

Algorithms to Detect Multiprotein Modularity Conserved during Evolution

Luqman Hodgkinson and Richard M. Karp

Abstract—Detecting essential multiprotein modules that change infrequently during evolution is a challenging algorithmic task that is important for understanding the structure, function, and evolution of the biological cell. In this paper, we define a measure of modularity for interactomes and present a linear-time algorithm, Produles, for detecting multiprotein modularity conserved during evolution that improves on the running time of previous algorithms for related problems and offers desirable theoretical guarantees. We present a biologically motivated graph theoretic set of evaluation measures complementary to previous evaluation measures, demonstrate that Produles exhibits good performance by all measures, and describe certain recurrent anomalies in the performance of previous algorithms that are not detected by previous measures. Consideration of the newly defined measures and algorithm performance on these measures leads to useful insights on the nature of interactomics data and the goals of previous and current algorithms. Through randomization experiments, we demonstrate that conserved modularity is a defining characteristic of interactomes. Computational experiments on current experimentally derived interactomes for *Homo sapiens* and *Drosophila melanogaster*, combining results across algorithms, show that nearly 10 percent of current interactome proteins participate in multiprotein modules with good evidence in the protein interaction data of being conserved between human and *Drosophila*.

Index Terms—Modularity, interactomes, evolution, algorithms.

1 INTRODUCTION

INTERACTIONS between proteins in many organisms have been mapped, yielding large protein interaction networks, or interactomes [1]. The present paper continues a stream of scientific investigation focusing on conservation of modular structure of the cell, such as protein signaling pathways and multiprotein complexes, across organisms during evolution, with the premise that such structure can be described in terms of graph theoretic properties in the interactomes [2], [3], [4], [5], [6]. This stream of investigation has led to many successes, discovering conserved modularity across a wide range of evolutionary distances. However, there remain many challenges, such as running time, potential false positive predictions of conservation, coherence of predicted conserved modules, and absence of a comprehensive collection of evaluation measures.

Biological systems tend to be modular, and, thus, evolvable [7]. In an interactome, each meaningful protein interaction places an evolutionary constraint on the protein interactants. Evolution of cells would be severely restricted if interactomes were complete graphs with each interaction being essential for biological viability. Modularity allows each module to evolve with limited dependence on the evolution of other modules.

In this paper, we propose a definition of modularity for interactomes. We present an algorithm, Produles, designed to detect modular regions conserved during evolution.

Produles runs in linear time in the size of the input and is efficient in practice while yielding exceptionally good results. Through computational experiments on current experimentally derived interactomes for *Homo sapiens* and *Drosophila melanogaster*, we find good evidence that nearly 10 percent of the interactome proteins participate in multiprotein modules that have been conserved across this evolutionary distance, and we demonstrate significance of these results.

Some previous algorithms for related problems, including NetworkBlast [3] and Graemlin [5], use a scoring function that is a sum of multiple scores: one score based on protein sequence similarity, and one score from each organism based on the density of interactions among the module proteins for that organism. These algorithms use a greedy search on this scoring function to find conserved modules. Due to the additivity, module pairs similar to the diagrams in Fig. 1 may receive high scores and be reported as conserved.

Good module boundaries are important for the modules that are returned by an algorithm. Fig. 1(a) illustrates the situation in which module boundaries may not be well defined as there is no evidence in the protein interaction data that the various components belong in the same module.

Evidence of conservation in the interaction data across organisms is essential for modules claimed by an algorithm to be conserved during evolution. Homologous proteins may be reorganized during evolution into multiprotein modules that differ both in composition and in function across organisms [8]. Due to the additivity of the scoring function for some previous algorithms, including NetworkBlast and Graemlin, in the interaction densities across organisms, a very dense network in one organism can be reported as conserved with homologous proteins in another organism that have zero or few interactions among them. In this case, as illustrated in Fig. 1(b), the interaction data does not

• The authors are with the Division of Computer Science and the Center for Computational Biology, University of California, Berkeley, and the International Computer Science Institute, Berkeley, CA 94720.
E-mail: {luqman, karp}@icsi.berkeley.edu.

Manuscript received 2 July 2011; accepted 22 Aug. 2011; published online 27 Sept. 2011.

For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBSI-2011-07-0170.

Digital Object Identifier no. 10.1109/TCBB.2011.125.

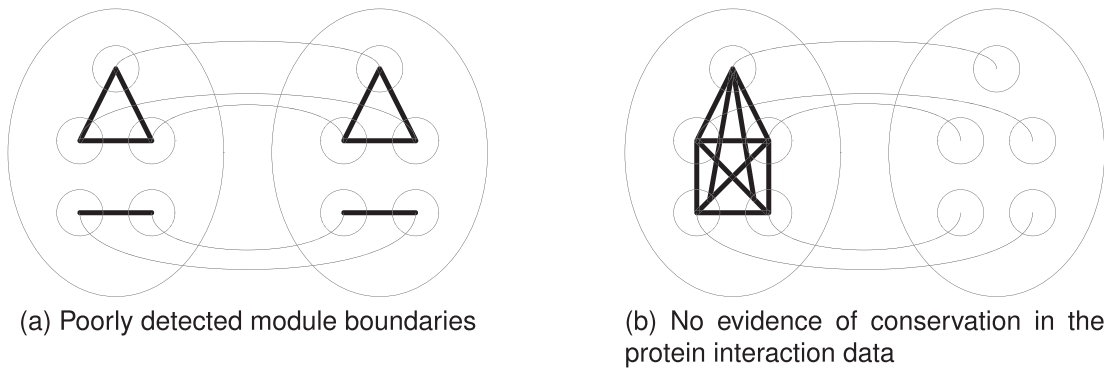


Fig. 1. Diagrams illustrating difficulties with additivity across data types and organisms. Organisms are represented by ovals. Proteins are represented by circles. Protein interactions are represented by thick lines. Proteins with high sequence similarity are connected with thin lines. Algorithms that are additive across the interaction and sequence data may predict module (a) to be conserved due to high sequence similarity. In this case, the module boundaries are not well defined, most likely containing portions of multiple modules that have no relation with each other. Algorithms that are additive in the interactions across organisms may predict module (b) to be conserved though there is no evidence for module conservation across the organisms in the protein interaction data.

support a claim of module conservation across the given organisms.

Produdes is an important step to address these issues. Produdes runs in linear time, scaling better than Match-and-Split [6] and MaWISH [4], and does not exhibit the recurrent anomalies that result from the additivity of the scoring function across organisms and data sources that forms the basis for NetworkBlast and Graemlin.

Demonstrating in Section 6 that existing evaluation measures are insufficient for a thorough comparison of algorithms and their goals, in Section 4 we present a collection of biologically motivated graph theoretic evaluation measures complementary to previous evaluation measures. These measures clearly elucidate the strengths and weaknesses of each approach. Results from computational experiments show that the approach we propose based on modularity detects multiprotein modules with good evidence of conservation. These measures lead to useful insights on the goals of various algorithms and illuminate characteristics of current interactomics data sets presented and discussed in Sections 6 and 7.

1.1 Form of Study Data

An interactome is an undirected graph $G = (V, E)$, where V is a set of proteins and $(v_1, v_2) \in E$ if and only if protein v_1 is found to interact with protein v_2 . In this study, the input is restricted to a pair of interactomes, $G_i = (V_i, E_i)$, for $i \in \{1, 2\}$, and protein sequence similarity values, $h: V_1 \times V_2 \rightarrow \mathbb{R}^+$, defined only for the most sequence similar pairs of proteins appearing in the interactomes. In this study, h is derived from BLAST [9] E-values. As BLAST E-values change when the order of the interactomes is reversed, h is defined with the rule

$$h(v_1, v_2) = h(v_2, v_1) = \frac{E(v_1, v_2) + E(v_2, v_1)}{2},$$

where $E(v_1, v_2)$ is the minimum BLAST E-value for $v_1 \in V_1$, $v_2 \in V_2$ when v_1 is tested for homology against the database formed by V_2 . An algorithm using this data as input is general to any pair of interactomes, including those for newly studied organisms.

2 MODULARITY

A modular system consists of parts organized in such a way that strong interactions occur within each group or module, but parts belonging to different modules interact only weakly [10]. Following this, a natural definition of multiprotein modularity recognizes that proteins within a module are more likely to interact with each other than to interact with proteins outside of the module. Let $G = (V, E)$ be an interactome. A multiprotein module is a set of proteins $M \subset V$ such that $|M| \ll |V|$ and M has a large value of

$$\mu(M) = \frac{|E(M)|}{|\text{cut}(M, V \setminus M)| + |E(M)|},$$

where $E(M)$ is the set of interactions with both interactants in M , and $\text{cut}(M, V \setminus M)$ is the set of interactions spanning M and $V \setminus M$. Of the interactions involving proteins in M , the fraction contained entirely within M is given by $\mu(M)$. This definition of modularity is similar, but not identical, to the recent definition of λ -module [11].

The conductance of a set of vertices in a graph is defined as

$$\Phi(M) = \frac{|\text{cut}(M, V \setminus M)|}{|\text{cut}(M, V \setminus M)| + 2 \min(|E(M)|, |E(V \setminus M)|)}.$$

When $|E(M)| \leq |E(V \setminus M)|$, as for all applications in this study,

$$\Phi(M) = \frac{|\text{cut}(M, V \setminus M)|}{|\text{cut}(M, V \setminus M)| + 2|E(M)|} = \frac{1 - \mu(M)}{1 + \mu(M)}.$$

Thus, when searching for relatively small modules in a large interactome, minimizing conductance is equivalent to maximizing modularity. This relationship allows us to modify powerful algorithms from theoretical computer science designed for minimizing conductance [12], [13]. It has previously been shown that conductance in protein interaction networks is negatively correlated with functional coherence [14], in agreement with our findings in Section 6.

