Analyzing Protein Lists with Large Networks: Edge-Count Probabilities in Random Graphs with Given Expected Degrees

Joël R. Pradines, Victor Farutin, Steve Rowley, Vlado Dancik

Publisher's Note / Disclaimer:

This is an accepted manuscript (post-print) of an article published in:
Journal of Computational Biology, Volume 12, Number 2, 2005, pp. 113-128.

The final authenticated version is available online at:
https://doi.org/10.1089/cmb.2005.12.113

Published by Mary Ann Liebert, Inc.

To appear in the Journal of Computational Biology
(Volume 12, Number 2, pp 113-128)
(submitted 12/01/03, accepted 07/09/04)

Analyzing Protein Lists with Large Networks: Edge-count probabilities in random graphs with given expected degrees

Running head: ANALYZING PROTEIN LISTS WITH LARGE NETWORKS
Key words: protein network, sparse graph, random graph, degree sequence problem

JOËL R. PRADINES,[1*] VICTOR FARUTIN,[1] STEVE ROWLEY[1,2] and VLADO DANČÍK[1]

[1] Computational Biology, Informatics, Millennium Pharmaceuticals Inc., 40 Landsdowne Street, Cambridge, MA 02139, USA.
*corresponding author: Phone: (617) 551-8667, Email: joel.pradines@mpi.com, Fax: (617) 577-3555
V. Farutin: Phone: (617) 679-7042, Email: victor.farutin@mpi.com
V. Dančík: Phone: (617) 761-6967, Email: vlado.dancik@mpi.com

[2] Current address: Aventis Pharmaceuticals, 26 Landsdowne Street, Cambridge, MA 02139, USA.
S. Rowley: Phone: (617) 768-4054, Email: steve.rowley@aventis.com

ABSTRACT

We present an analytical framework to analyze lists of proteins with large undirected graphs representing their known functional relationships. We consider edge-count variables such as the number of interactions between a protein and a list, the size of a subgraph induced by a list and the number of interactions bridging two lists. We derive approximate analytical expressions for the probability distributions of these variables in a model of random graph with given expected degrees. Probabilities obtained with the analytical expressions are used to mine a protein interaction network for functional modules, characterize the connectedness of protein functional categories and measure the strength of relations between modules.

1 Introduction

Cellular functions are carried out via thousands of biochemical interactions involving macromolecules and small molecules. To correctly describe, model, simulate and ultimately manipulate (Bailey, 1999) the functioning of such complex systems it is important to first understand their topology, namely the structure of the molecular networks on which the interactions take place. Over the last five years there have been considerable advances in the science that studies the structure of large networks (Albert and Barabási, 2002; Newman, 2003b). Powerful mathematical tools used in these studies are models of random graphs with fixed degree sequence or fixed degree distribution (Itzkovitz *et al.*, 2003; Molloy and Reed, 1995; Newman *et al.*, 2001). These models have been used to uncover fundamental features of real networks such as mixing patterns (Maslov and Sneppen, 2002; Newman, 2003a) and network motifs (Milo *et al.*, 2002; Shen-orr *et al.*, 2002). These features were identified as those unlikely to be observed in randomized versions of the network preserving the number of neighbors of each vertex.

Here, we use a similar approach to analyze lists of proteins with large networks representing their known interactions. For instance, let us assume that gene lists were obtained as a result of a microarray experiment. We would like to identify those lists containing significant portions of known 'pathways' or functional modules. Given a graph G of interactions between the gene products, we can score each list with the size of the subgraph it induces in G. A large number of edges indicates that the list members are closely functionally related. The notion of 'large number' should be conditional to the size of the list. We can also make it conditional to the degrees of the list members in the graph, i.e. conditional to how much knowledge the graph contains for each protein. This can be done by using the likelihood P for the size of the subgraph induced by the prescribed list in a model of random graph with the degree sequence of G. By likelihood we mean here the probability of observing an equal or greater number of edges.

