Identifying Evolutionarily Conserved Protein Interaction Modules Using GraphHopper

Corban G. Rivera* and T.M. Murali**

114 McBryde Hall, Department of Computer Science,
Virginia Polytechnic Institute and State University, Blacksburg VA 24061
cgrivera@vt.edu, murali@cs.vt.edu

Abstract. We study the question of detecting Conserved Protein Interaction Modules (CPIMs) in protein-protein interaction (PPI) networks. We propose a novel algorithm called GraphHopper that analyzes two PPI networks to find CPIMs. GraphHopper finds CPIMs by "hopping" from one network to another using orthology relationships. By decoupling the degree of evolutionary conservation in a CPIM from the reliability of the PPIs in a CPIM, GraphHopper finds CPIMs with a wide variety of topologies that previous algorithms cannot detect.

GraphHopper is competitive with NetworkBlast and Match-and-Split, two state-of-the-art algorithms for computing CPIMs, on the task of recapitulating MIPS processes and complexes. Upon applying GraphHopper to human, fly, and yeast PPI networks, we find a number of CPIMs involved in fundamental processes of the cell that are conserved in all three species. We present the first global map of human-fly CPIMs. This map sheds light on the conservation of protein interaction modules in multi-cellular organisms. CPIMs related to development and the nervous system emerge only in the human-fly comparison. For example, a set of 10 interconnected CPIMs suggest that fly proteins involved in eye development may have human orthologs that have evolved functions related to blood clotting, vascular development, and structural support.

1 Introduction

Protein-Protein Interaction (PPI) networks containing thousands of interactions are now available for a number of species, including human, yeast, worm, and fly. Pairwise comparison of these networks enables the computation of groups of interacting proteins that are conserved in different organisms [1], thus laying the basis for module-level modeling of cellular processes. Such conserved sets consist of two connected protein interaction sub-networks (or modules), one in each PPI network, such that proteins in each module have orthologs in the other module. In this paper, we call these *Conserved Protein Interaction Modules* (CPIMs).

* Current address: Johns Hopkins University, School of Medicine, 720 Rutland Ave, Traylor 613, Baltimore, MD 21205-2109.
** Corresponding author.

S. Rajasekaran (Ed.): BICoB 2009, LNBI 5462, pp. 67–78, 2009.

Sharan and Ideker [1] survey many techniques developed to address this problem. A common feature of a number of these approaches [2,3,4] is the combination of the PPI networks of two species into a single "alignment graph". A node in the alignment graph represents two orthologous proteins, one from each PPI network. An edge in the alignment graph represents an interaction that is conserved in both PPI networks. These methods add an edge to the alignment graph only if the proteins contributing to the nodes are connected through at most one intermediate protein in the respective PPI networks. The weight of an edge represents the likelihood that the corresponding interactions are conserved; this weight depends on the degree of orthology between the proteins and on assessed confidence estimates that the individual PPIs indeed take place in the cell. These authors find CPIMs by using various approaches to compute paths, complexes, and subgraphs of high weight in the alignment network and then expanding each such subgraph into the constituent PPIs.

Our approach. In this paper, we present a novel algorithm called GraphHopper for computing CPIMs in two PPI networks. GraphHopper computes two scores of quality for each CPIM. (i) The *conservation score* measures the total amount of sequence similarity among the proteins in the CPIM, averaged over the number of proteins in the CPIM. (ii) The *unreliability score* measures our total confidence in all the PPIs in the CPIM; this measure is useful since it is well documented that a number of high-throughput assays for detecting PPIs have high error rates [5]. A "good" CPIM has high conservation score and low unreliability score. Unlike the techniques mentioned earlier, GraphHopper treats the two PPI networks separately and connects each node in one PPI network to its potential orthologs in the other PPI network. GraphHopper starts by constructing a number of basis CPIMs, each of which is a pair of orthologous protein-pairs that directly interact. GraphHopper then expands each basis CPIM into a CPIM by "hopping" from one PPI network to another. In each hop, GraphHopper adds proteins and interactions to the current CPIM while ensuring that (i) the conservation score does not decrease and (ii) the unreliability score increases as little as possible. GraphHopper stops when it cannot add any more proteins without decreasing the conservation score.

Like GraphHopper, Narayanan and Karp's Match-and-Split algorithm [6] does not construct an alignment network. They use combinatorial criteria to decide when the local neighborhoods of a pair of orthologs match. Under their model, they prove that a given pair of proteins can belong to at most one CPIM. This observation leads to a top-down partitioning algorithm that finds all maximal CPIMs in polynomial time. The MULE algorithm developed by Koyuturk et al. [7] also keeps PPI networks separate; it uses ortholog contraction and frequent subgraph detection to identify CPIMs.

