Parallel Computing 34 (2008) 627–639

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

A graph-theoretic analysis of the human protein-interaction network using multicore parallel algorithms

David A. Bader, Kamesh Madduri *

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, United States

ARTICLE INFO

Article history:
Received 14 May 2007
Accepted 24 April 2008
Available online 9 July 2008

Keywords:
Human protein interactome
Complex networks
Betweenness centrality

ABSTRACT

Due to fundamental physical limitations and power constraints, we are witnessing a paradigm shift in commodity microprocessor architecture to multicore designs. Continued performance now requires the exploitation of concurrency at the algorithm level. In this article, we demonstrate the application of high performance computing techniques in systems biology and present multicore algorithms for the important research problem of protein-interaction network (PIN) analysis.

PINs play an important role in understanding the functional and organizational principles of biological processes. Promising computational techniques for key systems biology research problems such as identification of signaling pathways, novel protein function prediction, and the study of disease mechanisms, are based on topological characteristics of the protein interactome. Several complex network models have been proposed to explain the evolution of protein networks, and these models primarily try to reproduce the topological features observed in yeast, the model eukaryote interactome. In this article, we study the structural properties of a high-confidence *human* interaction network, constructed by assimilating recent experimentally derived interaction data. We identify topological properties common to the yeast and human protein networks.

Betweenness is a quantitative measure of centrality of an entity in a complex network, and is based on computing all-pairs shortest paths in the graph. A novel contribution of our work is the analysis of the degree–betweenness centrality correlation in the human PIN. Jeong et al. empirically showed that betweenness is positively correlated with the essentiality and evolutionary age of a protein. We observe that proteins with high betweenness, but low degree (or connectivity) are abundant in the human PIN. We have designed efficient and portable parallel implementations for the exact calculation of betweenness and other compute-intensive centrality metrics relevant to interactome analysis. For example, on the Sun Fire T2000 server with the UltraSparc T1 (Niagara) processor, we achieve a relative speedup of about 16 using 32 threads for a typical instance of betweenness centrality on a PIN, reducing the running time from nearly $3\frac{1}{2}$ min to 13 s.

Published by Elsevier B.V.

1. Introduction

Recent advances in high-throughput genomic experimental techniques have resulted in an abundance of diverse gene sequence and structure data. As a consequence, we are also faced with a significant volume of novel, unannotated gene products. The traditional methods of gene and protein annotation, such as homology-based transfer, are insufficient to characterize novel proteins, and are proving to be erroneous in many cases. This has led to a shift in research focus from the study of

* Corresponding author.
URLs: http://www.cc.gatech.edu/~bader (D.A. Bader), http://www.cc.gatech.edu/~kamesh (K. Madduri).

0167-8191/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.parco.2008.04.002

individual proteins to an integrative analysis of global characteristics and interactions between various cellular components using quantitative approaches. This research field, systems biology, has served as the foundation for the reconstruction of metabolic pathways, regulatory and signaling networks, and the identification of disease mechanisms. Protein function prediction is one of the key drivers for systems biology research. There are various approaches available for deducing the function of novel proteins, among which the study of interaction networks is one of the most promising techniques [40,10,41].

In order to design efficient computational techniques that are based on global connectivity patterns, it is essential to understand the topology of the network first. Genomic research in the past few years has enabled us to map high-confidence interactomes of model eukaryotes such as yeast [40,39], worm [24] and fly [16]. These protein interactions are mainly derived using the yeast two-hybrid (Y2H) assay technique, and have provided encouraging evidence that global topological structure and network features relate to known biological properties [20]. This has in-turn motivated several research groups to work on a global map of the *human interaction network*, and there have been several recent efforts on mapping the global human genome [34] using the Y2H assay. However, this system is prone to a high rate of false-positives and the interactions need to be validated with sophisticated techniques. Also, the identity of essential interactions in PINs differ significantly, depending on the experimental methodology [7]. The high-confidence interactions have been identified, filtered and are now readily available from online public domain databases (for example, BIND [1], DIP [35] and HPRD [28]). Most of these databases are literature-based and hand-curated with a sizable percentage of overlapping interactions.

The interaction networks of model eukaryotes such as yeast are analyzed extensively [43,22] using graph-theoretic and complex network analysis concepts. The yeast protein-interaction network (PIN) topology exhibits several interesting features that distinguish it from a random graph. For instance, the yeast PIN contains a larger number of highly connected (high degree) proteins than one would expect in a random Erdős-Rényi network. Jeong and Mason also observed that in the yeast network, the connectivity of a protein appears to be positively correlated with its essentiality [20], i.e., highly connected proteins tend to be more essential to the viability of the organism.

Large-scale network analysis is currently an active area of research in the social sciences [32,36], and several concepts from this field are being applied to computational biology. Important contributions from this field include analytical tools for visualizing networks [8,33], empirical quantitative indices to determine the key nodes in a network, and clustering algorithms [17]. Betweenness centrality [14] is a popular quantitative index that has been extensively used in recent years for the analysis of large-scale complex networks. Some applications include biological networks [20,29], study of sexual networks and AIDS [25], identifying key actors in terrorist networks [12], organizational behavior and transportation networks [18]. Joy et al. [22] report that in the yeast network, proteins with high betweenness are more likely to be essential, and that the evolutionary age of proteins is positively correlated with betweenness. Also, they observe that there are several proteins with low degree but high centrality scores in the yeast PIN.

Gandhi et al. [15] present a comprehensive analysis of a large-scale human interaction network [23,30]. They study a dataset of about 26,000 human protein interactions obtained from various public databases, compare the human interactome with the yeast, worm and fly datasets, and observe that only 42 interactions were common to all species. Also, they observe that unlike the yeast network, the available human PIN data does not support the presumption on the positive correlation between connectivity and essentiality.

We extend the work of Gandhi et al. [15] and Joy et al. [22] in this article. Our main contributions are the following:

- *Topological study of the largest human PIN constructed to date, comprising nearly 18,000 proteins and 34,000 interactions.* We analyze the global connectivity and clustering properties of a human PIN composed of high-confidence pairwise protein interactions. We do not model complex interactions as pairwise interactions, since it is not always known which proteins in the complex interact with each other.
- *Computation of centrality metrics for the human PIN.* We analyze betweenness centrality scores and find that proteins with high-betweenness centrality but low connectivity are abundant in the human PIN. We also observe that this finding cannot be explained by the widely-accepted models for scale-free networks.
- *Applying high performance computing techniques for large-scale PIN analysis.* Our efficient multicore implementation reduces the computation time of betweenness centrality to 13 s on 32 processors of the Sun Fire T2000 system, with a relative speedup of 16.

2. Preliminaries

2.1. Interactome datasets

There are several online databases devoted to the human interactome (see Table 1). In our previous study of the human PIN [5], we constructed the interaction map by merging information from Gandhi et al.'s human proteome analysis dataset [15] (updated February 2006), an interaction dataset from the Human Protein Reference Database [28] (updated May 2006), and IntAct (updated October 2006). The interaction network used in this article (referred to as HPIN throughout this article) is an updated dataset (January 2007) of binary interactions from HPRD. The latest version of the HPRD dataset includes interactions from MIPS, BIND, DIP and MINT. There is a complication using protein complex data (for example, from the MIPS

Table 1
Popular online human protein-interaction databases

Database	Details
HPRD [28]	Human Protein Reference Database. Experimentally verified protein–protein interactions obtained from manual curation of literature. 25,205 proteins and 37,581 interactions
BIND	Biomolecular Interaction Network Database. Collection of molecular interactions including high-throughput data submissions and hand-curated information from the scientific literature. 4644 human protein interactions
MIPS	Munich Information Center for Protein Sequences. 334 interactions
MINT	Molecular Interactions Database. 3544 interactions
IntAct [19]	Freely available, open source database system and analysis tools for protein-interaction data. European Bioinformatics Institute. 2420 interactions
OPHID	Online Predicted Human Interaction Database. Repository of already known experimentally derived human protein interactions, as well as 23,889 additional predicted interactions. This dataset is not included in our human PIN

database) to obtain protein interactions, since it is not always known which proteins in a complex interact with each other. Note that we *do not model* complex interactions as pairwise interactions in this study.