2.1 Modularity and Degree Bounds

Assuming we are searching for modules of size at most b with modularity at least d , the vertices in any such module have bounded degree. Let $\delta(u)$ be the degree of u in G .

Theorem 1 (Modularity-Maximizing Degree Bound). *If $d > 0$, the objective function in the optimization problem*

$$\begin{aligned} \max_{G, M, u} \quad & \delta(u) \\ \text{s.t.} \quad & u \in M, \\ & |M| = b, \\ & \mu(M) \geq d, \\ & \mu(M) > \mu(M \setminus \{u\}), \end{aligned}$$

satisfies the bound $\delta(u) < (b-1)(1+d)/d$.

Proof. Let $M' \triangleq M \setminus \{u\}$. Let $y \triangleq |E(M')|$. Let $x \triangleq |\text{cut}(M', \{u\})|$.

$$\mu(M') = \frac{y}{|\text{cut}(M', V \setminus M')| + y} < \mu(M),$$

so

$$|\text{cut}(M', V \setminus M')| > \frac{y(1 - \mu(M))}{\mu(M)}.$$

Thus,

$$\begin{aligned} \mu(M) &= \frac{x + y}{[\delta(u) - x] + [|\text{cut}(M', V \setminus M')| - x] + [x + y]} \\ &< \frac{x + y}{\delta(u) - x + y + \frac{y(1 - \mu(M))}{\mu(M)}}, \end{aligned}$$

which implies

$$\mu(M) < \frac{x}{\delta(u) - x}.$$

As $\mu(M) \geq d$,

$$\delta(u) < \frac{x(1+d)}{d} \leq \frac{(b-1)(1+d)}{d}. \quad \square$$

The motivation for the restriction $\mu(M) > \mu(M \setminus \{u\})$ is that when searching for modules with high modularity, there can be proteins with such high degrees that it always improves the modularity to remove them from the module.

Theorem 2 (Tightness of Degree Bound). *If $d \leq \frac{b-2}{b}$, the bound in Theorem 1 is tight and neither requiring connectivity of M in the underlying graph nor requiring connectivity of $M \setminus \{u\}$ in the underlying graph can allow the bound to be further tightened.*

Proof. Consider a module M of size b that induces a clique in an underlying graph G . Of these vertices, only two, u and v , are incident on edges that extend outside of the clique. Let

$$\begin{aligned} \delta(u) &= \frac{(b-1)(1+d)}{d} - \epsilon, \\ \delta(v) &= \binom{b-1}{2} \left(\frac{1-d}{d} \right) + \epsilon', \end{aligned}$$

for $\epsilon \geq \epsilon' > 0$ chosen so that $\delta(u)$ and $\delta(v)$ are integers. To see that $\delta(v) > b-1$, implying that v can indeed be in the clique with at least one edge extending outside of the clique, examine the equivalent claim:

$$\binom{b-1}{2} \left(\frac{1-d}{d} \right) \geq b-1,$$

which is equivalent to $d \leq \frac{b-2}{b}$, as seen by expanding and solving for d . Then,

$$\begin{aligned} \mu(M) &= \frac{\binom{b}{2}}{\delta(u) + \delta(v) - 2(b-1) + \binom{b}{2}} \\ &= \frac{db}{b + \frac{2d}{b-1}(\epsilon' - \epsilon)} \\ &\geq d \end{aligned}$$

and

$$\begin{aligned} \mu(M \setminus \{u\}) &= \frac{\binom{b-1}{2}}{\delta(v) - (b-2) + (b-2) + \binom{b-1}{2}} \\ &= \frac{d(b-1)(b-2)}{(b-1)(b-2) + 2d\epsilon'} \\ &< d, \end{aligned}$$

where in both cases the second equation follows from the first by multiplying numerator and denominator by $2d$ and simplifying.

It remains to show that ϵ can be taken arbitrarily small while maintaining integrality of $\delta(u)$ and $\delta(v)$. Let

$$\epsilon = \epsilon' = \frac{1}{b}.$$

By rearrangement, we have

$$\begin{aligned} \delta(u) &= \frac{b-1}{d} + (b-1) - \frac{1}{b}, \\ \delta(v) &= \frac{\binom{b-1}{2}}{d} - \binom{b-1}{2} + \frac{1}{b}, \end{aligned}$$

so it suffices to show that

$$\begin{aligned} \delta'(u) &\triangleq \frac{b-1}{d} - \frac{1}{b}, \\ \delta'(v) &\triangleq \frac{\binom{b-1}{2}}{d} + \frac{1}{b}, \end{aligned}$$

are integers by choosing d appropriately. Let

$$d = \frac{\binom{b-1}{2}}{k - \frac{1}{b}},$$

for k integer and sufficiently large. Then,

$$\begin{aligned} \delta'(u) &= \frac{2k-1}{b-2}, \\ \delta'(v) &= k, \end{aligned}$$

are both integers for b odd if k is chosen so that $2k-1$ is a multiple of $b-2$. That is,

$$k = \frac{k'(b-2) + 1}{2},$$

for b and k' sufficiently large positive integers with b odd and k' even. As $b \rightarrow \infty$, $\epsilon = \frac{1}{b} \rightarrow 0$. Finally, note that both M and $M \setminus \{u\}$ induce connected subgraphs in the underlying graph G , completing the proof. \square

If $d > \frac{b-2}{b}$, which is unlikely to be the case for applications in this paper, a second bound

$$\delta(u) \leq \frac{\binom{b}{2}}{d} - \binom{b-1}{2},$$

that holds for all $d > 0$, is tight, and, similarly, cannot be further tightened by requiring connectivity of M or $M \setminus \{u\}$ in the underlying graph.

2.2 Hierarchy of Modularity

When an algorithm detects conserved multiprotein modules that share proteins, these modules can be combined into a larger composite module. Whenever this union takes place, the following theorem shows that the modularity of the composite module is at least as large as the minimum modularity of the modules being combined.

Theorem 3 (Modularity Minimum Monotonicity). *For modules M_1, M_2 , it is true that*

$$\mu(M_1 \cup M_2) \geq \min\{\mu(M_1), \mu(M_2)\}.$$

Proof. Let $e(M) \triangleq |E(M)|$. Let $f(M, M') \triangleq |\text{cut}(M, M')|$. Let $g(M) \triangleq f(M, V \setminus M)$. Then,

$$\begin{aligned} \mu(M_1 \cup M_2) &= \frac{e(M_1 \cup M_2)}{g(M_1 \cup M_2) + e(M_1 \cup M_2)} \\ &= \frac{e(M_1) + e(M_2) + f(M_1, M_2)}{g(M_1) + g(M_2) - 2f(M_1, M_2) + e(M_1) + e(M_2) + f(M_1, M_2)} \\ &= \frac{e(M_1) + e(M_2) + f(M_1, M_2)}{g(M_1) + g(M_2) + e(M_1) + e(M_2) - f(M_1, M_2)} \\ &\geq \frac{e(M_1) + e(M_2)}{g(M_1) + e(M_1) + g(M_2) + e(M_2)} \\ &\geq \min\left\{\frac{e(M_1)}{g(M_1) + e(M_1)}, \frac{e(M_2)}{g(M_2) + e(M_2)}\right\} \\ &= \min\{\mu(M_1), \mu(M_2)\}. \end{aligned}$$

The final inequality follows from the lemma

$$\frac{a+b}{c+d} \geq \min\left\{\frac{a}{c}, \frac{b}{d}\right\} \quad \text{for } a, b \geq 0 \text{ and } c, d > 0,$$

which can be proved by observing that

$$\frac{a+b}{c+d} < \frac{a}{c} \Rightarrow c(a+b) < a(c+d) \Rightarrow bc < ad,$$

whereas

$$\frac{a+b}{c+d} < \frac{b}{d} \Rightarrow d(a+b) < b(c+d) \Rightarrow ad < bc,$$

which cannot both be true. \square

A hierarchy of modularity consists of larger conserved modules composed of smaller conserved modules while maintaining a desired minimum modularity for all modules at all levels of the hierarchy.

3 ALGORITHMS

3.1 Modularity Maximization Algorithms

3.1.1 PageRank-Nibble

PageRank-Nibble [12] is an algorithm for finding a module with low conductance in a graph $G = (V, E)$. Let A be the

adjacency matrix for G . Let D be a diagonal matrix with diagonal entries $D_{ii} = \delta(i)$ where $\delta(i)$ is the degree of vertex i in G . Let $W = (AD^{-1} + I)/2$ where I is the identity matrix. W is a lazy random walk transition matrix that with probability $1/2$ remains at the current vertex and with probability $1/2$ randomly walks to an adjacent vertex. A PageRank vector is a row vector solution $\text{pr}(\alpha, s)$ to the equation

$$\text{pr}(\alpha, s) = \alpha s + (1 - \alpha)\text{pr}(\alpha, s)W^T,$$

where $\alpha \in (0, 1]$, and s is a row vector distribution on the vertices of the graph. Define the distribution that places all mass at vertex v ,

$$\chi_v(u) = \begin{cases} 1, & \text{if } u = v, \\ 0, & \text{otherwise.} \end{cases}$$

When $s = \chi_v$, a PageRank vector is a weighted sum of the probability distributions obtained by taking a sequence of lazy random walk steps starting from v , where the weight placed on the distribution obtained after t walk steps decreases exponentially in t [12].

