data sets being noisy and covering the interactome only partially, resulting in a relatively small overlap between the yeast and fly PPI networks. Consequently, in absence of any other information a random mapping of nodes between the two networks might satisfy Eqn.1 better than the one corresponding to the "true" alignment. The use of sequence-based scores helps mitigate this, by directing the algorithm towards the true alignment.

When choosing the most appropriate value of the free parameter α, we rejected the choice corresponding to the largest common subgraph size— the input

networks are noisy and conserved edges may be simply due to noise; thus, the α leading to the largest-size subgraph may not be a biologically appropriate choice. Instead, for each choice of α, we compared the resulting node mappings to sequence-based ortholog predictions from the Inparanoid database [21] and chose the α ($= 0.6$) that resulted in the greatest overlap with these. While this approach is conservative and might undervalue the network component during the alignment, it also lowers the adverse impact of noise in the PPI data.

The differences between the node pairings found by our algorithm and those from Inparanoid broadly fall into two categories: (1) those corresponding to low R_{ij} values indicating low confidence of our approach in that mapping, and (2) functional orthologs where the use of network data genuinely changes the node mapping. We discuss the latter in more detail later in this section.

Comparing global and local alignment results: Our global alignment results compare favorably to the those of NetworkBlast [1] (an implementation of PathBlast) and sequence-only approaches. We compared the aggregate set of local alignments from NetworkBlast with our global alignment. Each local alignment defines one-to-one matches between some yeast and fly proteins. Many of the matches from our global alignment are seen in these local alignments: of the 701 matched protein-pairs in the former that consist of proteins seen in at least one local alignment, 83% (582) of the pairs are also observed in one or more local alignments. However, there are many overlapping local alignments, resulting in ambiguity and inconsistency: averaged across the entire set of local alignments, a yeast protein is aligned to 5.36 different fly proteins. Sometimes, such ambiguity may be biologically meaningful, e.g., in instances of gene duplication. However, the degree of ambiguity in some of the PathBlast results is clearly implausible. For example, the yeast protein SNF1, a Serine-Threonine Kinase (STK), is matched to 71 different fly proteins. In fact, PathBlast results for many of the yeast STKs are very ambiguous– over the set of 72 yeast proteins annotated as STKs, the average number of matching fly proteins per yeast STK is 29.3. STKs are part of many important signaling pathways, e.g, the MAPK, JNK and AKT cascades. Sequence-only approaches. (e.g. Inparanoid) too have performed poorly at ascertaining the correspondence between yeast and fly STKs: Inparanoid does not predict any fly orthologs for 58 of the 72 yeast STKs. Thus the use of GNA to resolve this ambiguity in correspondence is particularly valuable.

GNA and functional orthologs: In analogy with sequence-based comparative genomics methods [10], we apply IsoRank to the detection of functional orthologs (i.e., sets of proteins that perform the same function in two or more species) by exploiting the strong connection between these two problems: proteins that are aligned together in the global alignment should have similar interaction patterns in their respective species and are thus likely to be functional orthologs. There has been a lot of recent interest in the discovery of functional orthologs (FO). In particular, Bandyopadhyay *et al.* [3] took a fairly complex approach to FO detection between yeast and fly through local network alignment

(LNA): first, possible FOs for a protein are short-listed using a sequence-only approach; then, using a probabilistic technique (based on Markov Random Fields) and the results of a LNA of the yeast and fly networks (performed using PathBlast), the probability of each short-listed pair of proteins being true FOs is computed.

The results of IsoRank compare favorably with Bandyopadhyay *et al.*'s. 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 1), i.e, FOs predicted by our method are also the likely FOs predicted by their method. Furthermore, their method often proposes multiple FOs for a protein, and our method resolves the ambiguity in their results. 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). Bandyopadhyay *et al.*'s method is ambiguous here as *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.