We also present the topological characteristics of two large-scale yeast PINs for comparison with HPIN. The yeast PIN from Jeong et al. [20] (YPIN) is an undirected network of 2112 proteins and 7182 interactions, while Reguly et al. [31] (YPIN2) provide an undirected network of 3289 proteins and 11,334 interactions. Both the network are well studied, and all the reported interactions are high-confidence ones.

2.2. Definitions

We represent the PIN as an undirected graph $G(V, E)$ in the analysis that follows. The set V represents the proteins, and E the set of interactions. The number of vertices and edges are denoted by n and m, respectively. The interaction networks are unweighted. Since the interaction networks are unweighted, and so we assume that each edge $e \in E$ has unit weight. A *path* from protein (vertex) s to t is a sequence of interactions (edges) $\langle u_i, u_{i+1}\rangle$, $0 \leqslant i \leqslant l$, where $u_0 = s$ and $u_l = t$. The *length* of a path is the sum of the weights of edges. We use $d(s,t)$ to denote the distance between vertices s and t (the minimum length of any path connecting s and t in G). Let us denote the total number of shortest paths between vertices s and t by σ_{st}, and the number passing through vertex v by $\sigma_{st}(v)$.

Betweenness centrality is a global shortest paths enumeration-based metric, introduced by Freeman [14]. Let $\delta_{st}(v)$ denote the *pairwise dependency*, or the fraction of shortest paths between s and t that pass through v: $\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}$. Betweenness centrality of a vertex v is defined as

$$\mathrm{BC}(v) = \sum_{s \neq v \neq t \in V} \delta_{st}(v).$$

This metric measures the control a vertex has over communication in the network, and can be used to identify key vertices in the network. High centrality indices indicate that a vertex can reach other vertices on relatively short paths, or that a vertex lies on a considerable fraction of shortest paths connecting pairs of other vertices.

3. Parallel computation of betweenness centrality

A straight-forward way of computing Betweenness Centrality is to augment a single-source shortest path algorithm such as Dijkstra's algorithm to compute the pairwise dependencies. Define a set of *predecessors* of a vertex v on shortest paths from s as $pred(s,v)$. Now each time an edge $\langle u, v\rangle$ is scanned for which $d(s,v) = d(s,u) + d(u,v)$, that vertex is added to the predecessor set $pred(s,v)$. Setting the initial condition of $pred(s,v) = s$ for all neighbors v of s, we can proceed to compute the number of shortest paths between s and all other vertices. The computation of $pred(s,v)$ can be easily integrated into breadth-first search (BFS) for unweighted graphs.

To exploit the sparse nature of typical real-world graphs, Brandes [11] gives an algorithm that computes the betweenness centrality score for all vertices in the graph in $O(mn)$ time for unweighted graphs. The main idea is as follows. We define the *dependency* of a source vertex $s \in V$ on a vertex $v \in V$ as $\delta_s(v) = \sum_{t \in V} \delta_{st}(v)$. The betweenness centrality of a vertex v can be then expressed as $BC(v) = \sum_{s \neq v \in V} \delta_s(v)$. It can be shown that the dependency $\delta_s(v)$ satisfies the following recursive relation: $\delta_s(v) = \sum_{w: v \in pred(s,w)} \frac{\sigma_{sv}}{\sigma_{sw}}(1 + \delta_s(w))$.

The algorithm is now stated as follows. First, n BFS computations are done, one for each $s \in V$. The predecessor sets $pred(s,v)$ are maintained during these computations. Next, for every $s \in V$, using the information from the shortest paths tree and predecessor sets along the paths, compute the dependencies $\delta_s(v)$ for all other $v \in V$. To compute the centrality value of a vertex v, we finally compute the sum of all dependency values. The computational complexity of the algorithm is $O(mn)$ and the space requirements are $O(m+n)$.

We present novel parallel algorithms for exactly computing betweenness and other centrality measures in [4]. Algorithm 1 outlines the general approach for unweighted graphs. On each BFS computation from s, the queue Q stores the current set of vertices to be visited, S contains all the vertices reachable from s, and $P(v)$ is the predecessor set associated with each vertex $v \in V$. The arrays d and σ store the distance from s, and shortest path counts, respectively. The centrality values

are computed in steps 22–25, by summing the dependencies $\delta(v)$, $v \in V$. The final scores need to be divided by two if the graph is undirected, as all the shortest paths are counted twice.

We observe that parallelism can be exploited at two levels:

- The BFS/SSSP computations from each vertex can be done concurrently, provided the centrality running sums are updated atomically.
- The actual BFS/SSSP can be also be parallelized. When visiting the adjacencies of a vertex, edge relaxation can be done concurrently.

We will refer to the parallelization approach that concurrently computes the shortest path trees (steps 3–25 in Algorithm 1) as the *coarse-grained* parallel betweenness centrality algorithm, and the latter approach in which a single BFS/SSSP traversal is parallelized, as the *fine-grained* algorithm.

There are performance trade-offs associated with both these algorithms when implemented on parallel systems. The coarse-grained algorithm assigns each processor a fraction of the vertices from which to initiate SSSP computations. The vertices can be assigned dynamically to processors, so that work is distributed as evenly as possible. For this approach, graph traversal requires no synchronization, and the centrality metrics can be computed exactly provided they are accumulated atomically (step 25 in Algorithm 1). Alternately, each processor can store its partial sum of the centrality score for every vertex, and all the sums can be merged using an efficient global reduction operation. However, the problem with the coarse-grained algorithm is that the auxiliary data structures – the stack S, list of predecessors P, and the BFS queue Q – need to be replicated on each processor for doing concurrent traversals. The memory requirements scale as $O(p(m+n))$, and this approach becomes infeasible for large-scale graphs.

In the fine-grained algorithm, we parallelize each BFS/SSSP computation, and the memory requirement is $O(m+n)$. Note that every centrality computation iteration is composed of two phases, the BFS tree computation (steps 3–18), and accumulation of centrality scores (steps 19–24). In previous work, we discuss algorithms and efficient implementations for fine-grained parallel BFS [3] and single-source shortest paths [26,13]. These algorithms can be directly applied for parallelizing the traversal phase. In the subsequent phase, we need to visit vertices in the non-increasing order of their distance from the source vertex. Real-world social and technological networks typically demonstrate the small-world property, i.e., the graph diameter is usually a constant value, or in some cases $O(\log n)$. Thus, there will be a significant number of vertices at a given depth from the source vertex, and the centrality scores of all these vertices can be accumulated in parallel.