Our contributions. We used GraphHopper to analyze all pairwise combinations of human, fly, and yeast PPI networks. Other approaches have considered the conservation of human PPIs and CPIMs in the networks of other eukaryotes [3,7,8]. The primary contribution of our work is a significant expansion of

these results by (i) considering a dataset of human PPIs integrated from multiple sources, (ii) detecting large functionally-enriched CPIMs with diverse topologies, and (iii) computing an integrated high-level map of CPIMs conserved only in human and fly PPI networks. As far as we know, this paper is the first to construct such a high-level map of human-fly CPIMs. Modules found by Gandhi et al. [8] were restricted to fundamental processes of life such as DNA replication and repair and transcription. In contrast, we find many CPIMs with that are enriched in functions unique to multi-cellular organisms. For instance, we find a set of 10 interconnected CPIMs which suggest that fly proteins involved in eye development may have human orthologs that have evolved functions related to blood clotting, vascular development, and structural support.

We compared GraphHopper to NetworkBlast [9], a state-of-the-art algorithm based on alignment networks, and to Match-and-Split [6]. We measured the ability of these three algorithms to recover MIPS complexes and processes [10]. In general, GraphHopper is competitive with and sometimes outperforms Match-and-Split in spite of computing a much larger number of CPIMs. GraphHopper has better precision and recall than NetworkBlast for MIPS processes.

An important feature of GraphHopper is its ability to compute CPIMs of far more diverse topologies than algorithms based on alignment networks. Algorithms that operate on alignment networks compute (highly weighted) subgraphs and map them into modules in each PPI network being compared. Since each such module is likely to have a topology very similar to the subgraph in the alignment network, the modules themselves have very similar topologies. In contrast, GraphHopper keeps the two PPI networks separate, thereby decoupling the evolutionary conservation of the proteins in a CPIM from the reliability of the PPIs that connect the proteins. As a result, GraphHopper is able to adapt to differing patterns of interactions in the two PPI networks, e.g., matching a module with one topology (say, a star) in one PPI network to a module with a considerably different topology (say, a complex) in the other PPI network. GraphHopper outperforms not just NetworkBlast but also Match-and-Split in this comparison. Match-and-Split only outputs CPIMs with at most 50 proteins; this restriction may hamper its ability to find CPIMs with diverse topologies.

2 Algorithms

2.1 A Model for CPIMs

We represent the set of PPIs in an organism as an undirected graph $G(V, E)$, where V is the set of proteins in the organism and each edge $(a, b) \in E$ is an interaction between proteins a and b. We associate a weight l_e with each edge $e \in E$. Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be the PPI networks of two different organisms. We represent orthologous proteins as a bipartite graph H in which each edge $(a, b) \in V_1 \times V_2$ represents a pair of orthologous proteins a and b. Each edge $e \in H$ has a weight w_e equal to the BLASTP E-value between the connected proteins. We define a *Conserved Protein Interaction Module* (CPIM) as a triple (T_1, T_2, O) where T_1 and T_2 are connected subgraphs of G_1 and G_2,

respectively, and $O \subseteq H$ such that $(a, b) \in O$ if and only if a is a node in T_1 and b is a node in T_2. Thus, O is the subgraph of H induced by the nodes in T_1 and T_2. We define two quantities to measure the quality of a CPIM.

Conservation score. The conservation score of a CPIM (T_1, T_2, O) measures the amount of evolutionary similarity (at the amino acid level) between the protein interaction networks T_1 and T_2. Let P_1 (respectively, P_2) be the set of nodes in T_1 (respectively, T_2). We define the *conservation score* of a CPIM (T_1, T_2, O) as

$$\phi(T_1, T_2, O) = \frac{\sum_{e \in O} -\log(w_e)}{|P_1| + |P_2|}.$$

The larger this score, the more evolutionary conserved T_1 and T_2 are since fewer proteins without orthologs can belong to the CPIM.