Let p be a distribution on the vertices of G , and let the vertices be sorted in descending order by $p(v)/\delta(v)$, the frequency of v in distribution p normalized by the stationary distribution of an unrestricted random walk. Let $S_j(p)$ be the set of the first j vertices after sorting. For $j \in \{1, \dots, |V|\}$, the set $S_j(p)$ is called a sweep set [12].

PageRank-Nibble consists of computing an approximate Page-Rank vector with $s = \chi_v$, defined as

$$\text{apr}(\alpha, s, r) \triangleq \text{pr}(\alpha, s) - \text{pr}(\alpha, r),$$

where r is a residual vector defined below, and then returning the sweep set $S_j(\text{apr}(\alpha, \chi_v, r))$ with minimum conductance [12].

From the definition, if p is a vector that satisfies $p + \text{pr}(\alpha, r) = \text{pr}(\alpha, \chi_v)$, then $p = \text{apr}(\alpha, \chi_v, r)$. Thus, $0 = \text{apr}(\alpha, \chi_v, \chi_v)$. After initializing $p_1 = 0$, $r_1 = \chi_v$, the approximation $\text{apr}(\alpha, \chi_v, r)$ to $\text{pr}(\alpha, \chi_v)$ is improved iteratively. Each iteration, called a push operation, chooses an arbitrary vertex u such that $r_i(u)/\delta(u) \geq \epsilon$. Then $p_{i+1} = p_i$ and $r_{i+1} = r_i$, except for the following changes:

1. $p_{i+1}(u) = p_i(u) + \alpha r_i(u)$,
2. $r_{i+1}(u) = (1 - \alpha)r_i(u)/2$,
3. $r_{i+1}(v) = r_i(v) + (1 - \alpha)r_i(u)/(2\delta(u))$ for each v such that $(u, v) \in E$,

in which $\alpha r_i(u)$ probability is sent to $p_{i+1}(u)$, and the remaining $(1 - \alpha)r_i(u)$ probability is redistributed in r_{i+1} using a single lazy random walk step [12].

Each push operation maintains the invariant [12]

$$p_i + \text{pr}(\alpha, r_i) = \text{pr}(\alpha, \chi_v).$$

When no additional pushes can be performed, the final residual vector r satisfies

$$\max_{u \in V} \frac{r(u)}{\delta(u)} < \epsilon.$$

The running time for computing $\text{apr}(\alpha, \chi_v, r)$ is $O(1/(\epsilon\alpha))$ [12]. If ϵ and α are set to constants, reasonable given their meanings, and if only the first b sweep sets are considered, the algorithm runs in constant time. Ensuring the degrees of the vertices satisfy the bound in Theorem 1,

we do not consider sweep sets that contain vertices with degree $(b-1)(1+d)/d$ or greater, and we also require connectivity in the underlying graph.

3.1.2 Greedy Algorithm

To verify that PageRank-Nibble returns modules with near-optimal modularity, we use a greedy algorithm that grows a module by adding the neighboring protein that confers greatest improvement to the modularity. By considering only proteins that satisfy the degree bound from Theorem 1, the algorithm runs in time $O(b^3/d)$. Though this algorithm is slow, comparing its results with the faster PageRank-Nibble increases our confidence in PageRank-Nibble. Our studies show that PageRank-Nibble performs best in the minimum amount of time relative to this greedy algorithm and to Nibble [13], so Produles uses PageRank-Nibble by default and in the remainder of this study.

3.2 Algorithm to Detect Conservation

The algorithm begins by finding a multiprotein module,

$$M \subset V_1,$$

with high modularity in G_1 using a modularity maximization algorithm such as those described in Section 3.1. Let

$$\mathcal{H}_T(M) = \{v \mid \exists u \in M \text{ such that } h(u, v) \leq T\}.$$

Modules corresponding to the connected components of the subgraph of G_2 induced by $\mathcal{H}_T(M)$ are candidates for conservation with M . Let these modules be N_1, N_2, \dots, N_k . For $i = 1, \dots, k$, let

$$\mathcal{R}_T(M, N_i) = \{u \in M \mid \exists v \in N_i \text{ such that } h(u, v) \leq T\}.$$

If the following are true:

$$a \leq |\mathcal{R}_T(M, N_i)| \leq b,$$

$$a \leq |N_i| \leq b,$$

$$\frac{1}{c}|N_i| \leq |\mathcal{R}_T(M, N_i)| \leq c|N_i|,$$

$$\mu(\mathcal{R}_T(M, N_i)) \geq d,$$

$$\mu(N_i) \geq d,$$

where a is a lower bound on size, b is an upper bound on size, c is a size balance parameter, and d is a lower bound on desired modularity, and if $\mathcal{R}_T(M, N_i)$ yields a connected induced subgraph of G_1 , then we report the pair $(\mathcal{R}_T(M, N_i), N_i)$ as a conserved multiprotein module.

Each protein is used exactly once as a starting vertex for the modularity maximization algorithm. A counter is maintained for each protein in G_1 . When a protein is placed in a module by the modularity maximization algorithm, the counter for the protein is incremented. Each counter has maximum value e for some constant e . The modularity maximization algorithm is restricted to search over proteins with counter value less than e . If a protein in G_1 is reported to be in a conserved module, the counter for the protein is set to $e/2$ in order to reduce module overlap. Furthermore, interactions in the subgraphs induced by the module are marked, preventing these interactions from being used in future searches by the modularity maximization algorithm. When all proteins

in G_1 have been used as starting vertices, the roles of G_1 and G_2 are reversed, and the entire process is repeated.

3.3 Refinement of Large Connected Components

A module, $M \subset V_1$, may contain proteins that are homologous to a large number of proteins, $S \subseteq V_2$, and S may form a large connected induced subgraph in G_2 . In many cases, the size of S cannot reasonably be explained by duplication of module proteins after divergence from the most recent common ancestor. Two reasons explain the majority of this phenomenon. First, proteins may share peripheral domains that cause protein homology detection algorithms to detect proteins only partially homologous, which may interact with module proteins, either genuinely or due to artifacts from experimental assays such as yeast two-hybrid. Second, paralogous modules that may be kept separate by the cell, performing different functions, contain homologous proteins, leading proteins in paralogous modules to be incorrectly detected as interacting. Refinement of large connected components aims to remove partially homologous proteins and to separate paralogous modules.

A first approach is to consider biconnected components rather than connected components. If paralogous modules are connected only loosely by bridges, they can be separated using biconnected components.

As a second approach, we design a heuristic algorithm that requires time linear in the size of the subgraph induced by S . The algorithm proceeds by iterations. At each iteration, each subgraph protein, $u \in S \subseteq V_2$, records the difference between the modularity of the subgraphs induced by S and $S \setminus \{u\}$. Each module protein, $v \in M \subset V_1$, is assigned a distinct color and transfers its color to all homologous proteins in S . Of the proteins in S with the most frequent color, half are removed, precisely those that individually benefit the modularity least. The algorithm iterates for a maximum of b^2 iterations, ensuring that a subgraph of size 2^b can be separated into modules of size at most b . After each iteration, tests are performed for connected components and biconnected components that can reasonably be reported as conserved according to the tests in Section 3.2.

3.4 Proof of Linear Running Time

Each value of $h(v, \cdot)$ for $v \in V$ is considered only when constructing $\mathcal{H}_T(M)$ for $\{M : v \in M\}$, so each value of $h(v, \cdot)$ is considered at most e times. If v is stored at each vertex in $\mathcal{H}_T(M)$ when constructing $\mathcal{H}_T(M)$, then constructing $\mathcal{R}_T(M, N_i)$ is a union of vertex lists and does not require additional considerations of $h(v, \cdot)$ values. As for all $v \in V_1$,

$$|\{M : v \in M\}| \leq e,$$

the number of considerations of h values is

$$\begin{aligned} \sum_M \sum_{v \in M} |h(v, \cdot)| &= \sum_v \sum_{M: v \in M} |h(v, \cdot)| \\ &\leq e \sum_v |h(v, \cdot)| \\ &= e|h(\cdot, \cdot)|. \end{aligned}$$

After finding $\mathcal{H}_T(M)$, it is necessary to compute N_1, N_2, \dots, N_k . This can be problematic if any of the vertices in $\mathcal{H}_T(M)$ have large degree, which could

conceivably be as large as $|V_2| - 1$. However, as we desire N_i such that $\mu(N_i) \geq d$ and $|N_i| \leq b$, which ideally do not contain any vertex u such that $\mu(N_i \setminus \{u\}) > \mu(N_i)$, we can discard, by Theorem 1, any vertex $v \in \mathcal{H}_T(M)$ with degree in G_2 of $(b - 1)(1 + d)/d$ or greater. A modified depth-first search that transitions only among vertices in $\mathcal{H}_T(M)$ is then used to compute N_1, N_2, \dots, N_k . This requires time

$$O\left(\left(\frac{(b - 1)(1 + d)}{d}\right)|\mathcal{H}_T(M)|\right) = O(|\mathcal{H}_T(M)|).$$

As

$$|\mathcal{H}_T(M)| \leq \sum_{v \in M} |h(v, \cdot)|,$$

all of these depth-first searches over the full run of the algorithm require time

$$O\left(\sum_M |\mathcal{H}_T(M)|\right) = O\left(\sum_M \sum_{v \in M} |h(v, \cdot)|\right) = O(|h(\cdot, \cdot)|).$$

For a given M , constructing all $\mathcal{R}_T(M, N_i)$ by a union of lists stored at the vertices in the N_i requires time $O(\sum_i |N_i| b \log b) = O(|\mathcal{H}_T(M)|)$. Testing for connectivity of a single $\mathcal{R}_T(M, N_i)$ with a modified depth-first search that transitions only among vertices in $\mathcal{R}_T(M, N_i)$ requires constant time as $|\mathcal{R}_T(M, N_i)| \leq b$ and as each vertex in M has degree bounded by $(b - 1)(1 + d)/d$. All of these constructions and depth-first searches over the full run of the algorithm can be completed in time $O(\sum_M |\mathcal{H}_T(M)|) = O(|h(\cdot, \cdot)|)$.