Algorithm 1: Parallel betweenness centrality for unweighted graphs

Input: $G(V, E)$
Output: Array $BC[1..n]$, where $BC[v]$ gives the centrality metric for vertex v

```
1  for all v ∈ V in parallel do
2      BC[v] ← 0;
   for all s ∈ V in parallel do
3      S ← empty stack;
4      P[w] ← empty list, w ∈ V;
5      σ[t] ← 0, t ∈ V; σ[s] ← 1;
6      d[t] ← −1, t ∈ V; d[s] ← 0;
7      Q → empty queue;
8      enqueue s ← Q;
9      while Q not empty do
10         dequeue v ← Q;
11         push v → S;
12         for each neighbor w of v in parallel do
13             if d[w] < 0 then
14                 enqueue w → Q;
15                 d[w] ← d[v] + 1;
16             if d[w] = d[v] + 1 then
17                 σ[w] ← σ[w] + σ[v];
18                 append v → P[w];
19     δ[v] ← 0, v ∈ V;
20     while S not empty do
21         pop w ← S;
22         for v ∈ P[w] do
23             δ[v] ← δ[v] + (σ[v]/σ[w])(1 + δ[w]);
24         if w ≠ s then
25             BC[w] ← BC[w] + δ[w];
```

3.1. Parallel multicore performance

The sequential complexity for computing betweenness centrality and other shortest-path-based centrality metrics is $O(mn)$. The parallel algorithms for betweenness centrality described in the prior section are well suited for implementation on multicore and multiprocessor systems that have high memory bandwidth and a modest number of processor cores. While betweenness is compute-intensive, finding the clustering coefficients, assortativity, the joint degree distribution, and other topological measures are straight-forward to compute with linear-work algorithms. Portable, efficient implementations of these algorithms are freely available from our website as part of the SNAP (Small-world Network Analysis and Partitioning) framework [6]. As a representative case study for performance on multicore systems, we will present results of computing betweenness, closeness centrality, and diameter for HPIN in this section.

We report performance results on the Sun Fire T2000 multicore server, with the Sun UltraSPARC T1 (Niagara) processor. This system has eight cores running at 1.0 GHz, each of which is four-way multithreaded. There are eight integer units with a six-stage pipeline on chip, and four threads running on a core share the pipeline. The cores also share a 3 MB L2 cache, and the system has a main memory of 16 GB. Since there is only one floating point unit (FPU) for all cores, the UltraSparc T1 processor is mainly suited for programs with few or no floating point operations. We compile our codes with the Sun C compiler v2.8 and the flags `-xtarget=ultraT1 -xarch=v9b -xopenmp`.

Fig. 1 plots the execution time and relative speedup achieved on the Sun Fire T2000 for the exact computation of betweenness centrality. The performance scales nearly linearly up to 16 threads, but drops between 16 and 32 threads. This can be attributed to insufficient memory bandwidth on 32 threads, as well as the presence of only one floating point unit on the entire chip. We use the floating point unit for accumulating pair dependencies and centrality values. The execution times for other shortest path-based centrality metrics such as closeness (Fig. 2) and betweenness centrality differ by a constant multiplicative factor. Betweenness centrality computation is much more involved, as it requires maintaining a BFS stack, a queue and a predecessor list. Also, the BFS tree is traversed twice in the algorithm. For additional performance results and a discussion of scalability related to network properties, please refer to [2,4].

Computing the graph diameter and average path length also have a $O(mn)$ work complexity, and the parallel approach is very similar to betweenness centrality. In both cases, we need to run n breadth-first searches. We report performance results for computing the graph diameter on a dual-core 2.8 GHz Intel Xeon system. The two cores share a 2 MB L2 cache, and the system has 4 GB physical memory. Each processor is also equipped with hyper-threading, which gives an impression of four virtual processors as a whole. The code is compiled with the Intel C Compiler v9.1 and the flags (`-O3 -ipo -unroll_loops`). Fig. 3 plots the execution time and speedup as the number of threads is varied from 1 to 4. We achieve near-linear speedup up to 4 threads in this case.

4. Biological analysis

In this section, we will quantify *connectivity*, *centrality* and *clustering* in the human and yeast PINs using relevant social network analysis metrics.

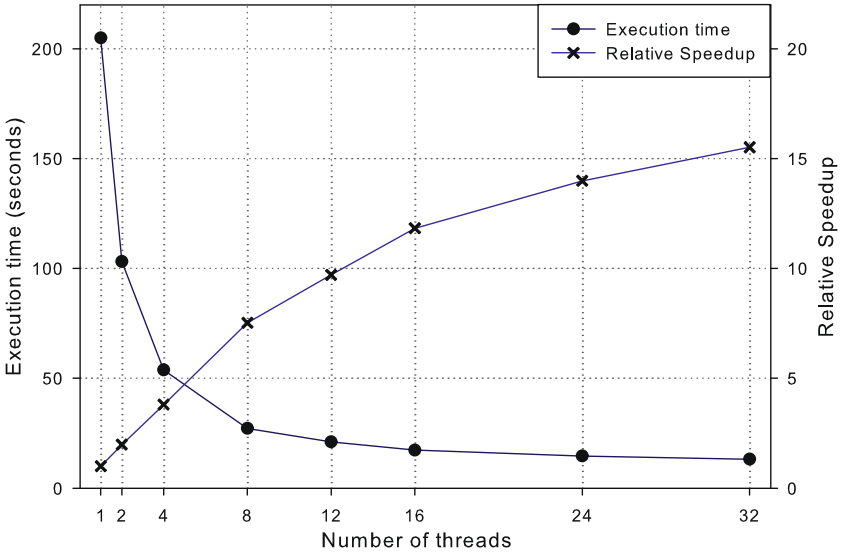

Fig. 1. Execution time and speedup on the Sun Fire T2000 system for the HPIN betweenness centrality computation.

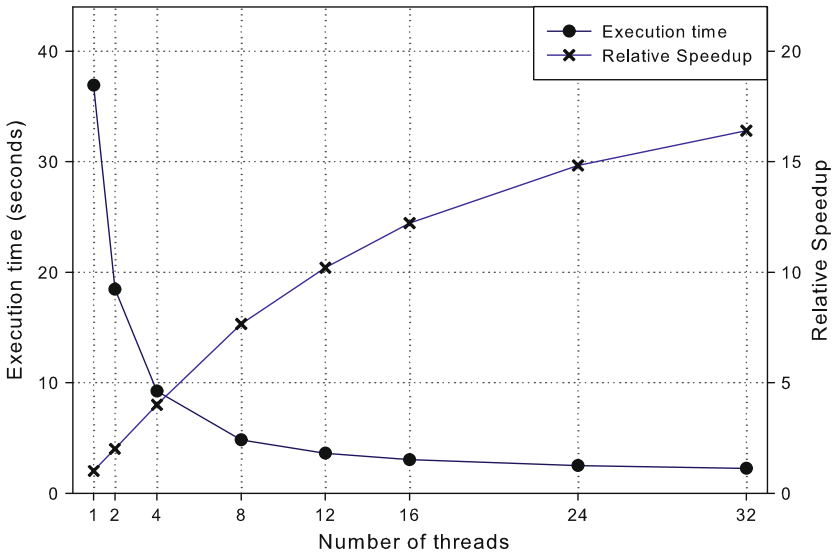

Fig. 2. Execution time and speedup on the Sun Fire T2000 system for the HPIN betweenness centrality computation.

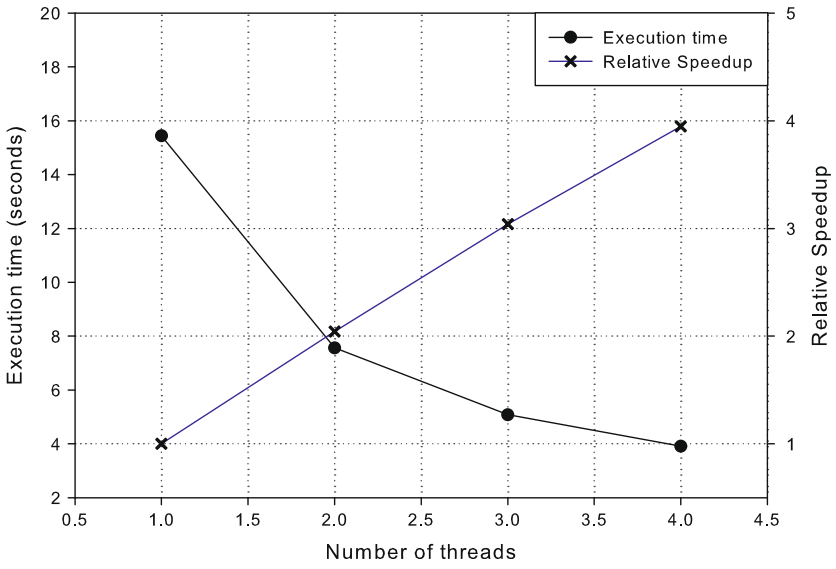

Fig. 3. Execution time and Speedup on a dual-core Intel Xeon system for graph diameter computation.