Unreliability score. Since many experimental techniques used to detect PPIs are error-prone, a number of methods have been developed to assess PPI reliabilities [5]. We do not consider methods that use gene expression data, since our goal is detect conservation purely at the level of PPIs. We also discard techniques that use functional annotations, since we use this data to assess the biological information in a CPIM. Therefore, we compute edge weights using the method proposed by Goldberg and Roth [11]: if the two nodes incident on an edge have more common neighbors than would be expected by chance, they assigned a high confidence (low p-value) to that edge. For a PPI e, we compute this p-value p_e using Fischer's exact test and set $l_e = -\log(1 - p_e)$. We use Bonferroni's correction to adjust for multiple hypotheses testing. We define the *unreliability score* $q(T_1, T_2, O)$ of a CPIM as follows:

$$q(T_1, T_2, O) = \sum_{e \in T_1 \cup T_2} l_e.$$

Since p_e is a probability, we combine the weights of multiple PPIs by adding their logarithms (the l_e values). A CPIM with high confidence edges has a small unreliability score.

2.2 The GraphHopper Algorithm

The GraphHopper algorithm finds CPIMs with high conservation and low unreliability scores. Our inputs are two protein interaction networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a set of orthologous protein pairs H. We define the *lightness* of a path π of PPIs in G_1 or G_2 to be $|\pi| = \sum_{e \in \pi} l_e$; thus, light paths contain high-confidence edges.

Computing basis CPIMs. We start by constructing a basis set of CPIMs in which each CPIM (T_1, T_2, O) has the following properties: (i) O contains two edges $(a, a') \in H$ and $(b, b') \in H$; (ii) a and b are adjacent in G_1 (i.e., T_1 is the edge (a, b)); and (iii) a' and b' are adjacent in G_1.

Expanding a basis CPIM. GraphHopper processes each CPIM in the basis set using the following iterative algorithm. Figure 1 displays these steps.

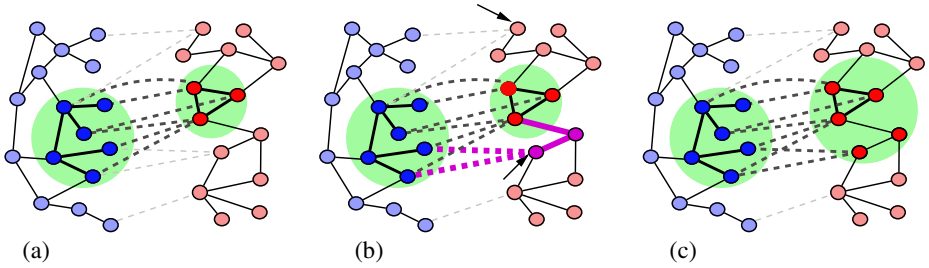


Fig. 1. An illustration of how GraphHopper expands a CPIM in iteration k. (a) A CPIM at the end of iteration $k-1$. (b) In iteration k, GraphHopper keeps the blue network in the CPIM fixed and expands the red network. Arrows mark the two nodes in the set P computed in Step (i). The node v' found in Step (iii) is the lower of these two nodes. In Steps (iv) and (v), GraphHopper adds the thick magenta PPIs and orthology edges to the CPIM. (c) The CPIM at the end of iteration k.

Let (T_1^1, T_2^1, O^1) be a basis CPIM. In iteration $k > 1$ (Figure 1 (a)), we construct a CPIM (T_1^k, T_2^k, O^k) such that $(T_1^{k-1}, T_2^{k-1}, O^{k-1})$ is a subgraph of (T_1^k, T_2^k, O^k) and $\phi(T_1^k, T_2^k, O^k) > \phi(T_1^{k-1}, T_2^{k-1}, O^{k-1})$. We also attempt to keep $q(T_1^k, T_2^k, O^k) - q(T_1^{k-1}, T_2^{k-1}, O^{k-1})$ as small as possible. We keep either T_1^{k-1} or T_2^{k-1} fixed and "expand" the other graph. Without loss of generality, we assume that $T_1^k = T_1^{k-1}$ and T_2^{k-1} is a subgraph of T_2^k in the following discussion. We construct (T_1^k, T_2^k, O^k) using the following steps:

(i) We identify a set $P \subseteq V_2$ of nodes such that each node $v \in P$ is not a node in T_2^{k-1} and is connected by an edge in H to at least one node in T_1^k.
(ii) For each node $v \in P$, we use Dijkstra's algorithm to compute the lightest path π_v in G_2 that connects v to T_2^{k-1}, i.e., for each node $u \in T_2^{k-1}$, we compute the lightest path between u and v in G_2, and set π_v to be the lightest of these paths.
(iii) We find the node v' in P such that $\pi_{v'}$ is the lightest among all paths computed in the previous step.
(iv) We set T_2^k to be the union of T_2^{k-1} and $\pi_{v'}$ (Figure 1 (b)).
(v) We set O^k to be the union of O^{k-1} and the set of edges in H incident on v' and a node in T_1^k (Figure 1 (b)).
(vi) We compute $\phi(T_1^k, T_2^k, O^k)$. If $\phi(T_1^k, T_2^k, O^k) > \phi(T_1^{k-1}, T_2^{k-1}, O^{k-1})$, we go to Step (i) and expand (T_1^k, T_2^k, O^k) while keeping T_2^k fixed (Figure 1 (c)). Otherwise, we proceed to the next basis CPIM.