Computing the modularity of module $U \in \{N_i, \mathcal{R}_T(M, N_i)\}$ requires computing the sum of degrees of the vertices in U and the number of edges with both endpoints in U . These can be computed in constant time when $|U| \leq b$ as each vertex in U has degree bounded by $(b - 1)(1 + d)/d$.

The refinement heuristic requires time $O(|\mathcal{H}_T(M)|)$ per iteration maintaining overall linear time. Attaining this time complexity requires using a linear-time median selection algorithm. A fast randomized median selection algorithm yields expected linear time whereas the median-of-medians algorithm [15] ensures worst-case linear time. Computing biconnected components maintains the same time complexity as connected components by using a classic algorithm [16].

4 EVALUATION MEASURES

In this section, we present a set of biologically motivated graph theoretic measures that illuminate the characteristics and goals of various algorithms and qualities of the interactomics data. Five goals arising from these measures are presented with comments on when they may not be attained.

4.1 Output and Coverage

The algorithms return output that can be expressed as follows:

Definition 1 (Algorithm Output). Let k pairs of conserved modules returned by an algorithm be

$$\mathcal{M} = \{(M_1^i, M_2^i) \mid i \in \{1, \dots, k\}\}.$$

Let $(M_1, M_2) \in \mathcal{M}$. Let $M \in \{M_1, M_2\}$.

Definition 2 (Proteome Coverage). Let

$$C_i = |\mathcal{U}_i|/|V_i|,$$

where \mathcal{U}_i is the set of proteins from V_i that are part of conserved modules. Let

$$C = (C_1 + C_2)/2.$$

Definition 3 (Module Size). Let $S(M) = |M|$.

4.2 Overlap

Algorithms differ in the amount and nature of module pair overlap allowed in the algorithm output. Three measures of overlap illuminate these characteristics.

Definition 4 (Maximum Overlap). Let

$$\mathcal{O}_k^{ji} = |M_k^j \cap M_k^i|/|M_k^i|.$$

Let

$$\mathcal{O}_{max}^i(M_1^i, M_2^i) = \max_{j \neq i} \min\{\mathcal{O}_1^{ji}, \mathcal{O}_2^{ji}\}.$$

A value of $\mathcal{O}_{max}^i = x$ implies that no module pair $j \neq i$ exists that covers more than fraction x of each module in module pair i .

Definition 5 (Sum Overlap). Let

$$\mathcal{O}_{sum}^i(M_1^i, M_2^i) = \sum_{j \neq i} \min\{\mathcal{O}_1^{ji}, \mathcal{O}_2^{ji}\}.$$

Definition 6 (Cardinality Overlap). Let

$$\mathcal{O}_{card}^i(M_1^i, M_2^i) = |\{j : j \neq i \wedge \min\{\mathcal{O}_1^{ji}, \mathcal{O}_2^{ji}\} > 0\}|.$$

Together, \mathcal{O}_{sum}^i and \mathcal{O}_{card}^i measure the extent of overlap in the algorithm output, and \mathcal{O}_{max}^i measures a limiting case. All three measures allow for module duplication during evolution.

Goal 1 (Reasonable Coverage and Overlap). k and \mathcal{C} should be in reasonable ranges with low average values of \mathcal{O}_{max}^i , \mathcal{O}_{sum}^i , and \mathcal{O}_{card}^i in a set of conserved modules with reasonable coverage and overlap.

4.3 Evidence for Claim of Conservation

These measures address the situation diagrammed in Fig. 1(b).

Definition 7 (Filled Module). Let

$$G_{int}(M) = (M, E(M)).$$

Definition 8 (Interaction Components). Let $C(M)$ be the number of connected components in $G_{int}(M)$.

Definition 9 (Module Density). Let

$$\Delta(M) = |E(M)| / \binom{|M|}{2}.$$

Definition 10 (Module Average). Let

$$f_a(M_1, M_2) = (f(M_1) + f(M_2))/2,$$

where $f \in \{\mu, S, \Delta, C\}$.

Definition 11 (Module Difference). Let

$$f_d(M_1, M_2) = |f(M_1) - f(M_2)|,$$

where $f \in \{\mu, S, \Delta, C\}$.

Goal 2 (Evidence of Module Conservation). C_d , C_a , and Δ_d should be reasonably low to provide evidence for the claim of module conservation across organisms. This may be problematic for algorithms based on models that are additive in the interaction densities across organisms.

4.4 Ancestral Multiprotein Modules

By grouping homologous proteins, this measure focuses on the number of sequence dissimilar proteins that participate in the module, presumably proteins with diverse origins and functions.

Definition 12 (Module Homology Graph). Let

$$G_{hom}(M_1, M_2) = (M_1 \cup M_2, H(M)),$$

where, for $p_1 \in M_1$, $p_2 \in M_2$, $(p_1, p_2) \in H(M)$ iff $h(p_1, p_2)$ is defined.

Definition 13 (Ancestral Protein). Let

$$p = (P_1, P_2),$$

where $P_1 \subseteq M_1$, $P_2 \subseteq M_2$, and $G_{hom}(P_1, P_2)$ is a connected component of $G_{hom}(M_1, M_2)$.

Definition 14 (Ancestral Module). Let $M_a(M_1, M_2)$ be the set of ancestral proteins for (M_1, M_2) . The arguments, M_1, M_2 , may be omitted for brevity when the context is clear.

Goal 3 (Reasonable Number of Ancestral Proteins). $|M_a|$ is a measure of the number of sequence dissimilar proteins and should be reasonably large for multiprotein modules containing proteins with diverse origins and functions.

4.5 Interaction Level Model of Evolution

This collection of measures examines agreement of conserved modules with an interaction level evolutionary model that includes interaction formation and divergence, protein duplication and divergence, and protein loss.

Definition 15 (Relationship Disagreement). Let $p, q \in M_a$, where $p = (P_1, P_2)$, $q = (Q_1, Q_2)$. For $i, j \in \{1, 2\}$, relationship disagreement means there is an interaction in G_i between some $p' \in P_i$ and some $q' \in Q_i$, but no interaction in G_j between any $p'' \in P_j$ with any $q'' \in Q_j$. Let $\mathcal{R}(M_1, M_2)$ be the number of relationship disagreements.

Definition 16 (Relationship Evolution). Let

$$E_r(M_1, M_2) = \mathcal{R}(M_1, M_2) / \binom{|M_a|}{2},$$

the fraction of possible relationship disagreements.

Definition 17 (Ancestral Module Projection). For $i \in \{1, 2\}$, let

$$\pi_i(M_a) = \{P_i \mid (P_1, P_2) \in M_a \wedge P_i \neq \emptyset\}.$$

Definition 18 (Number of Protein Duplications). Let

$$\mathcal{D}(M_1, M_2) = |M_1| - |\pi_1(M_a)| + |M_2| - |\pi_2(M_a)|.$$

Definition 19 (Protein Duplication Evolution). Let

$$E_d(M_1, M_2) = \mathcal{D}(M_1, M_2) / (|M_1| + |M_2| - 2),$$

the fraction of possible protein duplications.

Definition 20 (Number of Protein Losses). Let

$$\mathcal{L}(M_1, M_2) = 2|M_a| - |\pi_1(M_a)| - |\pi_2(M_a)|.$$

Definition 21 (Protein Loss Evolution). Let

$$E_\ell(M_1, M_2) = \mathcal{L}(M_1, M_2) / (|M_2| + |M_1|),$$

the fraction of possible protein losses.

Goal 4 (Interaction Level Evolutionary Signal). E_r , E_d , and E_ℓ should be reasonably low to detect an interaction level signal of evolutionary conservation. Unfortunately, conserved natural modules, even those that have been highly studied, may not satisfy this goal in current interactomics data sets due to artifacts in the way the data is collected and stored, leading to large amounts of noise at the individual interaction level. Furthermore, it has been proposed that there is not much selective pressure at the individual interaction level [8], which, if true, may lead to genuinely high E_r scores in natural conserved modules. The high value of E_d in the conserved modules found by numerous algorithms, as shown in Section 6.2, may reflect characteristics of modules in interactomes, such as a tendency to contain homologous domains that facilitate protein interactions in the modules [8].

4.6 Quality of Module Boundaries

This measure addresses the situation diagrammed in Fig. 1(a).

Definition 22 (Ancestral Protein Projection). For ancestral protein $p = (P_1, P_2)$, P_i is the projection of p on M_i for $i \in \{1, 2\}$.

Definition 23 (Ancestral Components). Let $C(M_a)$ be the number of connected components in a graph with vertex set M_a , where an edge is defined between two ancestral proteins $p, q \in M_a$ if any protein in the projection of p on M_i interacts with any protein in the projection of q on M_i , for some $i \in \{1, 2\}$.

Goal 5 (Single Ancestral Component). Any value of $C(M_a) > 1$ implies that the module pair is not well defined as there is no evidence that the various ancestral components belong in the same module.