The HPIN dataset we obtain from HPRD [28] has 18,755 proteins and 34,367 pairwise interactions. We process this network and extract the largest connected component, which is composed of 8503 proteins and 32,191 interactions. Thus, the original dataset includes around 10,000 non-interacting proteins. Apart from the large component, there are 89 connected components of size 2, 16 connected components of size 3 and 5 components of size 5. We also ignore complex interactions in the dataset.

4.1. Connectivity

Fig. 4a plots the degree distribution of the vertices in the largest component of HPIN. We report the normalized frequency count $p(k) = \frac{n(k)}{n}$, where $n(k)$ is the total number of degree-k vertices. Similarly, Fig. 4b gives the degree distribution of the yeast PIN. In HPIN, nearly 25% of vertices have a degree of 1, and 15% are of degree-2. In comparison, 31% of vertices are of degree-1 and 15% are degree-2 in YPIN. The degree distribution can be roughly approximated by a power-law, but with a heavy tail.

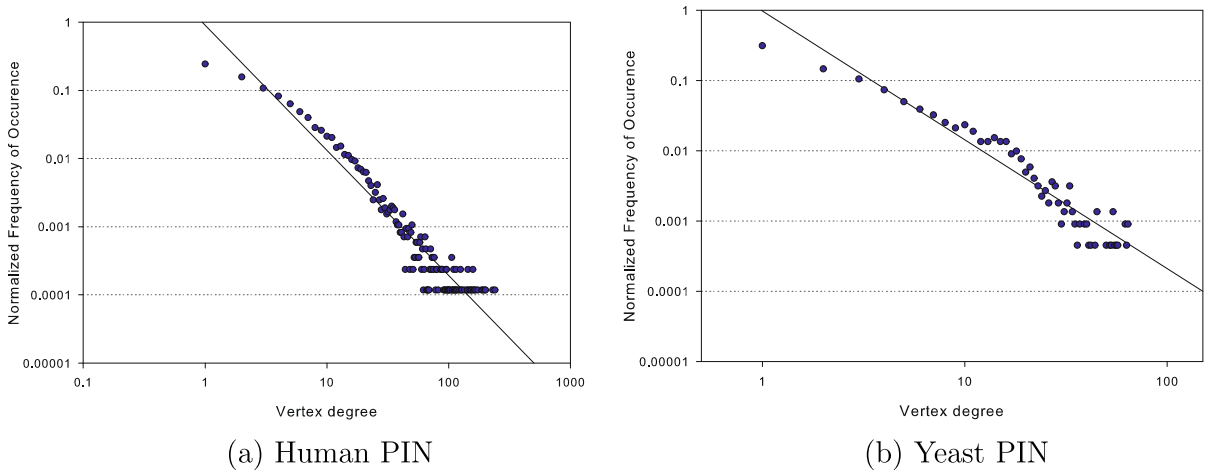

Fig. 4. Degree distributions of protein-interaction networks.

The protein with the highest degree in HPIN is TP53 (230 interactions). TP53 stands for tumor-protein 53 – it regulates the cell division process by keeping cells from growing and dividing too fast, or in an uncontrolled way. The p53 tumor protein is located in the nucleus of cells throughout the body and can bind directly to DNA. Since the p53 tumor protein is essential for regulating cell division, it is also known as the *guardian of the genome*.

4.2. Clustering

We calculate the average clustering coefficient, a measure of the tendency of proteins in a network to form clusters or groups. For a vertex v of degree d, the clustering coefficient CC is defined as the $CC(v) = \frac{2k}{d(d-1)}$, where k is the number of links connecting the d neighbors of v, considered pairwise. The average clustering coefficient $CC_a(d)$ for a particular degree d is simply the average of the clustering coefficients of all vertices of degree d. We find that, on an average, CC_a is a constant value of 0.1 for both HPIN and YPIN (see Fig. 5). This is a strong indicator that these networks *do not* have a hierarchical organization, or trivially noticeable community structure [17]. Newman [27] observes that networks with high clustering coefficients are prone to virus outbreaks, and faster epidemic spreading. A constant average clustering coefficient across the network might be one of the distinguishing features of PINs. Also, we observe that low-degree vertices in HPIN and YPIN show some variation in the clustering coefficient values, whereas high-degree vertices (degree greater than 20) show little or no variation (see error bars in Fig. 5). Also, note that we do not model protein complexes as pairwise interactions in this study. Complex interactions lead to an occurrence of proteins of sizable degree as well as high clustering coefficients in the graph.

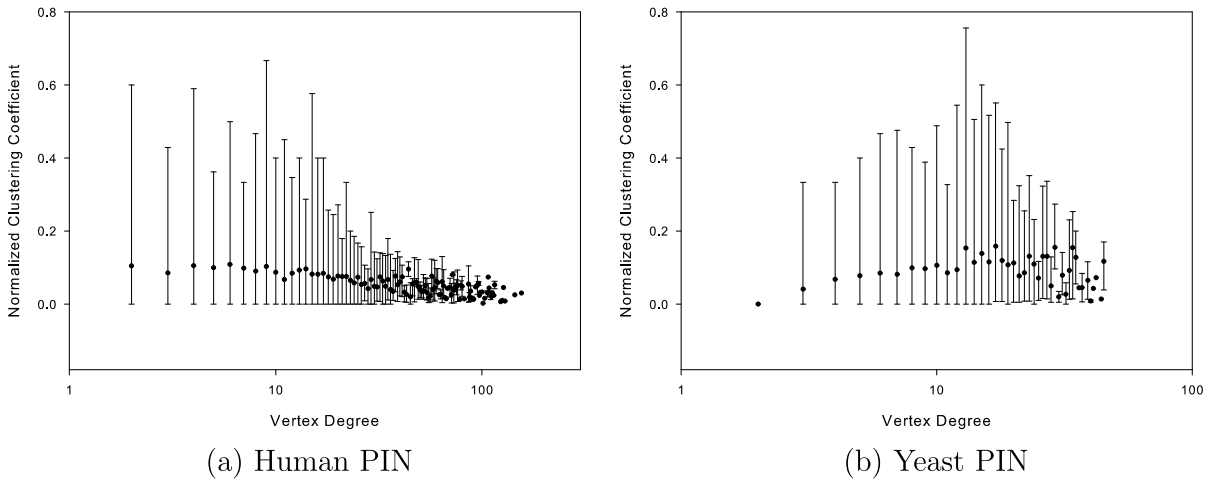

Fig. 5. Average clustering coefficient for degree-k vertices (the error bars indicate the maximum and minimum values).