We provide the rationale for these steps. To expand the CPIM $(T_1^{k-1}, T_2^{k-1}, O^{k-1})$ after setting $T_1^k = T_1^{k-1}$, we first identify the set P of nodes in G_2 that do not belong to T_2^{k-1} but are orthologs of nodes in T_1^k (Step (i)). Each node in P is a candidate that we can add to T_2^{k-1} in order to construct T_2^k. However, such a node $v \in P$ may not be adjacent to any node in T_2^{k-1}, as displayed in Figure 1 (b). Since our goal is to keep $q(T_1^k, T_2^k, O^k) - q(T_1^{k-1}, T_2^{k-1}, O^{k-1})$ as small as possible, we would like to connect v to T_2^{k-1} using the edges with the highest

possible confidence in G_2. A natural candidate for this set of edges is the lightest path π_v connecting v to T_2^{k-1}, where this minimum is taken over the set of lightest paths connecting v to each node in T_2^{k-1}. Therefore, for each node v in P, we compute the lightest path π_v by which we can connect v to T_2^{k-1} using only edges in G_2 (Step (ii)). In Steps (iii) and (iv), we add that path π'_v to T_2^{k-1} that is lightest among all the paths computed i.e., $v' = \arg\min_{v \in P} |\pi_v|$. After computing T_2^k, we set O^k to be the subgraph of H induced by the nodes in T_1^k and T_2^k by adding the edges in H that are incident on v' and any node in T_1^k (Step (v)); by construction, no node in π'_v other than v' is connected by an edge in H to a node in T_1^k. This step completes the construction of (T_1^k, T_2^k, O^k). Finally, in Step (vi), we continue expanding (T_1^k, T_2^k, O^k) if its conservation score is greater than $\phi(T_1^{k-1}, T_2^{k-1}, O^{k-1})$. Otherwise, we stop the iteration and move on to the next basis CPIM. By induction, the graphs T_1^k, T_2^k and $T_1^k \cup T_2^k \cup O^k$ are connected. Note that $q(T_1^k, T_2^k, O^k)$ implicitly plays a role in the expansion: since both the unreliability score of a CPIM and the lightness of a path are defined as the sum of the l_e values of the edges that appear in the CPIM or the path, by choosing to add the lightest path $\pi_{v'}$ to T_2^k, we are attempting to minimize $q(T_1^k, T_2^k, O^k) - q(T_1^{k-1}, T_2^{k-1}, O^{k-1})$.

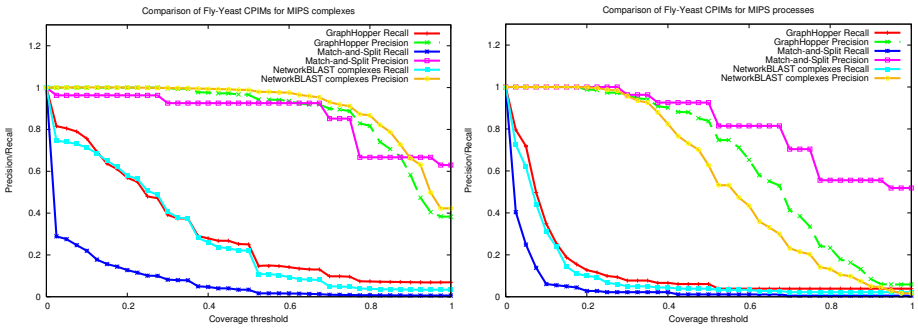
Merging CPIMs. Following Sharan et al. [9], we compute the statistical significance of a CPIM by comparing its conservation score to the distribution of conservation scores of CPIMs found by GraphHopper in random PPI and orthology networks with the same degree distributions as G_1, G_2, and H. We retain CPIMs with p-value at most 0.05. The remaining CPIMs may have considerable overlap. We merge CPIMs by modifying the procedure used by Sharan et al. [9]. For each CPIM C, we compute all the biological functions it is enriched in using Fischer's exact test and note the function f_C that is most enriched (has smallest p-value) in C. Let F be the set of all such most-enriched functions. For each function $l \in F$, we compute a CPIM C_l as the union of all CPIMs C for which $l = f_C$, i.e., $C_l = \bigcup_{l=f_C} C$. We report results for these CPIMs. Note that this method (i) does not require us to provide a cutoff on the overlap of two CPIMs that should be merged, (ii) allows merged CPIMs to share both proteins and interactions, and (iii) may yield disconnected CPIMs.