5 E_r SCORES AT RANDOM

Let M_1 and M_2 be modules with densities $\Delta(M_1)$ and $\Delta(M_2)$ generated uniformly at random so that the probability of an

TABLE 1
 E_r Scores Expected at Random

x/Δ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0.00	0.18	0.32	0.42	0.48	0.50	0.48	0.42	0.32	0.18	0.00
2	0.00	0.45	0.48	0.36	0.23	0.12	0.05	0.02	0.00	0.00	0.00
3	0.00	0.47	0.23	0.08	0.02	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.30	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.13	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

edge between any two proteins $p, q \in M_i$ is $\Delta(M_i)$. The average size of an ancestral protein is $(|M_1| + |M_2|)/|M_a|$. Let $x = (|M_1| + |M_2|)/(2|M_a|)$. Suppose for simplicity that all ancestral proteins have size $2x$ with x proteins from each interactome and that $\Delta = \Delta(M_1) = \Delta(M_2)$. Then, for $p, q \in M_a$, the probability of an interactome having no interaction between p and q is $(1 - \Delta)^{x^2}$. The probability of neither interactome having an interaction between p and q is $((1 - \Delta)^{x^2})^2$. The probability of an interactome having an interaction between p and q is $1 - (1 - \Delta)^{x^2}$ and the probability of both interactomes having an interaction between p and q is $(1 - (1 - \Delta)^{x^2})^2$. Thus, the probability of a disagreement in the relationship between p and q is $g(x) = 1 - ((1 - \Delta)^{x^2})^2 - (1 - (1 - \Delta)^{x^2})^2$, which is also equal to the expected value of E_r as $E(E_r) = \frac{E(\mathcal{R})}{\mathcal{R}_i} = \frac{g(x)\mathcal{R}_i}{\mathcal{R}_i} = g(x)$ where $\mathcal{R}_i = \binom{|M_a|}{2}$. Table 1 lists $E(E_r) = g(x)$ for various values of x and Δ . Nonzero entries are in bold. For all algorithms in this study, there are no protein losses, so $x \geq 1$.

6 EXPERIMENTS AND RESULTS

6.1 Experiments

Produlcs, NetworkBlast-M [18], Match-and-Split [6], and MaWISH [4] were applied to iRefIndex [19] binary interactions, Release 8.0, for *Homo sapiens* and *Drosophila melanogaster*, filtered to remove computationally predicted interactions and mapped to UniProtKB identifiers to remove isoforms. The interactomes consisted of 69,574 interactions on 12,652 proteins for *H. sapiens* and 38,699 interactions on 9,759 proteins for *D. melanogaster*. All programs were run with varying h threshold, corresponding to varying numbers of homologous protein pairs: $h = 10^{-80}$: 5,730 pairs, $h = 10^{-30}$: 29,598 pairs, $h = 10^{-20}$: 54,012 pairs, and $h = 10^{-9}$: 115,709 pairs. Because of the large number of module pairs returned by NetworkBlast-M, as shown in Fig. 2, all of which were assigned NetworkBlast-M quality scores, the set of modules for each h -threshold with highest NetworkBlast-M quality scores of the same size as the set returned by Produlcs was extracted and included in the comparisons.

The evaluation was performed on the module pairs returned having between 7 and 40 proteins per organism. This removes a significant fraction of the output from Match-and-Split and MaWISH that consists of subgraphs with two or three proteins, single edges or triangles, and it removes four large module pairs from MaWISH at threshold

$h = 10^{-30}$ with modules of size up to 78 proteins. This has little effect on NetworkBlast-M for which more than 99 percent of its modules have between 7 and 15 proteins per organism. Restricting the analysis to this size range allows meaningful comparison of the algorithms according to the various evaluation measures without the statistics being affected by very large or very small modules.

Graemlin has 19 network-specific parameters over a wide range of values, and together with the authors of Graemlin, we were unable to find settings that would yield results for the networks in this study. Graemlin 2.0 [20] was designed to address the parameter choice problem faced by Graemlin but requires data outside the scope of this study and presently has issues with usability.

6.2 Detailed Evaluation of Algorithms

In Fig. 2, the linear running time of Produlcs is seen to be very desirable. Match-and-Split could not complete on the data set with 29,598 homologous protein pairs and MaWISH could not complete on the data set with 54,012 homologous protein pairs after running for more than 12 hours. Fig. 3 shows that after restricting the output to modules having between 7 and 40 proteins per organism, the average sizes of modules from all algorithms are similar, though NetworkBlast-M has less variance in its size distribution than the other algorithms.

GO biological process enrichment was computed for all modules, separately for each organism, using Ontologizer [21] with Bonferroni correction for multiple hypothesis testing at 0.05 significance level. Fig. 4 shows that all algorithms perform similarly with Produlcs slightly outperforming MaWISH and NetworkBlast-M when considering full output sets, whereas NetworkBlast-M slightly outperforms Produlcs at some h -thresholds when considering only its top scoring sets. Fig. 2 shows that the top scoring sets from NetworkBlast-M at the higher h -thresholds have slightly more than half the coverage of Produlcs.

Fig. 4 shows that Produlcs returns modules with highest modularity followed by the top scoring sets from NetworkBlast-M. This indicates that modularity is correlated with GO enrichment. In Section 6.5, we show that several modules returned by Produlcs without GO enrichment at Bonferroni corrected 0.1 significance are biologically meaningful, demonstrating that conserved modularity is highly correlated with biological function. In Section 6.5, we describe how GO enrichment analysis may fail to report enrichment

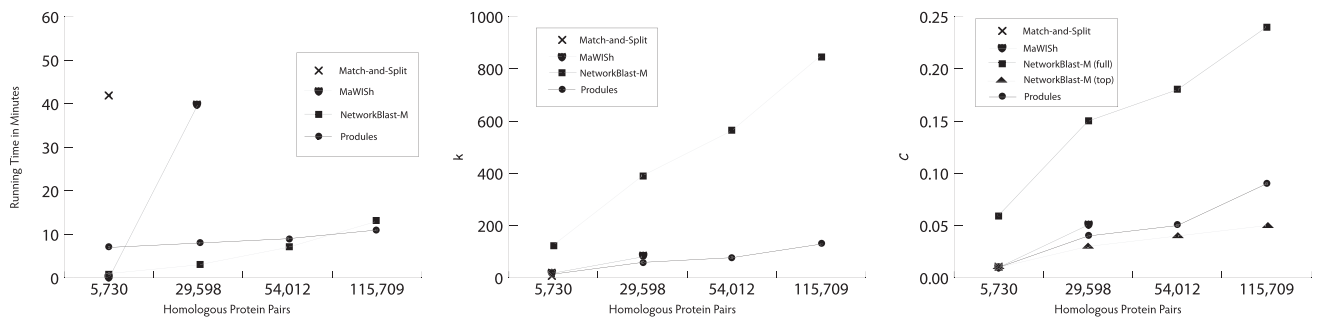


Fig. 2. Comparison of running time and basic characteristics. The x -axis is the number of homologous protein pairs. The y -axis, from left to right, is the running time, number of modules k , and coverage C .

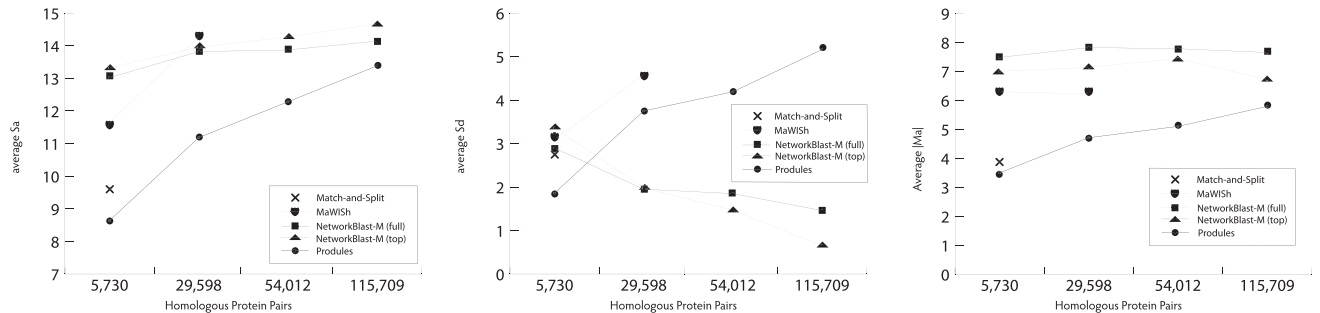


Fig. 3. Comparison of module sizes. The x -axis is as in Fig. 2. The y -axis, from left to right, is the average S_a , average S_d , and average $|M_a|$.

for biologically meaningful modules. More results from GO enrichment are given in Fig. 6, including the percent enriched at Bonferroni corrected 0.1 significance. Comparing with the results for 0.05 significance in Fig. 4 demonstrates that choosing an arbitrary threshold can lead to changes in relative performance, for example, the relative performance of MaWISH. In Fig. 6, we display average Bonferroni corrected GO enrichment p -values in a form that is independent of a threshold, calling the measures GE_a and GE_d , extending the definitions of module average and module difference, Definitions 10 and 11 in Section 4.3, to include GO enrichment p -values.

Fig. 5 shows that NetworkBlast-M produces many overlapping module pairs which appear in many cases to be similar to sliding a window across the modules. Produles focuses on optimizing module boundaries by using the definition of modularity and requiring evidence of module conservation in the interaction data, leading to lower overlap.

Fig. 7 shows that MaWISH, uniquely among the algorithms studied, produces modules with low E_r score due to its scoring model that rewards matching graph topologies. Using the random model in Section 5, estimating $x \approx (\text{average } S_a / \text{average } |M_a|) \approx 2$ using Fig. 3 and $\Delta \approx \Delta_a \in [0.2, 0.5]$ using Fig. 9, Table 1 shows that only MaWISH has an E_r score significantly lower than expected by the random model. All algorithms yield modules with large values of E_d , indicating that natural modules may include many proteins with homologous regions. None of the algorithms considered allow protein losses so E_l is zero for all algorithms.