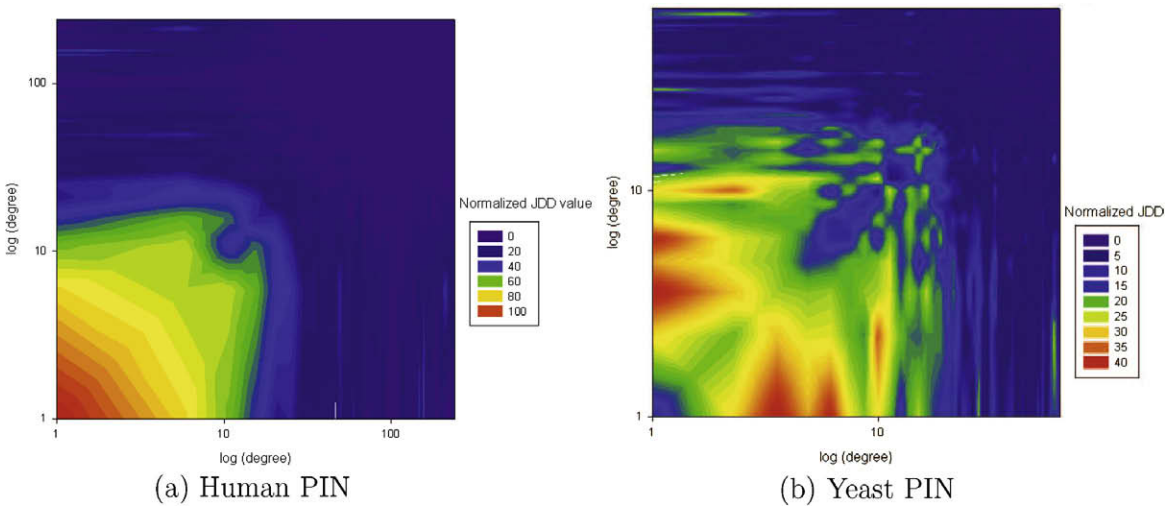

Fig. 6. Joint degree distribution.

There are several metrics to study correlations between vertex degree and the connectivity of neighbors of that vertex. Based on extensive empirical studies, Newman [27] proposes a simple classification of networks into three classes: assortative, disassortative, and neutral mixing [27]. A vertex with high degree in an assortative (disassortative) network tends to connect to nodes with other higher (low) degree vertices, whereas in a neutral mixing there are no such patterns. Disassortative networks are vulnerable to both random failures and targeted attacks at the high-degree vertices. Other metrics, such as likelihood, radial, and tangential, are also directly related to assortativity. Tangential links are used to refer to edges connecting vertices of similar degrees, and radial links refer the links connecting high-degree vertices with low-degree ones. On calculating Newman's assortativity coefficient, we found that both HPIN and YPIN exhibit mildly disassortative to neutral mixing.

The joint degree distribution (JDD) is another metric similar to assortativity that is used to study clustering. JDD is the probability that a randomly selected edge connects vertices of degree k_1 and k_2, respectively, $P(k_1, k_2) = \frac{m(k_1,k_2)}{m}$, where $m(k_1, k_2)$ is the total number of edges between vertices of degree k_1 and k_2, respectively. We plot the joint degree distributions of HPIN and YPIN in Fig. 6. In HPIN, we find that the majority of edges are links between low-degree vertices (bottom-left corner). For yeast, there are some links connecting high-degree vertices with low-degree ones (bottom-right and top-left). However, observe that there are no tangential links between high-degree vertices (top-right). This is an important network characteristic for which the PIN topology differs significantly from a physical network, such as the AS-level network topology.

4.3. Centrality

Next we study centrality and criticality in PINs. The problem of identifying central nodes in large complex networks is of fundamental importance with applications in several areas varying from computational biology to social and business networks. Many quantitative metrics for this purpose have originated from the social network analysis community, commonly referred to as *centrality measures*. We will use the Betweenness centrality metric to analyze the human interactome. Researchers have paid particular attention to the relation between centrality and *essentiality* or *lethality* of a protein (for instance [20]). A protein is said to be essential if the organism cannot survive without it. Essential proteins can only be determined experimentally, so alternate approaches to *predicting essentiality* are of great interest and have potentially significant applications such as drug target identification [21]. Previous studies on yeast have shown that proteins acting as hubs (or high-degree vertices) are three times more likely to be essential [20]. So we wish to analyze the interplay between degree and centrality scores for proteins in the human PIN in this section.

Fig. 7 plots the normalized betweenness centrality scores (absolute centrality score divided by $(n-1)*(n-2)/2$, the highest possible centrality score for a vertex in an undirected network) of all the proteins in HPIN, ordered by degree. We repeat the analysis for both the yeast datasets (see Fig. 8). In all the cases, we observe that there is a strong correlation between the degree of a vertex and its betweenness centrality score. All highly connected vertices have high centrality scores. However, observe that low-degree vertices show a significant variation in their centrality scores. The protein with the highest degree (p53) also has the highest centrality score.

We now try to explain the variation in the centrality scores of low-degree vertices. Clearly, all degree-1 vertices have a centrality score of zero. We would like to determine the connectivity patterns of high centrality, low-degree vertices in the

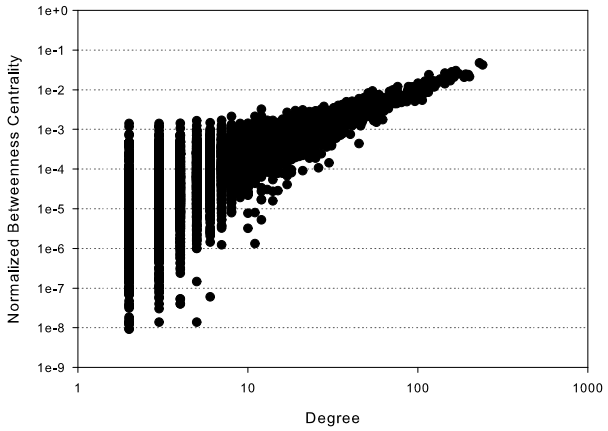

Fig. 7. Normalized betweenness centrality vs. degree in HPIN.

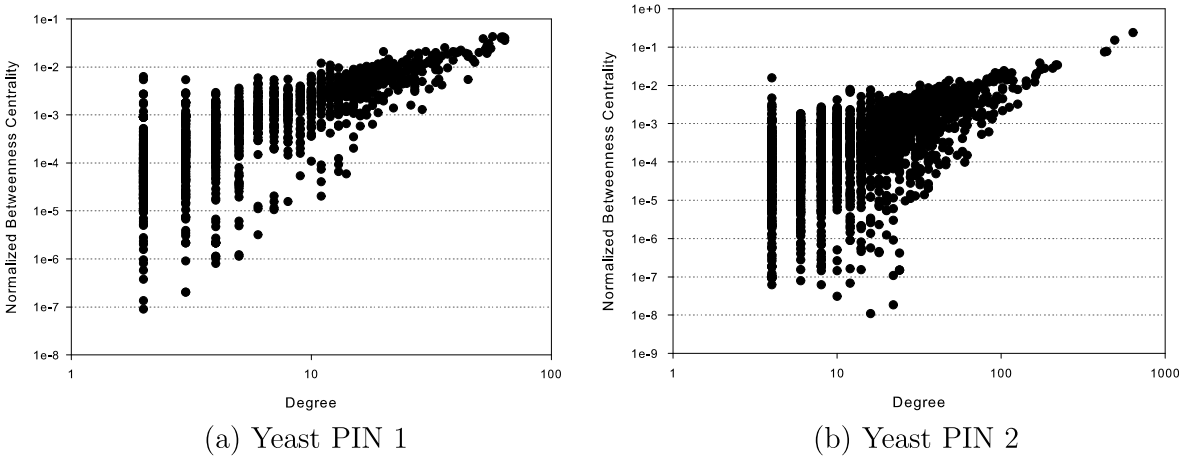

Fig. 8. Normalized betweenness centrality vs. degree.