5 Conclusion

In this paper, we focus on the global network alignment problem, and describe an intuitive yet powerful algorithm for computing the global alignment of two PPI networks; in contrast, much of the previous work has been focused on the local alignment problem. Our algorithm, IsoRank, simultaneously uses network and sequence information and is tolerant of noise in the inputs; furthermore, it is easy to control the relative weights of the network and sequence information in the alignment. We use IsoRank to compute a global alignment of the *S. cerevisiae* and *D. melanogaster* PPI networks. The results provide valuable insights about the conserved functional components between the two species. They also allow us to predict functional orthologs between the fly and yeast; the quality of our predictions compare favorably with previous work.

Our algorithm is similar— in spirit— to Google's PageRank algorithm, which ranks web-pages in the order of their "authoritativeness". The intuition behind the two algorithms has a similar flavor: in PageRank, a page has a high score if many pages with high scores link to it. The intuitions are also formalized similarly– by constructing an eigenvalue problem. Our actual algorithm is quite distinct from PageRank: in our case the input is a pair of undirected, weighted graphs and the output is an alignment; PageRank's input is a directed, unweighted graph (where the nodes indicate web-pages and directed edges, hypertext links), and it outputs node rankings.

Protein (species)	**Predicted Functional Ortholog by Our Method**	**Related Predictions from (Bandyopadhyay et al.)**		**Remarks**
		Yeast/Fly pair	**Prob.**	
Gid8 (yeast)	CG6617	Gid8/CG6617	76.51%	Our predictions consistent with Bandyopadhyay et al.[1]
		Gid8/CG18467	-	
Tpm2 (yeast)	Tm1	Tpm2/Tm1	-	Consistent predictions.[1]
Tpm1 (yeast)	Tm2	Tpm1/Tm2	43.98%	Consistent predictions.[1]
Gpa1 (yeast)	G-oα47a	Gpa1/G-oα47a	41.53%	Consistent predictions.[1]
		Gpa1/G-ia65a	-	
Rpl12 (fly)	Rpl12a	Rpl12a/Rpl12	48.39%	Consistent predictions.[1]
		Rpl12b/Rpl12	-	
Btt1 (yeast)	CG11835	Btt1/CG11835	70.5%	Consistent predictions.[1]
		Btt1/Bcd	40.86%	
CG18617 (fly)	Vph1	Vph1/CG18617	43.53%	Consistent predictions.[1]
		Stv1/CG18617	38.44%	
Kap104 (fly)	Trn	Kap104/Trn	40.64%	Partially consistent predictions.[2]
		Kap104/CG8219	46.78%	
Act1 (yeast)	Act5c	Act1/Act5c	39.56%	Partially consistent predictions.[2]
		Act1/Act42a	39.24%	
		Act1/Act87e	43.53%	
		Act1/Act88f	40.17%	
		Act/CG10067	38.20%	
Kel2 (yeast)	CG12081	Kel2/CG12081	-	Partially consistent predictions.[2]
		Kel1/CG12081	45.41%	
Cmd1 (yeast)	Cam	Cmd1/Cam	35.90%	Partially consistent predictions.[2]
		Cmd1/And	44.39%	
Hsc70-4 (fly)	Ssa3	Hsc70-4/Ssa3	-	Partially consistent predictions.[2]

Table 1: **Interpreting two-way global alignment results as functional orthologs (FOs)**: Comparison of our results with Bandyopadhyay *et al.*'s results [3]. Our method is often consistent with their results and, moreover, often resolves the ambiguity in their predictions. [1]Our predicted FO for the protein matches Bandyopadhyay *et al.*'s predicted FO, or the most likely FO if their method predicted multiple FOs.[2]Our predicted FO for the protein is one of the likely FOs predicted by Bandyopadhyay *et al.* (but not the most likely one).

We have already extended IsoRank to perform global alignment of multiple networks, but this is beyond the scope of this paper. In future work, we plan to improve the algorithm, better characterize its theoretical behavior, and identify other applications for it. Since PPI data is noisy, it might be useful to generate multiple near-optimal alignments and rank them by their significance. Also, the algorithm can be applied to other biological and non-biological data. It might also be possible to extend such an eigenvalue approach to perform local network alignment; as noted before, the use of an eigenvalue approach removes

the restriction of being able to find subgraphs with only certain topologies– a limitation of some of the existing local network alignment methods.