Both Match-and-Split and Produles guarantee that $C_a = C(M_a) = 1$ and $C_d = 0$. Fig. 8 shows that MaWISH and NetworkBlast-M have high average values of $C(M_a)$, returning many modules similar to the diagram in Fig. 1(a). Figs. 8 and 9 show that NetworkBlast-M has

large values of C_a , C_d , and Δ_d , due to additivity of its scoring model in the interaction densities across organisms. NetworkBlast-M frequently aligns a dense module in one organism with a module that has zero or few interactions in the other organism. As indicated by Figs. 8 and 9, for many of these, similar to the diagram in Fig. 1(b), the interaction data does not support a claim of conservation.

6.3 Hierarchy of Modularity

To investigate the hierarchy of modularity, all module pairs from Produles with threshold $h = 10^{-9}$ were combined into composite modules. When two conserved modules had overlapping proteins in both interactomes, they were combined. Nineteen nonoverlapping composite modules were formed. The largest of these composite modules consists of 611 proteins in *D. melanogaster* and 843 proteins in *H. sapiens* with a modularity of $\mu = 0.19$ in *D. melanogaster* and $\mu = 0.18$ in *H. sapiens*. This conserved modular hierarchy from the largest conserved composite module contains 6.3 percent of the proteins in *D. melanogaster* and 6.7 percent of the proteins in *H. sapiens* with $|M_a| = 182$.

6.4 Modularity in Random Graphs

To examine the extent of conserved modularity in the current interactomes for *H. sapiens* and *D. melanogaster*, both interactomes were randomized while keeping protein sequence similarities fixed. The randomization step consisted of swapping the endpoints of a pair of edges chosen randomly without replacement. More precisely, if $(u_1, u_2), (u_3, u_4)$ are two randomly chosen edges in an interactome, these edges are replaced by the edges $(u_1, u_4), (u_3, u_2)$ unless the four endpoints are not distinct. This randomization maintains the degree of each vertex in the interactome. After all edges in the interactomes were randomized, the various algorithms for conserved module detection were applied to the resulting randomized graphs.

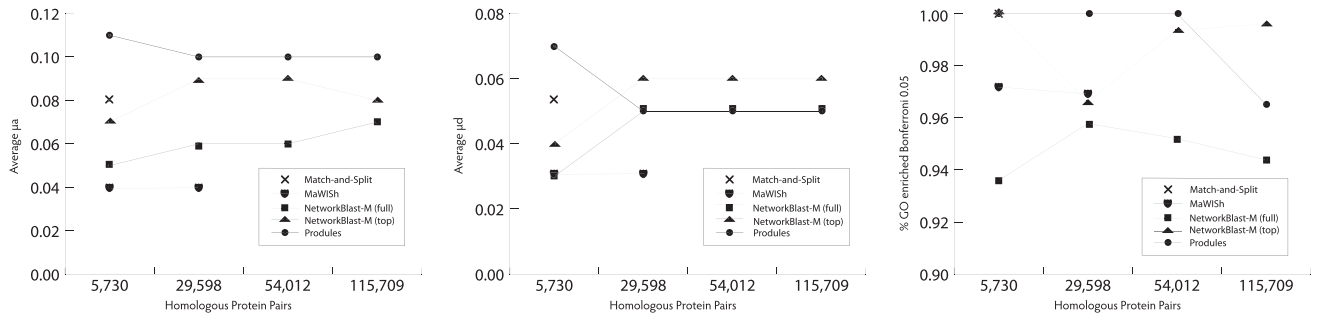


Fig. 4. Comparison of modularity and GO enrichment showing their correlation. The x -axis is as in Fig. 2. The y -axis, from left to right, is the average μ_a , the average μ_d , and the percent of modules enriched at 0.05 significance after Bonferroni correction.

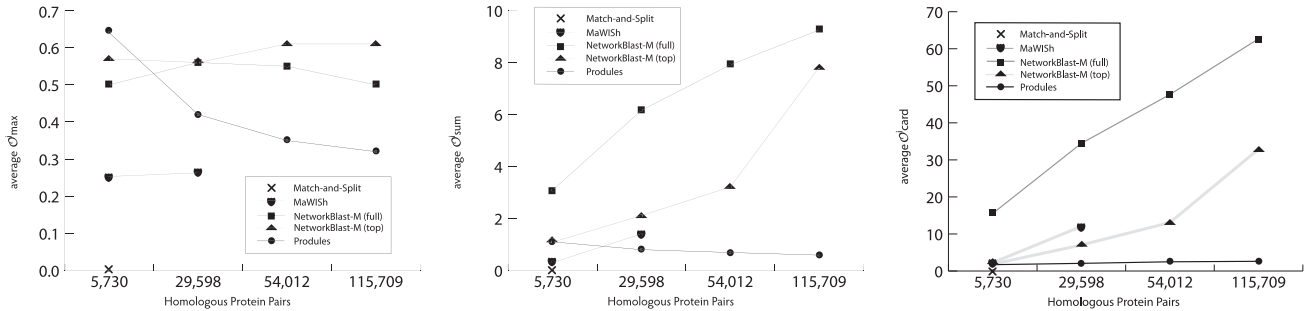


Fig. 5. Comparison of module overlap. The x -axis is as in Fig. 2. The y -axis, from left to right, is the average O_{max}^i , the average O_{sum}^i , and the average O_{card}^i .

With values of $d \geq 0.05$, Produles did not report any conserved module pairs in the random graphs. All other algorithms reported potential conserved modules in the random graphs but none with $\min\{\mu(M_1), \mu(M_2)\} \geq 0.05$ and $S_a \geq 7$. MaWISH with threshold $h = 10^{-30}$ reported 838 potential conserved modules in the random graphs, but none with $\mu_a \geq 0.04$ and $S_a \geq 7$. Match-and-Split with threshold $h = 10^{-80}$ reported five potential conserved modules in the random graphs, but none with $S_a \geq 7$ or with $\mu_a \geq 0.04$. NetworkBlast-M with threshold $h = 10^{-9}$ reported 107 potential conserved module pairs in the random graphs with average $S_a = 13.89$, the same size range as reported on the real interactomes. For these results from NetworkBlast-M, average μ_a was only 0.03. This comparison with random graphs indicates that conserved modularity is a defining characteristic of interactomes.

6.5 Multiprotein Modules without GO Enrichment

As shown in Fig. 6, when run with $h = 10^{-9}$, approximately 3 percent of multiprotein modules returned by Produles do not have GO enrichment at Bonferroni corrected 0.1 significance. This corresponds to eight multiprotein modules without significant GO enrichment. Most of these are multiprotein modules reported in the literature with boundaries very well detected by Produles. This section examines four of these conserved modules.

The four conserved modules we examine are from a recent experimental study [22] that compared the p53 family of tumor suppressor genes in *H. sapiens* with Dmp53, the sole p53-like protein in *Drosophila*. Through independent experiments in both organisms, coimmunoprecipitation in human and in vitro pull-down in *Drosophila*, this study found a collection of conserved proteins that interact with the p53 family in human and with Dmp53, indicating that Dmp53 has p53-like function. The h value for Dmp53 with human p73, the member of the

p53 family central in the human modules, is $h = 10^{-11}$, so these conserved modules are not detected by algorithms at lower h thresholds. These modules are not detected by NetworkBlast-M presumably because they have density at most $\Delta(M) = 0.18$ despite having modularity as high as $\mu_a = 0.08$. Additional RNAi experiments confirmed function of select module proteins in growth arrest of cancer cells indicating function of these modules in tumor suppression [22]. It is possible that these annotations have not yet been propagated to GO due to the recency of the study.

These four overlapping modules were combined into a composite module consisting of 22 proteins in *H. sapiens* and 25 proteins in *D. melanogaster*. This composite module has $\mu_a = 0.08$. Six interactions are from other studies and allow us to make predictions to refine the interaction topology of the coimmunoprecipitated complexes. We report here one example. Among the proteins found to coimmunoprecipitate with the p53 family are Asp/ASPM and Sqh/MYL9 [22]. The proteins Asp/ASPM regulate mitotic spindle formation in *Drosophila* and human, respectively. The proteins Sqh/MYL9 are myosin regulatory light chains in *Drosophila* and human, respectively, that have retained their ability to bind calcium. Indeed, Sqh/MYL9 are homologous to calmodulin. A yeast two-hybrid study found that Asp directly binds to calmodulin [23]. This indicates that Asp/ASPM may directly bind to Sqh/MYL9, which is why they were coimmunoprecipitated together, and that at most one binds directly to the p53 family.

Through analyses such as these and detailed follow-up direct interaction experiments, it may be possible to refine results from coimmunoprecipitation experiments to determine interaction topologies, which may allow an improved interaction level signal of evolutionary conservation in the form of lower E_r scores.