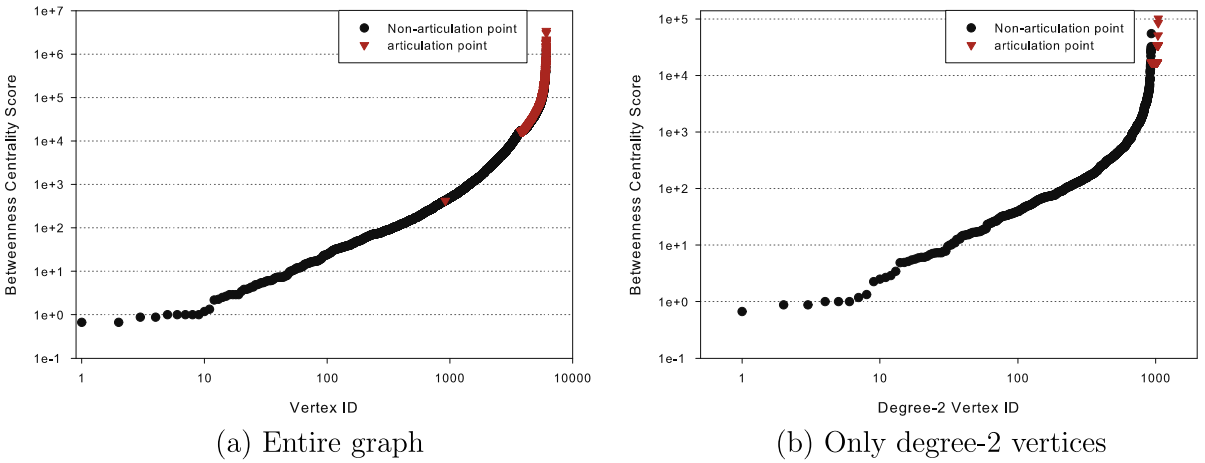

Fig. 9. Betweenness centrality scores of articulation and non-articulation vertices.

graph. For this purpose, consider decomposing the graph into its biconnected components. These are the maximal subsets of vertices such that the removal of a vertex from a particular component will not disconnect the component. Unlike connected

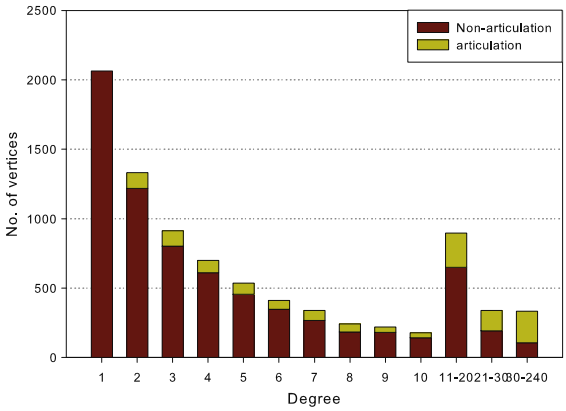

Fig. 10. Percentage of articulation points in HPIN.

components, vertices may belong to multiple biconnected components: vertices that belong to more than one biconnected component are called *articulation points* or, equivalently, *cut vertices*. A graph without articulation points is biconnected. Fig. 9 replots betweenness centrality scores of vertices in the graph, indicating articulation points separately this time. We observe that *nearly all articulation vertices have high centrality scores*. We can filter a significant percentage of vertices in the graph based on the observation that low-degree vertices that are not articulation points have low centrality scores, and are unlikely to be critical.

Fig. 10 plots the percentage of articulation points vs. degree in the Human PIN. Observe that 20% of all low-degree vertices are articulation points. Similarly, a high percentage of high-degree vertices are again articulation points. We can filter non-articulation low-degree vertices and high-degree articulation vertices, as the centrality scores for these vertices are predictably low and high, respectively.

We now plot betweenness centrality scores in the pruned graph (after removing non-articulation low-degree vertices). The average centrality scores for a given vertex degree are indicated on a linear scale in Fig. 11, along with the maximum and minimum values. Observe that centrality scores only vary by two orders of magnitude in this case. For a given degree, the centrality scores vary by an order of magnitude in both HPIN and YPIN. The centrality score variation is slightly higher in YPIN.

Based on analysis of the betweenness measure, we identify a new topological feature in the yeast and Human PINs that may not found in randomly generated scale-free networks: a significant percentage of proteins characterized by high betweenness (one order of magnitude less than the maximum), yet low connectivity. The fact that a majority of these proteins are also articulation points indicates suggests that these proteins may represent important that link components with a low degree of clustering.

We now investigate whether synthetic models for network evolution reproduce this low-degree, high-betweenness behavior. To address this question, we analyzed different computational models of biological network evolution that gener-

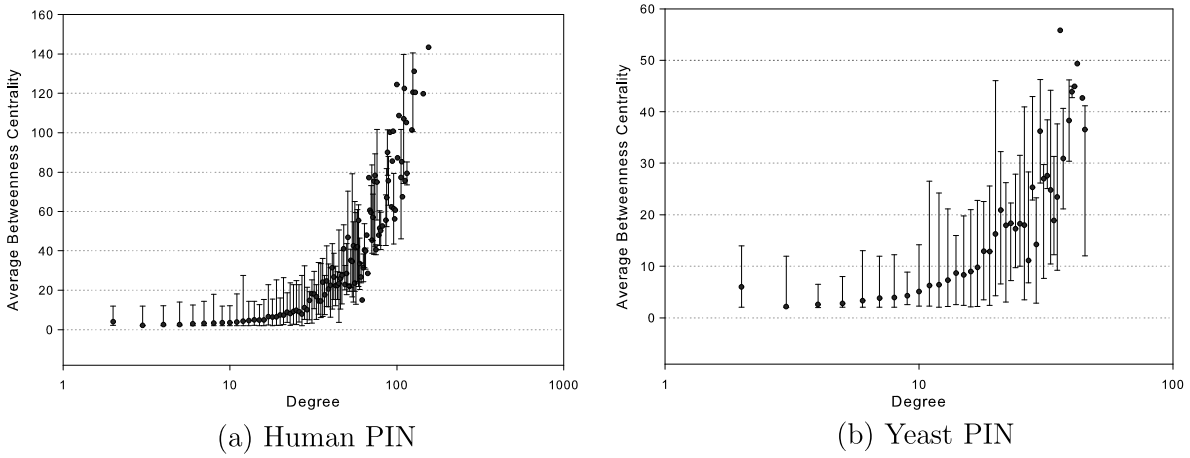

Fig. 11. Average betweenness centrality vs. degree after removing low-degree non-articulation vertices (the error bars indicate the maximum and minimum values).

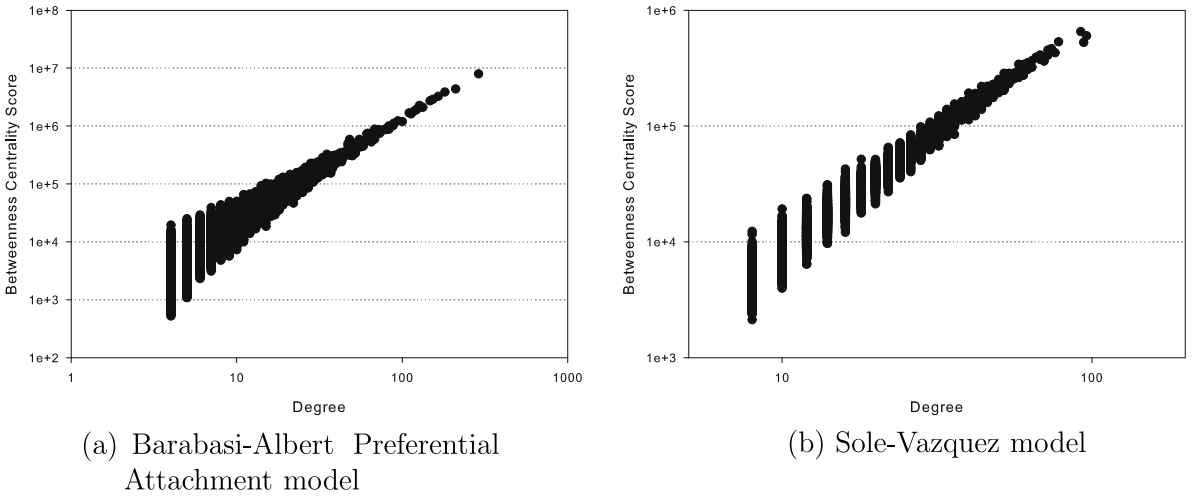

Fig. 12. Degree–betweenness correlation for synthetic graphs.

ate scale-free networks. We experimented with a range of parameters for these models and selected ones that gave a power-law distribution that matched the slope of HPIN. In each case, we quantified the variation of betweenness for a particular connectivity, and its change for the value of connectivity. Thus, an increase in the standard deviation of betweenness values for low-degree values indicated the presence of high centrality, low-degree proteins.