References

1. *http://chianti.ucsd.edu/NetworkBlast.*
2. *http://www.ensembl.org.*
3. S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome Res*, 16(3):428–35, 2006.
4. B.J. Breitkreutz, C. Stark, and M. Tyers. The GRID: the general repository for interaction datasets. *Genome Biology*, 4(3):R23, 2003.
5. FlyBase Consortium. The FlyBase database of the drosophila genome projects and community literature. *Nucleic Acids Res*, 31(1):172–175, 2003.
6. B.P. Kelley et al. Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Res*, 32(Web Server issue):W83–8, 2004.
7. I. Xenarios et al. DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30(1):303–305, 2002.
8. J.D. Han et al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995):88–93, 2004.
9. J.P. Miller et al. Large-scale identification of yeast integral membrane protein interactions. *Proc Natl Acad Sci USA*, 102(34):12123–12128, 2005.
10. M. Kellis et al. Methods in comparative genomics: genome correspondence, gene identification and regulatory motif discovery. *J of Computational Biology*, 11(2-3):319–355, 2004.
11. N.J. Krogan et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–43, 2006.
12. P. Uetz et al. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–7, 2000.
13. R.Y. Pinter et al. Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408, 2005.
14. T. Ito et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci USA*, 98(8):4569–74, 2001.
15. J. Flannick, A. Novak, B.S. Srinivasan, H.H. McAdams, and S. Batzoglou. Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res*, 16(9):1169–81, 2006.
16. G.H. Golub and C. Van Loan. Matrix computations. *Johns Hopkins University Press)*, 2006.
17. I. Gat-Viks, A. Tanay, D. Raijman, and R. Shamir. A probabilistic methodology for integrating knowledge and experiments on biological networks. *J of Computational Biology*, 13(2):165–181, 2006.
18. M. Koyuturk, A. Grama, and W. Szpankowski. Pairwise local alignment of protein interaction networks guided by models of evolution. *Proc of the 9^{th} International Conference on Research in Computational Molecular Biology (RECOMB)*, 2005.
19. S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002.
20. E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21 Suppl 1:i302–10, 2005.

21. K.P. O'Brien, M. Remm, and E.L. Sonnhammer. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res*, 33(Database issue):D476–80, 2005.
22. C. Papadimitriou and K. Steiglitz. Combinatorial optimization: algorithms and complexity. *Dover)*, 1998.
23. Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph. Random forest similarity for protein-protein interaction prediction from multiple sources. *Proc of the Pacific Symposium on Biocomputation*, 2005.
24. R. Singh, J. Xu, and B. Berger. Struct2net: Integrating structure into protein-protein interaction prediction. *Proceedings of the Pacific Symposium on Biocomputation*, 2006.
25. D. Sontag, R. Singh, and B. Berger. Probabilistic modeling of systematic errors in yeast two-hybrid experiments. *To Appear. Proceedings of the Pacific Symposium on Biocomputation*, 2007.
26. B.S. Srinivasan, A. Novak, J. Flannick, S. Batzoglou, and H. McAdams. Integrated protein interaction networks for 11 microbes. *Proc of the 10^{th} International Conference on Research in Computational Molecular Biology(RECOMB)*, 2006.
27. C. von Mering et al. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.
28. M.Y. Yao, T.W. Lam, and H.F. Ting. An even faster and more unifying algorithm for comparing trees via unbalanced bipartite matchings. *J of Algorithms*, 40:212, 2006.
29. C.H. Yeang and M. Vingron. A joint model of regulatory and metabolic networks. *BMC Bioinformatics*, 7:332, 2006.
30. S.H. Yook, Z.N. Oltvai, and A.L. Barabasi. Functional and topological characterization of protein interaction networks. *Proteomics*, 4(4):928–42, 2004.