In this paper we consider the following problem: given a graph G and prescribed vertex lists we define edge sets as functions of the lists and want to compute the likelihoods of their sizes in a random graph model. Since the analytical treatment of random graphs with fixed degree sequence is hard (Bender and Canfield, 1978; Itzkovitz *et al.*, 2003; Molloy and Reed, 1995), we consider an alternate model where vertices have given expected degrees (Chung and Lu, 2002). We present a general approach to derive analytical expressions for the likelihoods of edge-count variables in this model. These expressions are approximations valid for sparse graphs and make use of the stability of the Poisson distribution. We also present examples of applications of the derived likelihoods with a yeast protein interaction network and lists corresponding to functional categories of yeast proteins. The interaction network is the 'spoke' version of the BIND database [1] (Bader *et al.*, 2003). The functional categories are based on the Gene Ontology (GO) annotation provided in the RefSeq database [2] (Pruitt and Maglott, 2003).

The paper is organized as follows. In section 2 we introduce the notations, briefly describe the models of random graphs, derive the probability of observing a given edge and approximate the probability distribution of edge-count variables for random graphs with given expected degrees. Section 3 examines the likelihood P_a for the number of edges between a vertex and a list. The divergence of our approximations from frequencies observed in random graphs with fixed degree sequence is studied. P_a is then used to extract subgraphs of high connectedness from the protein interaction network. Section 4 deals with the size of a subgraph induced by a vertex list and its associated likelihood P_L. We present an algorithm to compute the maximal size of the subgraph from the degree sequence of the list members. We then use P_L to characterize the connectedness of GO categories. In section 5 we consider the size of a bipartite subgraph induced by two vertex lists. We show that the associated likelihood P_b is an interesting measure to estimate the strength of relations between lists and to group them. In section 6 we summarize the results, discuss potential applications of our approach and its limitations.

[1] release of September 2003

[2] release of June 2003

2 Random graph models, edge probability and Poisson approximation

We first introduce the notations used throughout the paper and briefly describe two models of random graphs. We then derive the probability of observing an edge given the degrees of its end vertices, and introduce an approximation for the probability distribution of edge-count variables in random graphs with given expected degrees.

2.1 Notations

We call $G = (V, E)$ the undirected graph of interest. V is the vertex set (proteins) and E the edge set (interactions). $N = |V|$ is the order of G and $M = |E|$ its size, where $|S|$ stands for the cardinality of a set S. An edge between vertices v_i and v_j is referred to as v_iv_j. k_i is the degree of vertex v_i, i.e. the number of its incident edges. We assume that G has no isolated vertex ($\forall i, k_i \geq 1$), no multiple edge and no loop (v_iv_i). A list L is any subset of V. We call $n = |L|$ the "size" of L. A graph G is factorable into $G_1 \oplus G_2$ if $V = V_1 \cup V_2$, $E = E_1 \cup E_2$ and $E_1 \cap E_2 = \emptyset$. The subgraph of G induced by L has vertex set L and its edges have end vertices in L. The bipartite subgraph of G induced by L_1 and L_2, with $L_1 \cap L_2 = \emptyset$, has all its edges with one end vertex in one list and the other end vertex in the other list. We use the two following criteria to say that G is sparse

$$\max_{v_i \in V} (k_i) \ll M \tag{1}$$

$$\max_{(v_i \in V, v_j \in V)} (k_i k_j) \ll M \tag{2}$$

In the context of a random graph, following an analogy similar to that introduced in (Newman *et al.*, 2001), we refer to vertex v_i as a "sea urchin" having k_i "spikes". These spikes can be seen as the result of breaking in half all edges of G. We will use uppercase symbols to denote random variables (X) and lowercase symbols for their realizations (x). If a variable is defined with respect to a random graph of fixed degree sequence, instead of given expected degrees, it is marked with a prime (X').