The simplest generative algorithm, first proposed by Barabasi and Albert (BA preferential attachment model) to explain the power-law distribution of connectivity, does not predict the existence of low-degree, high centrality vertices. Betweenness and degree are almost linearly correlated in this graph (see Fig. 12a). Also, the extended Barabasi–Albert (EBA) model, where link addition and rewiring occur along with node addition with preferential attachment, also did not produce networks with this characteristic, although low-degree vertices showed a better spread in this case. Moreover, this algorithm has no biological basis.

A biologically motivated model put forward by Sole et al. [38] and Vazquez et al. [42] incorporated *gene duplication* as the driving mechanism for genome growth. In this model, the existing nodes (proteins) are copied with all their existing links, followed by divergence of the duplicated nodes introduced by rewiring and/or addition of connections, imitating mutations of duplicated genes. For the model parameter range that produces power-law networks, the Sole–Vazquez (SV) model also failed to produce the same bias towards betweenness exhibited by HPIN (see Fig. 12b). These results show that existing evolutionary algorithms that produce scale-free networks do not predict the existence of high-betweenness, low-degree vertices found within the yeast and human PINs.

Joy et al. [22] propose a model to explain a similar occurrence in the yeast PIN. They present a new model based on the Berg model [9], which considers point mutations in addition to gene duplication. With this model, the authors generate a synthetic network that matches the low degree, high centrality property in the yeast PIN. However, the paper [22] does

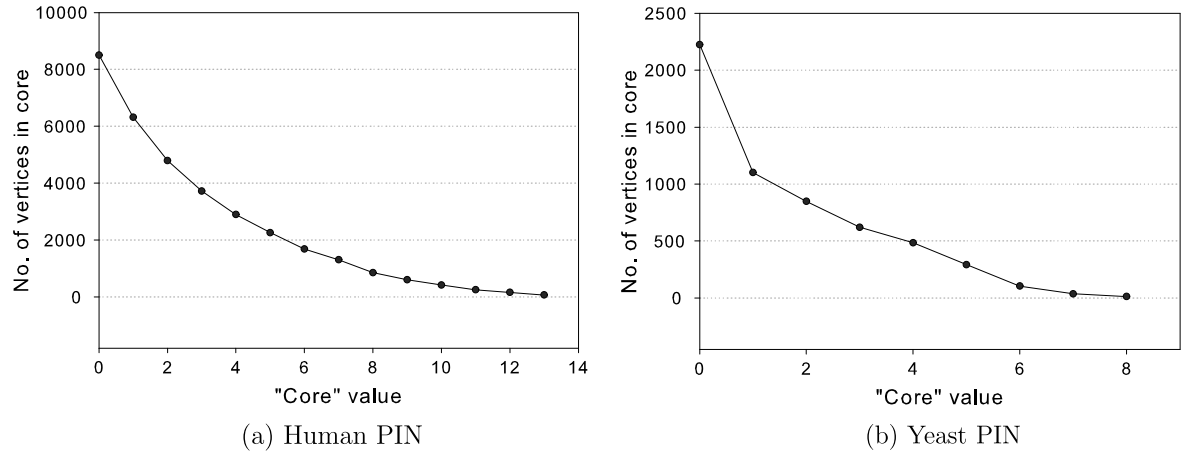

Fig. 13. Core-*k* distribution.

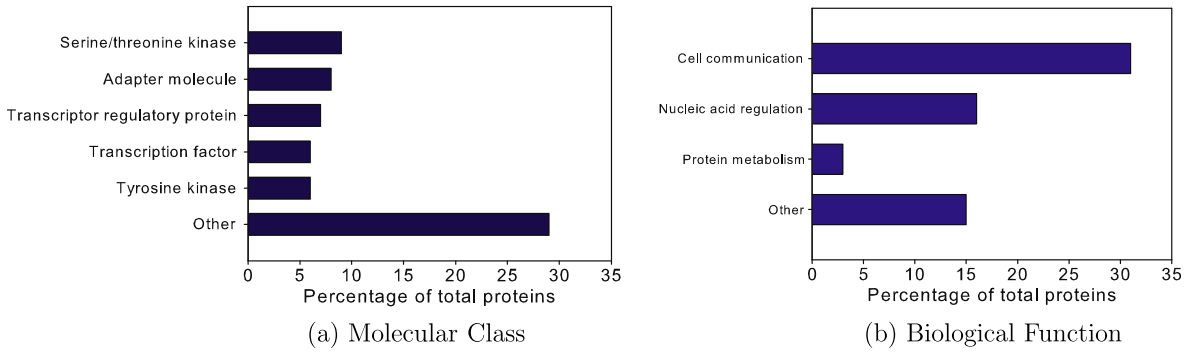

Fig. 14. The dominant molecular classes (left) and biological functions (right) among proteins that are common to both the top 1% betweenness centrality and degree lists.

not discuss algorithms for determining model parameters (the duplication and point mutation rates) to fit data given an arbitrary dataset, such as HPIN in our study. Thus, we were unable to determine whether this model could explain the centrality variance in HPIN.

Finally, we study the *coreness* of the graph using a simple heuristic. The k-core of a graph is defined as the subgraph obtained by recursively removing all vertices of degree less than k from the original graph. If a vertex belongs to the k-core but not to the $(k+1)$-core, we say that its vertex coreness is k. Vertices with degree-1 have $core = 0$. Coreness gives us an idea of how *deep* in the core a vertex is. It is related to vertex degree, but is a more sophisticated measure. A node with small coreness is not well connected and can be disconnected easily by removing its poorly connected neighbors, even if its degree is high. We see that this is exactly the case in both the human and yeast PINs (see Fig. 13). This indicates an *absence of a high-degree core* in these PINs, which implies that the network may be vulnerable to attacks on select hub proteins.

Fig. 14 is a graphical representation of the dominant molecular class and biological function among high-betweenness and high-connectivity proteins (the common proteins in the top 1% lists). These proteins belong to a variety of molecular classes (Fig. 14a), with cell communication and signal transduction being the most common biological function (Fig. 14b). The functional annotations are derived from Gene Ontology data. Due to the lack of comprehensive functional annotation datasets for the human interactome and software tools to process human interaction datasets, we could not complete functional enrichment tests on HPIN.

5. Conclusions

In this article, we demonstrate the use of multicore algorithmic techniques for large-scale protein-interaction network analysis. The source code of the various graph analysis programs is freely available online from our web site. We also intend to provide our centrality analysis codes as plug-ins to the biological network visualization tool Cytoscape [37].

Using complex network analysis techniques, we conduct an extensive study of the global topological characteristics of the human protein-interaction network. We report a new topology feature in the yeast and human PINs not found in synthetic scale-free networks: the prevalence of low-degree proteins with high-betweenness values. Our results show that existing evolutionary algorithms that produce scale-free networks do not predict the existence of high-betweenness, low-degree vertices found within the yeast and human PINs. The high-betweenness, low-centrality vertices also provide some insight into the clustering nature and coreness of the network. We find that vertices with high centrality scores are very likely to be articulation points in the graph, and also have low clustering coefficients.