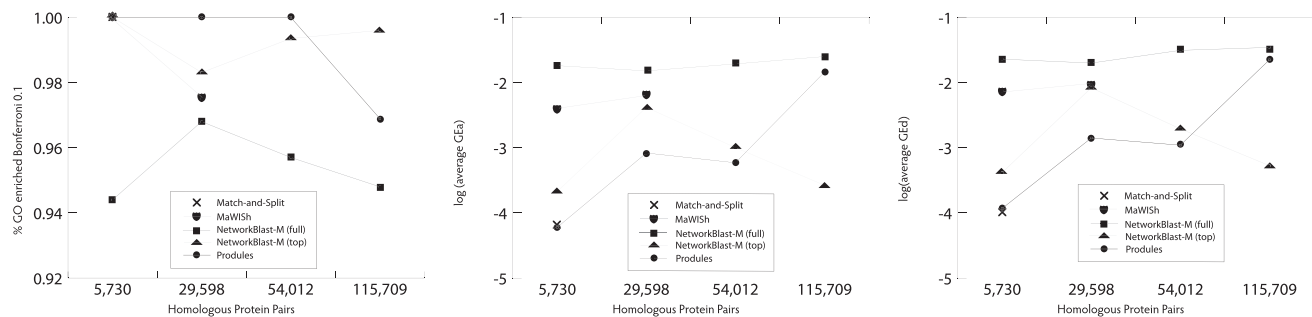


Fig. 6. Comparison of GO enrichment. The x -axis is as in Fig. 2. The y -axis, from left to right, is the percent of modules enriched at 0.1 significance after Bonferroni correction, the log of the average GO enrichment p -value averaged across organisms, and the log of the average difference in GO enrichment p -values between organisms.

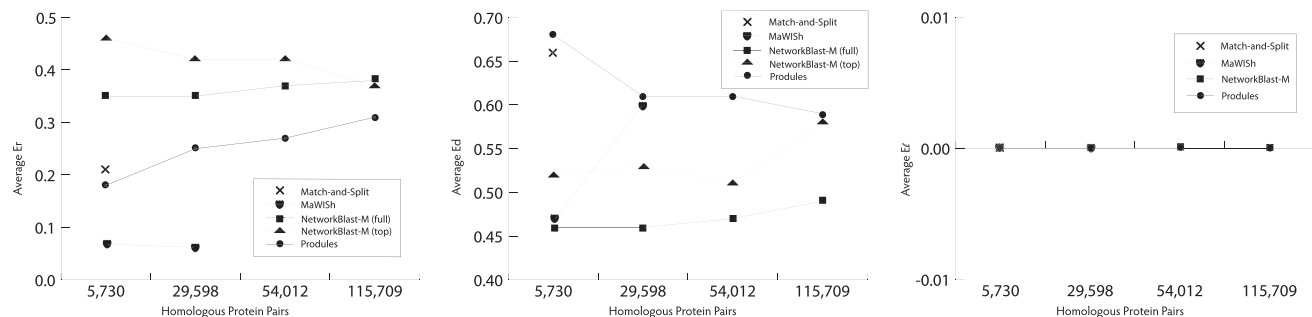


Fig. 7. Comparison using the interaction level evolutionary model. The x -axis is as in Fig. 2. The y -axis, from left to right, is the average E_r , the average E_d , and the average E_t .

6.6 Extent of Conserved Modularity

To examine comprehensively the extent of conserved modularity in the current interactomes for *H. sapiens* and *D. melanogaster*, all modules from NetworkBlast-M with threshold $h = 10^{-9}$ that satisfied $C_a = C(M_a) = 1$ and $\min\{\mu(M_1), \mu(M_2)\} \geq 0.05$, a total of 30 module pairs, were examined. The coverage was $C = 0.03$ with average modularity $\mu_a = 0.08$. As shown in Fig. 2, Produles detected coverage of $C = 0.09$ in conserved modules with $C_a = C(M_a) = 1$ and $\min\{\mu(M_1), \mu(M_2)\} \geq 0.05$, with average $\mu_a = 0.10$. When the sets detected by Produles and NetworkBlast-M were combined, the coverage remained $C = 0.09$ with a small increase in coverage, showing that the conserved modules detected by Produles contained nearly all proteins from the conserved modules detected by NetworkBlast-M. Comparing with Section 6.4 on random graphs, this shows that approximately 9 percent of interactome proteins, 9.6 percent in human and 8.9 percent in *Drosophila*, are included in conserved multiprotein modules with good evidence of conservation between the organisms. This can be compared with the number of proteins that have detectable sequence homology between the organisms, which at threshold $h = 10^{-9}$, includes 55 percent of interactome proteins in human and 60 percent of interactome proteins in *Drosophila*.

7 DISCUSSION

7.1 Modularity versus Density

Modularity indicates relatively dense regions separable from the rest of the graph. Density alone does not consider separability of dense regions. Several algorithms for related problems search for dense subgraphs [3], [5], [18]. Some recent studies on related problems, [11], [14], explicitly or implicitly are based on modularity.

7.2 Produles as Multiple Validation of Modularity

Produles requires validation of modularity by independent experiments across organisms. This validation depends on the existence of separate experiments across organisms. In earlier studies, [24], [25], we were not aware that iRefIndex included a small percentage of interactions that were computationally derived by interolog prediction. This led to the detection by Produles of isomorphic subgraphs from interolog prediction with low E_r scores. After removing the interactions predicted by interolog prediction, the E_r scores increased to the levels reported in this study.

7.3 Interaction Level Evolutionary Signal

Unfortunately, in current interactomics data sets, due to artifacts in the way experiments are performed and recorded, we may not see a strong evolutionary signal of similar ancestral protein relationships or similar graph topologies. Coimmunoprecipitation experiments are often considered more reliable than other assays, but in terms of the evolutionary signal at the individual interaction level, they are among the noisiest. A central protein may not directly interact with all proteins that are coimmunoprecipitated with it. Rather the interactions may be mediated by other proteins in the coimmunoprecipitated cluster. The true graph topology may be a linear path or some other topology rather than the star graph usually recorded in the interaction databases. This causes the interaction level signal of evolutionary conservation to be weak. However, as shown in this study, the module level signal of evolutionary conservation remains strong. As experimental coverage improves, guided by analysis techniques such as those described in Section 6.5, the interaction topologies may be improved, yielding lower E_r scores in conserved modules.

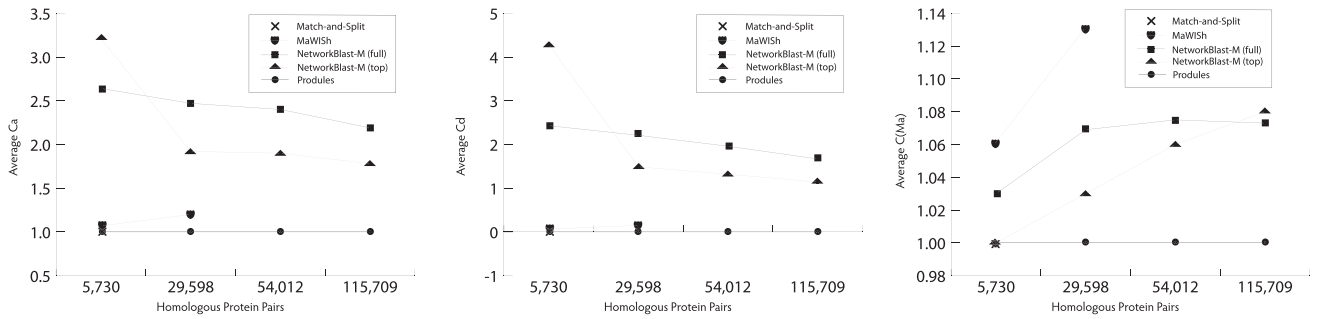


Fig. 8. Comparison of module boundaries. The x -axis is as in Fig. 2. The y -axis, from left to right, is the average C_a , average C_d , and average $C(M_a)$.

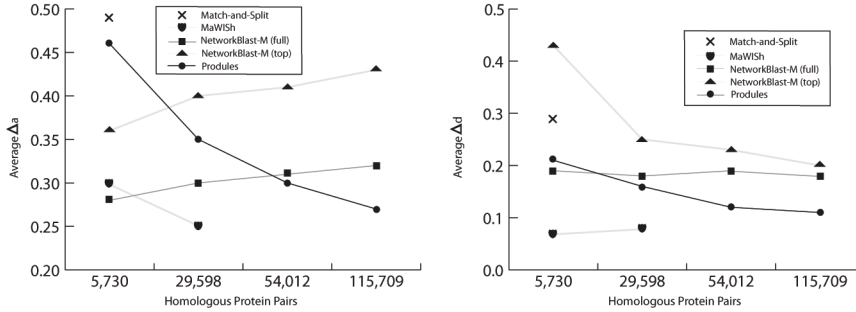


Fig. 9. Comparison of density and density balance. The x -axis is as in Fig. 2. The y -axis, from left to right, is the average Δ_a and average Δ_d .

7.4 Complementarity of Algorithms

Each algorithm examined in this study has different goals, and they can be considered complementary to each other. MaWISH may be useful when looking for regions with similar graph topologies and low E_r . These regions may be less noisy due to maintenance of an interaction level evolutionary signal, but often do not coincide with natural module boundaries. NetworkBlast-M may be useful when desiring a very large set of potential conserved modules that are frequently very dense in one of the organisms, may not have evidence of conservation in the protein interaction data, and may have $C(M_a) > 1$. Match-and-Split and Produles may be useful when desiring guarantees such as $C_a = C(M_a) = 1$. Match-and-Split is ideal when desiring a very high quality set of conserved modules with minimal overlap and when it is acceptable to expend large amounts of running time on small data sets. Produles is ideal for detecting conserved module boundaries using evidence from multiple validation in independent experiments across organisms, for fast running time with good scaling properties, and for examining the extent of conserved modularity in current interactomes. Produles is especially useful for detecting conservation of multiprotein modules that are not particularly dense.

8 EXTENSIONS

8.1 Three or More Interactomes

Detecting multiprotein modularity conserved across three or more interactomes requires only minor modification and maintains running time linear in the size of the input. Note that the size of the input is quadratic in the number of interactomes. Modularity maximization algorithms are applied to each interactome to find natural modules. After computing connected components on homologous proteins in the other interactomes and refining the resulting subgraphs as described in Section 3.3, the original module and all refinements are reported if they pass the

requirements in Section 3.2. The proof of running time in Section 3.4 extends without difficulty.