Remarks. There is considerable scope for variation in our algorithm. For instance, we can define the conservation and unreliability scores differently, combine the two scores, use simulated annealing-like techniques to optimize these scores, or focus on optimizing the unreliability score instead of the conservation score. We have experimented with a number of such choices (data not shown) and found that the algorithm presented consistently achieves good results.

3 Results

3.1 Comparison to NetworkBlast and Match-and-Split

We compared GraphHopper to NetworkBlast [9], a state-of-the-art method for computing CPIMs from alignment networks, and to Match-and-Split [6], which



(a) MIPS complexes (b) MIPS processes

Fig. 2. Comparisons of GraphHopper, Match-and-Split, and NetworkBLAST

that like GraphHopper finds CPIMS by keeping the two PPI networks separate. Both papers used the same fly and yeast datasets. We downloaded these datasets and the results obtained by these algorithms from the supplementary websites accompanying the respective papers. We ran GraphHopper on exactly the same fly and yeast datasets. We used the procedure suggested by Narayanan and Karp [6] to compare the algorithms. We computed fly-yeast CPIMs and considered only the yeast sub-network in each CPIM. We considered two sets of gold standard modules defined by MIPS process annotations and by MIPS complex annotations for yeast genes [10]. We defined one set S of proteins as being *covered by* another set S' if $|S \cap S'|/|S| \geq t$, for a threshold $0 \leq t \leq 1$.[1] For a given value of t, we measured the *precision* of an algorithm as the fraction of computed CPIMs covered by at least one gold standard module and the *recall* of the algorithm as the fraction of gold standard modules covered by at least one computed CPIM. For each algorithm, we plotted precision and recall at different values of t. Precision and recall are both equal to 1 for $t = 0$. Both measures decrease monotonically with increasing t.

Figure 2 displays these results. For MIPS complexes, all three algorithms have comparable precision across almost the entire range of the coverage threshold. However, GraphHopper and NetworkBLAST have better recall than Match-and-Split. Match-and-Split achieves the best precision for MIPS processes. For this gold standard, GraphHopper has better precision and recall than NetworkBLAST. We obtain results similar to Figure 2(b) for KEGG processes (data not shown). These results are based on 766 GraphHopper CPIMs, 835 NetworkBLAST modules, and 27 Match-and-Split modules. Thus, Match-and-Split computes many fewer modules than the other two algorithms. On average, Match-and-Split modules are much smaller than those computed by GraphHopper and NetworkBLAST. Thus, GraphHopper is competitive with and sometimes outperforms Match-and-Split in spite of computing a much larger number of CPIMs.

Comparison of topological diversity. To underscore the diversity of the topologies of the CPIMs computed by GraphHopper, we performed another comparison

[1] Narayanan and Karp only considered $t = 0.5$.

of the three algorithms. We partitioned each computed CPIM into its two species-specific components and computed the ratio of the number of proteins in the larger component and the numbers of proteins in the smaller component. We observed that both Match-and-Split and NetworkBLAST computed CPIMs for which these ratios were between one and two. In contrast, GraphHopper computed a number of CPIMs with ratio at least 2.5. An example is a CPIM containing 4 yeast and 11 fly proteins that is enriched in the cellular component "myosin" (7.8×10^{-7})[2]. Myosin is a protein complex that functions as a molecular motor, using the energy of ATP hydrolysis to move actin filaments or cargo on actin filaments. This CPIM may suggest how interactions between myosin proteins have evolved from single-celled to multi-cellular organisms.

3.2 Datasets

In the rest of this section, we present results obtained by GraphHopper on human, fly, and baker's yeast protein interaction networks. We obtained 31610 interactions between 7393 human proteins from the IDSERVE database [12]. We removed interactions in the IDSERVE data that were obtained by transfer from lower eukaryotes based on sequence similarity. We also included 3270 human interactions derived using large scale yeast two-hybrid experiments from Stelzl et al. [13], and 6726 human PPIs from Rual et al. [14]. Overall, this human PPI network contained 7787 proteins and 30703 interactions and represents interactions from a diverse variety of sources. From the Database of Interacting Proteins [15], we collected 22004 interactions between 7350 fly proteins and 15317 interactions between 5019 yeast proteins. To find orthologous pairs of proteins, we ran BLASTP on a database containing all human, fly, and yeast protein sequences and retained only bidirectional hits with E-values less than 10^{-7}. We gathered functional annotations from the Gene Ontology (GO).