2.2 Random graph models

Given a graph G we consider the set Ω of all undirected graphs of same order N and same degree sequence. A realization of a random graph with the degree sequence of G is obtained by uniformly choosing one element of Ω (Milo *et al.*, 2003). An approximation for the size of Ω can be found in Bender and Canfield (1978) . If N is large, instead of working with Ω one can consider the family of infinite order graphs having the degree distribution of G. Analytical expressions for the expected values of many properties (e.g. average clustering coefficient) of such random graph models have been obtained (Newman *et al.*, 2001). To simulate random graphs with fixed degree sequence we use an edge rewiring algorithm (Maslov and Sneppen, 2002; Maslov *et al.*, 2003), also known as switching algorithm. Starting from the initial graph G, pairs of edges (uv, wx) are randomly chosen and changed with equal probabilities to (ux, wv) or (uw, vx). The rewiring is performed only if it does not create an edge already present in the network or a loop. This rewiring is performed QM times, with $Q = 100$ leading to a uniform sampling of Ω (Milo *et al.*, 2003).

The computation of probabilities associated with random graphs of fixed degree sequence is a difficult problem (Itzkovitz *et al.*, 2003). Here we use an alternate random graph model that makes analytical treatment much easier. In this model an edge v_iv_j is created with probability $\min(1, k_ik_j/(2M))$ (Chung and Lu, 2002). In other words, edges are treated as independent random variables even if they have common end vertices. As we will see in the next section for sparse networks, the degree of a vertex v_i is now a random variable K_i having a Poisson probability distribution of parameter k_i. Since the coefficient of variation of K_i is $1/\sqrt{k_i}$, such random graph with prescribed expected degrees is a reasonable null model of the real network G.

2.3 Probability p_{ij} of an edge

We first focus on random graphs with fixed degree sequence. We call p_{ij} the probability of observing an edge between two vertices v_i and v_j. An expression of p_{ij} as a function of k_i and k_j has been recently published (Itzkovitz *et al.*, 2003). Here, we follow a different path to derive the same expression, stressing on the way the passage to a Poisson distribution.

To approximate p_{ij} we follow the switching algorithm explained above, allowing temporarily for multiple edges. The switching is performed by pairing edges. The proportion of edges containing v_i is k_i/M and that of edges involving v_j is k_j/M. Consequently, the probability of an edge pair containing v_i in one edge and v_j in the other is $2k_ik_j/M^2$. This pair has probability $1/2$ of generating v_iv_j upon rewiring. Assuming that all edges have been paired at least once, we can then approximate the number of v_iv_j edges generated by a random variable X'_{ij} having binomial distribution of parameters $\lfloor M/2 \rfloor$ and k_ik_j/M^2. For a sparse graph (1) $k_ik_j/M^2 \ll 1$, and we can further approximate to a Poisson distribution of parameter

$$\lambda_{ij} = \frac{k_ik_j}{2M} \tag{3}$$

Therefore, the probability of observing at least one v_iv_j edge is approximately

$$\frac{e^{-\lambda_{ij}}}{\alpha_{ij}} \sum_{x=1}^{k_m} \frac{\lambda_{ij}^x}{x!} = 1 - \frac{e^{-\lambda_{ij}}}{\alpha_{ij}} \tag{4}$$

where the normalization factor α_{ij} is the sum of Poisson terms from 0 to $k_m = \min(k_i, k_j)$, the maximal number of v_iv_j edges. Now, if the initial graph G satisfies (2), for all pairs of vertices we have $\lambda_{ij} \ll 1$ and consequently $\Pr\left(X'_{ij} = x+1\right) \ll \Pr\left(X'_{ij} = x\right)$. This means that the normalization coefficient α_{ij} is very close to 1. One can therefore approximate the probability of observing v_iv_j with

$$p_{ij} \simeq 1 - \exp\left(-\frac{k_ik_j}{2M}\right) \tag{5}$$

which is the expression obtained by Itzkovitz *et al.* (2003). Using (2) we can further approximate to the probability of creating v_iv_j in the random graph with given expected degrees (Chung and Lu, 2002)