8.2 Weighted Interactomes

If desiring to focus on modules that preferentially incorporate particular interactions, weights can be assigned to the interactions. This has been used, for example, by NetworkBlast [3], to focus on interactions among proteins that have been found not only to interact but also to be coexpressed. The definition of modularity in this study is easily extended to weighted interactomes by using weight sums rather than edge counts in the definition of μ . If the weights are bounded from below, which is usually the case due to thresholding, a variant of Theorem 1 for weighted graphs holds, and the proof of linear running time follows.

Weights can also be used to count the number of independent experiments that report a given interaction. This is a form of multiple validation on the same proteins in the same organism. A disadvantage of this approach is its implicit down-weighting of interactions in regions of newly studied proteins that are frequently regions of greatest interest. Produles implicitly enforces multiple validation from independent experiments across organisms and across the various interactions in the module to increase confidence in a higher level signal of conserved modularity, making Produles ideal for noisy and newly generated interactomics data sets.

9 CONCLUSION

We present a definition of modularity for interactomes, a linear-time algorithm to detect conserved multiprotein modularity, and a new set of evaluation measures with associated goals. The measures introduced are sensitive to important issues not addressed by previous measures, and together provide a framework for analyzing the goals, characteristics, and performance of algorithms for this

important problem. Several algorithms are examined in depth. Conserved modularity is shown to be a defining characteristic of interactomes. Nearly 10 percent of proteins in current interactomes for *Homo sapiens* and *Drosophila melanogaster* are included in conserved multiprotein modules with good evidence for conservation.

ACKNOWLEDGMENTS

The authors would like to thank those who helped with the study: Manikandan Narayanan with Match-and-Split, Mehmet Koyutürk with MaWISH, Maxim Kalaev with NetworkBlast-M, Jason Flannick and Antal Novak with Graemlin, and Sabry Razick and Ian M. Donaldson with iRefIndex. The authors would also like to thank Pavol Jancura and Chaojun Li for helpful discussions regarding the study, and to thank Shuai Cheng Li and Bonnie Kirkpatrick for critical readings of the manuscript. This work is supported in part by US National Science Foundation grant IIS-0803937.

REFERENCES

- [1] M. Vidal, "Interactome Modeling," *FEBS Letters*, vol. 579, pp. 1834-1838, 2005.
- [2] B.P. Kelley, R. Sharan, R.M. Karp, T. Sittler, D.E. Root, B.R. Stockwell, and T. Ideker, "Conserved Pathways within Bacteria and Yeast as Revealed by Global Protein Network Alignment," *Proc. Nat'l Academy of Sciences USA*, vol. 100, no. 20, pp. 11394-11399, 2003.
- [3] R. Sharan, S. Suthram, R.M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R.M. Karp, and T. Ideker, "Conserved Patterns of Protein Interaction in Multiple Species," *Proc. Nat'l Academy of Sciences USA*, vol. 102, no. 6, pp. 1947-1979, 2005.
- [4] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, "Pairwise Alignment of Protein Interaction Networks," *J. Computational Biology*, vol. 13, no. 2, pp. 182-199, 2006.
- [5] J. Flannick, A. Novak, B.S. Srinivasan, H.H. McAdams, and S. Batzoglou, "Graemlin: General and Robust Alignment of Multiple Large Interaction Networks," *Genome Research*, vol. 16, pp. 1169-1181, 2006.
- [6] M. Narayanan and R.M. Karp, "Comparing Protein Interaction Networks via a Graph Match-and-Split Algorithm," *J. Computational Biology*, vol. 14, no. 7, pp. 892-907, 2007.
- [7] *Modularity: Understanding the Development and Evolution of Natural Complex Systems*, Vienna Series in Theoretical Biology, W. Callebaut and D. Rasskin-Gutman, eds. MIT Press, 2005.
- [8] P. Beltrao and L. Serrano, "Specificity and Evolvability in Eukaryotic Protein Interaction Networks," *PLoS Computational Biology*, vol. 3, no. 2, article e25, 2007.
- [9] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, "Basic Local Alignment Search Tool," *J. Molecular Biology*, vol. 215, no. 3, pp. 403-410, 1990.
- [10] H.A. Simon, "The Structure of Complexity in an Evolving World: The Role of Near Decomposability," *Modularity: Understanding the Development and Evolution of Natural Complex Systems*, Vienna Series in Theoretical Biology, W. Callebaut and D. Rasskin-Gutman, eds., pp. ix-xiii, MIT Press, 2005.
- [11] J. Wang, M. Li, J. Chen, and Y. Pan, "A Fast Hierarchical Clustering Algorithm for Functional Modules Discovery in Protein Interaction Networks," *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 607-620, May/June 2011.
- [12] R. Andersen, F. Chung, and K. Lang, "Using PageRank to Locally Partition a Graph," *Internet Math.*, vol. 4, no. 1, pp. 35-64, 2007.
- [13] D.A. Spielman and S. Teng, "A Local Clustering Algorithm for Massive Graphs and Its Application to Nearly-Linear Time Graph Partitioning," *CoRR*, article abs/0809.3232, 2008.
- [14] K. Voevodski, S. Teng, and Y. Xia, "Finding Local Communities in Protein Networks," *BMC Bioinformatics*, vol. 10, article 297, 2009.
- [15] M. Blum, R.W. Floyd, V. Pratt, R.L. Rivest, and R.E. Tarjan, "Time Bounds for Selection," *J. Computer and System Sciences*, vol. 7, no. 4, pp. 448-461, 1973.
- [16] J. Hopcroft and R. Tarjan, "Efficient Algorithms for Graph Manipulation," *Comm. ACM*, vol. 16, no. 6, pp. 372-378, 1973.
- [17] Gene Ontology Consortium, "The Gene Ontology in 2010: Extensions and Refinements," *Nucleic Acids Research*, vol. 38, Database Issue, pp. D331-D335, Jan. 2010.
- [18] M. Kalaev, V. Bafna, and R. Sharan, "Fast and Accurate Alignment of Multiple Protein Networks," *J. Computational Biology*, vol. 16, no. 8, pp. 989-999, 2009.
- [19] S. Razick, G. Magklaras, and I.M. Donaldson, "iRefIndex: A Consolidated Protein Interaction Database with Provenance," *BMC Bioinformatics*, vol. 9, article 405, 2008.
- [20] J. Flannick, A. Novak, C.B. Do, B.S. Srinivasan, and S. Batzoglou, "Automatic Parameter Learning for Multiple Network Alignment," *J. Computational Biology*, vol. 16, no. 8, pp. 1001-1022, 2009.
- [21] S. Bauer, S. Grossmann, M. Vingron, and P.N. Robinson, "Ontologizer 2.0—A Multifunctional Tool for GO Term Enrichment Analysis and Data Exploration," *Bioinformatics*, vol. 24, no. 14, pp. 1650-1651, 2008.
- [22] A. Lunardi et al., "A Genome-Scale Protein Interaction Profile of *Drosophila* p53 Uncovers Additional Nodes of the Human p53 Network," *Proc. Nat'l Academy of Sciences USA*, vol. 107, no. 14, pp. 6322-6327, Apr. 2010.
- [23] L. Giot et al., "A Protein Interaction Map of *Drosophila Melanogaster*," *Science*, vol. 302, pp. 1727-1736, 2003.
- [24] L. Hodgkinson and R.M. Karp, "Algorithms to Detect Multi-Protein Modularity Conserved during Evolution," Technical Report UCB/EECS-2011-7, EECS Dept., Univ. of California, Berkeley, 2011.
- [25] L. Hodgkinson and R.M. Karp, "Algorithms to Detect Multiprotein Modularity Conserved during Evolution," *Proc. Int'l Symp. Bioinformatics Research and Applications (ISBRA '11)*, pp. 111-122, 2011.



Luqman Hodgkinson received the BA degree in computer science from Hiram College in 2003, the MS degree in computer science from Columbia University in 2007, and is currently working toward the PhD degree in computational biology at the University of California, Berkeley, as a student of Professor Richard M. Karp.



Richard M. Karp attended Boston Latin School and Harvard University, receiving the PhD degree in 1959. From 1959 to 1968, he was a member of the Mathematical Sciences Department at IBM Research. From 1968 to 1994 and from 1999 to the present, he has been a professor at the University of California, Berkeley, where he held the Class of 1939 Chair and currently holds the position of University Professor. From 1995 to 1999, he was a professor at the University of Washington. From 1988 to 1995 and 1999 to the present, he has been a research scientist at the International Computer Science Institute in Berkeley. His current activities center on algorithmic methods in genomics and computer networking. He has more than 200 publications in books, journals, and conference proceedings, and has supervised more than 40 PhD dissertations. He has received a number of honors for his work, including: the US National Medal of Science, Turing Award, Kyoto Prize, Fulkerson Prize, Harvey Prize (Technion), Centennial Medal (Harvard), Lanchester Prize, Von Neumann Theory Prize, Von Neumann Lectureship, Distinguished Teaching Award (Berkeley), Faculty Research Lecturer (Berkeley), Miller Research Professor (Berkeley), Babbage Prize, and 10 honorary degrees. He is a member of the US National Academies of Sciences and Engineering, the American Philosophical Society, and the French Academy of Sciences, and a fellow of the American Academy of Arts and Sciences, the American Association for the Advancement of Science, the Association for Computing Machinery, and the Institute for Operations Research and Management Science.