3.3 A Global Map of Human-Fly CPIMs

We find 265 human-fly CPIMs enriched in 969 functions, 149 human-yeast CPIMs enriched in 784 functions, and 34 fly-yeast CPIMs enriched in 273 functions. 161 functions enriched in all three comparisons span a diverse range of cellular activities including biological process such as cytokinesis, protein metabolism, and reproduction; molecular functions including microfilament motor activity, GTPase activity, and cyclin-dependent protein kinase activity; and cellular components such as the microtubule and the endoplasmic reticulum.

We find 163 functions enriched exclusively in human-fly CPIMs. Many of these functions are unique to multi-cellular organisms, for example, cell-matrix adhesion (2×10^{-13}), tissue development (3×10^{-6}), cell differentiation (1.2×10^{-13}), and ectoderm development (1.6×10^{-10}). Several CPIMs are enriched in functions related to sexual reproduction, such as embryonic development (4.2×10^{-11}), germ-line stem cell division (6.7×10^{-9}) and ovarian follicle cell development

[2] Numbers in parentheses are Bonferroni-corrected p-values of functional enrichment.

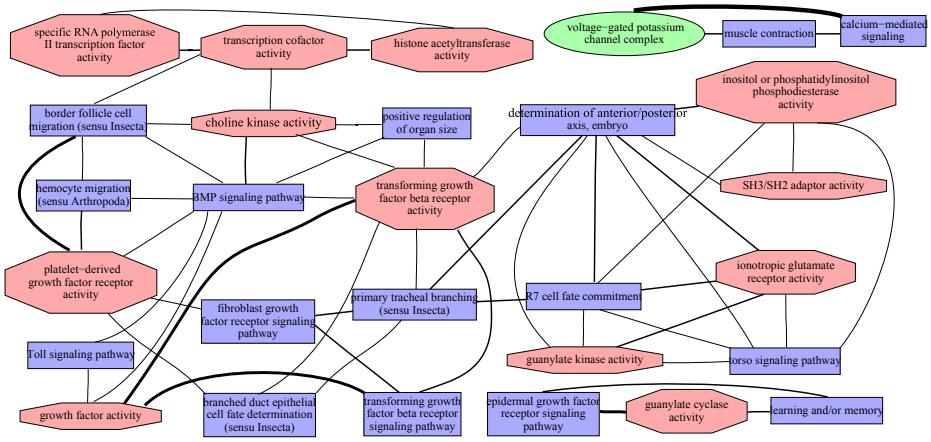


Fig. 3. A network of functions enriched in human-fly CPIMs. To construct this image, we associate each human-fly CPIM with the GO function most enriched in that CPIM, restricting our attention to functions with p-value of 10^{-4} or better. We ignore a CPIM c if there is another CPIM c' such that the GO function associated with c is an ancestor of the GO function associated with c'. We construct a network where each node is a CPIM and an edge connects two nodes if their CPIMs overlap. The thickness of an edge in Figure 3 represents the degree of overlap in terms of fraction of shared proteins (the ratio of the size of the intersection to the size of the union). We discard CPIMs that had at most 5% similarity to every other CPIM. We visualize the resulting network using the Graphviz package [16]. Blue rectangles are GO biological processes, red octagons are GO molecular functions, and green ellipses are GO cellular components.

(9.03×10^{-8}). A number of CPIMs are related to the development of the nervous system, for example, axon guidance (0.03), dopamine receptor activity (5.6×10^{-17}), and voltage-gated potassium channel complex (8.6×10^{-6}).