$$p_{ij} \simeq \beta = \frac{k_ik_j}{2M} \tag{6}$$

To derive (5) we have neglected the fact that multiple edges are forbidden. Therefore, we numerically checked that (5) provided reasonable approximations for random graphs with no multiple edges and having the degree sequence of the protein interaction network used in this paper. We first counted the number n_β of pairs (v_i, v_j) of vertices such that $k_ik_j/2M = \beta$. We then simulated $n_s = 10^5$ random graphs with the same degree sequence following the switching procedure explained above. We estimated the true probability $p_t(\beta)$ of observing an edge such that $k_ik_j/2M = \beta$ with $\tilde{p}_t(\beta) = n_{obs}(\beta) / (n_\beta n_s)$, where n_{obs} is the number of edges obtained with the simulations. Since the standard deviation of $\tilde{p}_t$ is $1/\sqrt{n_\beta n_s}$, the largest errors of estimation are obtained for $n_\beta = 1$ and are of the order of 0.003. Figure 1 shows $\tilde{p}_t$, (5) and (6) as a function of β. One can see that the approximation is quite good. The relative error is always small even for edges that clearly violate the sparseness condition (2) with $\beta > 1$. Moreover, the proportion of such edges is small, as less than 0.1% of all possible pairs of vertices are such that $\beta > 0.1$.

While random graphs with fixed degrees or given expected degrees lead to the same expression of p_{ij} for sparse networks, we will see in section 3 that the probability distributions of edge-count variables can differ very significantly between the two models.

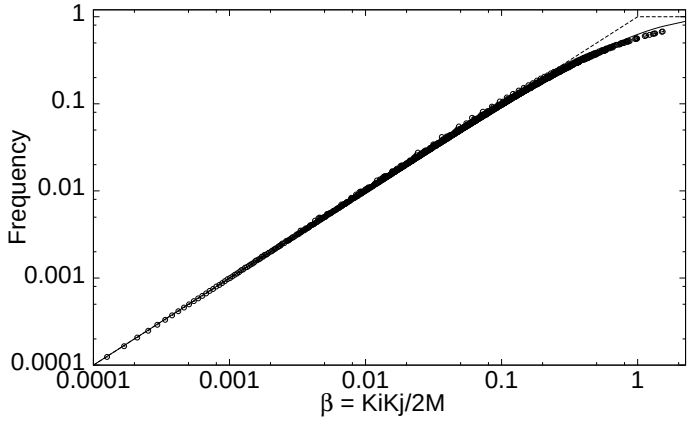

FIG. 1. Probability of observing an edge as a function of the normalized degree product β of the end vertices. The plain line corresponds to equation (5) and the dotted one to equation (6). The dots are the values estimated with 10^5 graphs having the degree sequence of the protein interaction network.

2.4 Poisson approximation for edge count probabilities

Given a graph G and lists of vertices we define edge sets as functions of the lists and are interested in comparing their cardinality. For instance, we want to compare the connectedness of GO categories. The comparison should take into account the degrees of the vertices in a list. To do so we use the set of random graphs with given expected degrees as a reference point, compute the likelihood of each edge set and compare the edge sets by their likelihoods. By likelihood we mean here $P(z) = \Pr(Z \geq z)$, where z is the size of an observed edge set and Z is the corresponding random variable in the random graph. We now present a simple and general approach to compute an approximation of $P(z)$ for sparse graphs. Z is the sum of independent Bernoulli random variables, each representing one edge:

$$Z = \sum_{h=1}^{m} B_h \tag{7}$$

The parameter p_h of B_h is the probability of observing a given edge (5). m is the maximal number of edges that could be observed. The probability generating function of Z is

$$U_Z(t) = \prod_{h=1}^{m} (1 - p_h(1-t)) \tag{8}$$

By definition $p_h < 1$ and $p_h \ll 1$ for a sparse graph (2). Consequently, U_Z can be approximated with the generating function of a Poisson probability distribution (Feller, 1970)

$$\ln(U_Z(t)) = \sum_{h=1}^{m} \ln(1 - p_h(1-t)) \simeq -\lambda(1-t) \tag{9}$$

with

$$\lambda = \sum_{h=1}^{m} p_h \tag{10}$$

The passage to a Poisson distribution is not dependent upon m being large compared to 1. Such dependence is required only in an asymptotic model where $M \to \infty$ and $p_h \to 0$. Here, we always work with a graph of finite size. Now, if the edge set consists of all the connections a vertex v_i could have, equation (9) shows that the degree K_i of v_i in the random graph with given expected degrees has a Poisson distribution of parameter close to k_i.