Acknowledgements

This work was supported in part by NSF Grants CNS-0614915, CAREER CCF-0611589, NSF DBI-0420513, ITR EF/BIO 03-31654, and NASA Grant NP-2005-07-375-HQ.

References

[1] C. Alfarano et al, The biomolecular interaction network database and related tools: 2005 update, Nucleic Acids Res. 33 (2005) D418–D424.
[2] D.A. Bader, S. Kintali, K. Madduri, M. Mihail, Approximating betweenness centrality, in: Proceedings of the Fifth Workshop on Algorithms and Models for the Web-Graph (WAW2007), Lecture Notes in Computer Science, vol. 4863, Springer-Verlag, San Diego, CA, 2007, pp. 124–137. Decemeber.
[3] D.A. Bader, K. Madduri, Designing multithreaded algorithms for breadth-first search and *st*-connectivity on the Cray MTA-2, in: Proceedings of the 35th International Conference on Parallel Processing (ICPP), August, IEEE Computer Society, Columbus, OH, 2006.
[4] D.A. Bader, K. Madduri, Parallel algorithms for evaluating centrality indices in real-world networks, in: Proceedings of the 35th International Conference on Parallel Processing (ICPP), IEEE Computer Society, Columbus, OH, 2006.
[5] D.A. Bader, K. Madduri, A graph-theoretic analysis of the human protein interaction network using multicore parallel algorithms, in: Proceedings of the Sixth Workshop on High Performance Computational Biology (HiCOMB 2007), Long Beach, CA, March 2007.

[6] D.A. Bader, K. Madduri, SNAP, small-world network analysis and partitioning: an open-source parallel graph framework for the exploration of large-scale networks, in: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2008), Miami, FL, April 2008.
[7] N.N. Batada, T. Reguly, A. Breitkreutz, L. Boucher, B.-J. Breitkreutz, L.D. Hurst, M. Tyers, Stratus not altocumulus: a new view of the yeast protein interaction network, PLoS Biol. 4 (10) (2006) e317.
[8] V. Batagelj, A. Mrvar, Pajek – program for large network analysis, Connections 21 (2) (1998) 47–57.
[9] J. Berg, M. Lassig, A. Wagner, Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications, BMC Evol. Biol. 4 (1) (2004) 51.
[10] P. Bork, L.J. Jensen, C. von Mering, A.K. Ramani, I. Lee, E.M. Marcotte, Protein interaction networks from yeast to human, Curr. Opin. Struct. Biol. 14 (2004) 292–299.
[11] U. Brandes, A faster algorithm for betweenness centrality, J. Math. Sociol. 25 (2) (2001) 163–177.
[12] T. Coffman, S. Greenblatt, S. Marcus, Graph-based technologies for intelligence analysis, Commun. ACM 47 (3) (2004) 45–47.
[13] J.R. Crobak, J.W. Berry, K. Madduri, D.A. Bader, Advanced shortest path algorithms on a massively-multithreaded architecture, in: Proceedings of Workshop on Multithreaded Architectures and Applications, Long Beach, CA, March 2007.
[14] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1) (1977) 35–41.
[15] T.K. Gandhi et al, Analysis of the human protein interactome and comparison with yeast, worm and fly interaction datasets, Nat. Genet. 38 (2006) 285–293.
[16] L. Giot et al, A protein interaction map of drosophila melanogaster, Science 302 (2003) 1727–1736.
[17] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.
[18] R. Guimerà, S. Mossa, A. Turtschi, L.A.N. Amaral, The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles, Proc. Natl. Acad. Sci. USA 102 (22) (2005) 7794–7799.
[19] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, R. Apweiler, IntAct: an open source molecular interaction database, Nucleic Acids Res. 32 (2004) D452–D455.
[20] H. Jeong, S.P. Mason, A.-L. Barabási, Z.N. Oltvai, Lethality and centrality in protein networks, Nature 411 (2001) 41–42.
[21] H. Jeong, Z. Oltvai, A.-L. Barabási, Prediction of protein essentiality based on genomic data, ComPlexUs 1 (2003) 19–28.
[22] M.P. Joy, A. Brock, D.E. Ingber, S. Huang, High-betweenness proteins in the yeast protein interaction network, J. Biomed. Biotechnol. 2 (2005) 96–103.
[23] B. Lehner, A.G. Fraser, A first-draft human protein-interaction map, Genome Biol. 5 (9) (2004) R63.
[24] S. Li et al, A map of the interactome network of the metazoan *C. elegans*, Science 303 (5657) (2004) 540–543.
[25] F. Liljeros, C.R. Edling, L.A.N. Amaral, H.E. Stanley, Y. Åberg, The web of human sexual contacts, Nature 411 (2001) 907–908.
[26] K. Madduri, D.A. Bader, J.W. Berry, J.R. Crobak, An experimental study of a parallel shortest path algorithm for solving large-scale graph instances, in: Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX07), New Orleans, LA, January 2007.
[27] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[28] S. Peri et al, Development of human protein reference database as an initial platform for approaching systems biology in humans, Genome Res. 13 (2003) 2363–2371.
[29] J.W. Pinney, G.A. McConkey, D.R. Westhead, Decomposition of biological networks using betweenness centrality, in: Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2005), Cambridge, MA, May 2005, Poster session.
[30] A.K. Ramani, R.C. Bunescu, R.J. Mooney, E.M. Marcotte, Consolidating the set of known human protein–protein interactions in preparation for large-scale mapping of the human interactome, Genome Biol. 6 (5) (2005) R40.
[31] T. Reguly, A. Breitkreutz, L. Boucher, B.-J. Breitkreutz, G.C. Hon, C.L. Myers, A. Parsons, H. Friesen, R. Oughtred, A. Tong, C. Stark, Y. Ho, D. Botstein, B. Andrews, C. Boone, O.G. Troyanskya, T. Ideker, K. Dolinski, N.N. Batada, M. Tyers, Comprehensive curation and analysis of global interaction networks in *Saccharomyces Cerevisia*, J. Biol. 5 (2006) 11.
[32] W. Richards, International network for social network analysis, 2005. <http://www.insna.org>.
[33] W. Richards, Links to software for network analysis, 2005. <http://www.insna.org/INSNA/soft_inf.html>.
[34] J.-F. Rual et al, Towards a proteome-scale map of the human protein–protein interaction network, Nature 437 (2005) 1173–1178.
[35] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, D. Eisenberg, DIP: the database of interacting proteins: 2004 update, Nucleic Acids Res. 32 (2004) D449–D451.
[36] J.P. Scott, Social Network Analysis: A Handbook, SAGE Publications, Newbury Park, CA, 2000.
[37] P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks, Genome Res. 13 (11) (2003) 2498–2504.
[38] R.V. Sole, R. Pastor-Satorras, E. Smith, T.B. Kepler, A model for large-scale proteome evolution, Adv. Compl. Syst. 5 (1) (2002) 43–54.
[39] A.H.Y. Tong et al, Global mapping of the yeast genetic interaction network, Science 303 (5659) (2004) 808–813.
[40] P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, J.M. Rothberg, A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*, Nature 403 (2000) 623–627.
[41] A. Vazquez, A. Flammini, A. Maritan, A. Vespignani, Global protein function prediction in protein–protein interaction networks, Nat. Biotechnol. 21 (6) (2003) 697–700.
[42] A. Vazquez, A. Flammini, A. Maritan, A. Vespignani, Modeling of protein interaction networks, ComPlexUs 1 (2003) 38–44.
[43] S. Wuchty, E. Almaas, Peeling the yeast protein network, Proteomics 5 (2) (2005) 444–449.