Figures 3 and 4(a) display connected components of a global network of functions enriched only in human-fly CPIMs and connections between these CPIMs. The largest component of the network in Figure 3 spans a diverse set of processes and functions, of which many are unique to multi-cellular organisms. The connected component of the human-fly CPIM network in Figure 4(a) connects five GO biological processes (negative regulation of fusion cell fate specification, Notch signaling pathway, ommatidial rotation, R3/R4 cell differentiation, and regulation of R8 spacing) to three GO molecular functions (calcium ion binding, extracellular matrix structural constituent, and transmembrane receptor protein phosphatase activity). Three of these CPIMs describe processes involved in eye development in fly (ommatidial rotation, R3/R4 cell differentiation, and regulation of R8 spacing). These CPIMs are connected by a module enriched in "transmembrane receptor protein phosphatase activity." Tyrosine protein phosphatases such as *Dlar* play a critical role in controlling motor neuron guidance and targeting R cells correctly to different layers of the fly compound eye [17]. Other CPIMs are enriched in the molecular functions "calcium ion binding," and "extracellular matrix structural constituent," which reflect the roles played by the human proteins in these

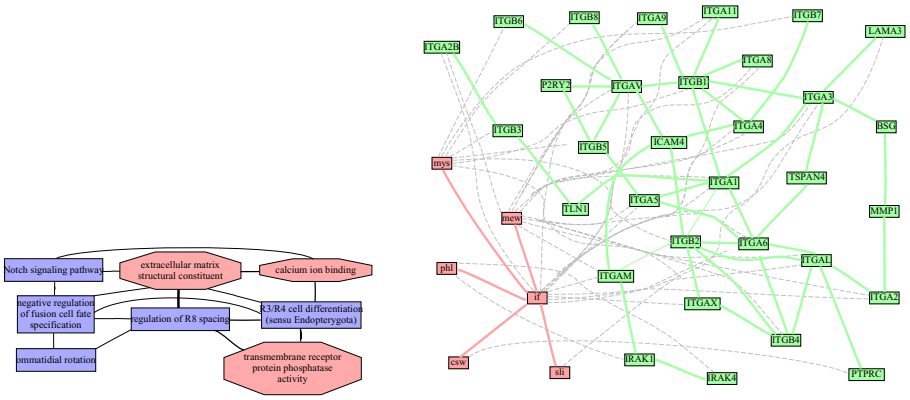


Fig. 4. (a) A connected component of the global map of functions enriched in human-fly CPIMs. (b) A human-fly CPIM conserved in the cellular component integrin complex. Fly proteins and PPIs are colored in light red, human proteins and PPIs in light green, and orthologous pairs are connected by dashed edges.

CPIMs. Many of these proteins contain calcium-binding domains and are localized to the extracellular matrix: fibrinogen beta chain (FGB) is cleaved by thrombin to form fibrin, which is an important component of blood clots; fubulin 1 (FBLN1) mediates platelet adhesion by binding fibrinogen; fibulin 5 (FBN5) is expressed in developing arteries; fibronectin 1 (FN1) is involved in cell adhesion and migration; aggrecan (ACAN) is an important part of cartilage; and fibrillin 1 (FBN) and fibrillin 2 (FBN2) are structural components of calcium-binding microfibrils, which provide structural support in elastic and nonelastic connective tissue throughout the body. These CPIMs suggest that fly proteins involved in eye development have human orthologs that have evolved functions related to blood clotting, vascular development, and structural support.

Figure 4(b) displays a CPIM enriched in the cellular component integrin complex (6.3×10^{-62}). The fly sub-network in this CPIM contains only six proteins of which three proteins (mew, if and mys) are members of the fly integrin complex, while the human sub-network contains 32 proteins of which 22 are members of the integrin complex. As the integrin complex is involved in cell-matrix adhesion, we would not expect the integrin complex to be present in yeast; indeed no yeast genes are annotated with this component. The fly PPI network contains very few interactions between integrins, which are membrane proteins. The fly PPI network was generated using a large-scale yeast two-hybrid assay [18] and it is well-known that this assay fails to detect interactions involving membrane proteins. On the other hand, the interactions between the proteins in the integrin complex in the human PPI network are manually curated from the literature and included in the HPRD database [19], which in turn is included in the IDSERVE dataset [12] used in this paper.

4 Discussion

Earlier methods [2,4,9] for computing CPIMs have succeeded in detecting complexes and pathways conserved between two or more species. For instance, these models assume a pathway-like [2] or a complex-like [4] interaction structure between all the proteins in a module. Methods that integrate multiple PPI networks into a single alignment graph [2,3,4] are likely to compute CPIMs where the constituent protein interaction modules have similar topologies. The Graemlin algorithm allows the user to specify the topology of the protein interaction modules to be aligned; however, both modules in a CPIM must have similar topologies. An advantage that some previous methods have over GraphHopper is that they can simultaneously align more than two PPI networks [20,7,9].

CPIMs found by GraphHopper have a wider range of topologies than those computed by other methods. For example, the CPIM in Figure 4(b) maps the integrin complex in fly (6 proteins, 5 interactions) to a much larger and more dense human integrin network (32 proteins, 85 interactions). Such CPIMs are useful for capturing the increased diversity and complexity of a module of proteins in a higher eukaryote. This CPIM also demonstrates GraphHopper's ability to align a clique-like module with a module like a star graph.

We conclude by noting that CPIMs have been used to transfer protein functional annotations from one organism to another [6,9]. Most predicted functions correspond to fundamental processes of life. Our results, e.g., the suggested evolution of eye development proteins in fly to human proteins involved in blood clotting and vascular development, indicate the transfer of function for processes unique to multi-cellular organisms requires new techniques.

Acknowledgments. Grants from the ASPIRES program and the Institute for Critical Technology and Applied Science at Virginia Tech supported this research. We thank Vandana Sreedharan for many useful discussions.

References

1. Sharan, R., Ideker, T.: Modeling cellular machinery through biological network comparison. Nat. Biotechnol. 24(4), 427–433 (2006)
2. Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proc. Natl. Acad. Sci. U S A 100(20), 11394–11399 (2003)
3. Koyuturk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., Grama, A.: Pairwise alignment of protein interaction networks. J. Comput. Biol. 13(2), 182–199 (2006)
4. Sharan, R., Ideker, T., Kelley, B.P., Shamir, R., Karp, R.M.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. In: RECOMB 2004: Proceedings of the eighth annual international conference on Computational molecular biology, pp. 282–289. ACM Press, New York (2004)

5. Suthram, S., Shlomi, T., Ruppin, E., Sharan, R., Ideker, T.: A direct comparison of protein interaction confidence assignment schemes. BMC Bioinformatics 7, 360 (2006)
6. Narayanan, M., Karp, R.M.: Comparing Protein Interaction Networks via a Graph Match-and-Split Algorithm. J. Comput. Biol. 14(7), 892–907 (2007)
7. Koyuturk, M., Kim, Y., Subramaniam, S., Szpankowski, W., Grama, A.: Detecting conserved interaction patterns in biological networks. J. Comput. Biol. 13(7), 1299–1322 (2006)
8. Gandhi, T.K., et al.: Analysis of the human protein interactome and comparison with yeast, worm and fly interaction datasets. Nat. Genet. 38(3), 285–293 (2006)
9. Sharan, R., Suthram, S., Kelley, R.M., Kuhn, T., Mccuine, S., Uetz, P., Sittler, T., Karp, R.M., Ideker, T.: From the cover: Conserved patterns of protein interaction in multiple species. Proc. Natl. Acad. Sci. U S A 102(6), 1974–1979 (2005)
10. Guldener, U., Munsterkotter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H.W., Stumpflen, V.: MPact: the MIPS protein interaction resource on yeast. Nucleic Acids Res. 34, D436–D441 (2006)
11. Goldberg, D.S., Roth, F.P.: Assessing experimentally derived interactions in a small world. Proc. Natl. Acad Sci. U S A 100(8), 4372–4376 (2003)
12. Ramani, A.K., Bunescu, R.C., Mooney, R.J., Marcotte, E.M.: Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. Genome Biol. 6(5), R40 (2005)
13. Stelzl, U., et al.: A human protein-protein interaction network: a resource for annotating the proteome. Cell 122(6), 957–968 (2005)
14. Rual, J., et al.: Towards a proteome-scale map of the human protein-protein interaction network. Nature 437(7062), 1173–1178 (2005)
15. Xenarios, I., Rice, D.W., Salwinski, L., Baron, M.K., Marcotte, E.M., Eisenberg, D.: DIP: the database of interacting proteins. Nucleic Acids Res. 28(1), 289–291 (2000)
16. Gansner, E.R., North, S.C.: An open graph visualization system and its applications to software engineering. Software – Practice and Experience 30(11), 1203–1233 (2000)
17. Araujo, S.J., Tear, G.: Axon guidance mechanisms and molecules: lessons from invertebrates. Nat. Rev. Neurosci. 4(11), 910–922 (2003)
18. Giot, L., et al.: A protein interaction map of drosophila melanogaster. Science 302(5651), 1727–1736 (2003)
19. Peri, S., et al.: Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res. 13(10), 2363–2371 (2003)
20. Flannick, J., Novak, A., Srinivasan, B.S., McAdams, H.H., Batzoglou, S.: Graemlin: general and robust alignment of multiple large interaction networks. Genome Res. 16(9), 1169–1181 (2006)