Since the values of Z are bounded by m, we normalize as follows

$$P(z) = \frac{e^{-\lambda}}{\alpha} \sum_{h=z}^{m} \frac{\lambda^h}{h!}; \qquad \alpha = e^{-\lambda} \sum_{h=0}^{m} \frac{\lambda^h}{h!} \tag{11}$$

In many cases the sparseness of G implies that $m \gg \lambda$ and $\alpha \simeq 1$. It also means that $P(z) \simeq e^{-\lambda}\lambda^z/z!$, when $z > \lambda$. However, knowing m is still required for many numerical evaluations of (11). The computation of m is in general trivial for the random graph with given expected degrees. For random graphs with fixed degrees, the maximal size z_m of an edge set defined by a list of vertices can be a complicated function of their degrees. We will provide means of computing such maximal sizes. This will allow us to compare the usage of $P(z)$ or z/z_m to analyze protein lists.

We next present results for three types of edge sets of practical utility.

3 Attachment of a vertex to a list

Let L be a list of size n with $n < N$, and v_i a vertex that does not belong to L. We call x_{iL} the attachment of the vertex to the list, i.e. the number of edges between v_i and the elements of L. We want to approximate the following probability

$$P_a(x) = \Pr(X_{iL} \geq x) \tag{12}$$

where X_{iL} is the random variable corresponding to x_{iL} in the random graph with given expected degrees. We apply the model Z described above and use (11) with

$$\lambda_{iL} = \sum_{v_j \in L} p_{ij}; \quad \alpha_{iL} = e^{-\lambda_{iL}} \sum_{u=0}^{n} \frac{\lambda_{iL}^u}{u!} \tag{13}$$

Figure 2A compares the predicted values of P_a to those numerically estimated with 10^5 random graphs with given expected degrees. These networks were generated by independently creating each edge v_iv_j with probability p_{ij} (5). The parameters chosen for the attachment are $k_i = 5$ and $n = 50$. 1,000 random lists were generated and all vertices of degree 5 were used to estimate all possible values of P_a. One can see that the predicted values are in good agreement with the numerical results. In contrast, the values of P_a differ significantly from the frequencies obtained by simulating random graphs with fixed degrees (Figure 2B). The divergence increases with the value of X_a. This discrepancy is caused by the fact that edges sharing an end vertex are not independent variables in a random graph of fixed degree sequence. For the attachment of a vertex to a list it is possible to take such dependence into account by decreasing k_i as the edges of v_i are created (Newman *et al.*, 2001; Itzkovitz *et al.*, 2003). Namely, one graph configuration realizing an attachment x has approximate probability $k_ik_1(k_i-1)k_2(k_i-2)k_3\ldots(k_i-x+1)k_x/(2M)^x$. $\Pr(X'_a = x)$ is the sum of all configuration probabilities. Summing over subsets can be computationally expensive, so we simplify to $k_j = \langle K \rangle_L$ for all $v_j \in L$. After summing over the configurations, cumulating $\Pr(X'_a = x)$ gives

$$P'_a(x) = \sum_{y=x}^{x_m} \binom{n}{y} \left(\frac{\langle K \rangle_L}{2M}\right)^y \prod_{z=0}^{y-1} (k_i - y); \quad (x \geq 1) \tag{14}$$

with $x_m = \min(k_i, n)$. Figure 2B shows that (14) provides excellent approximations for random graphs with fixed degrees. The divergence between the two random graphs models can be quantified as follows

$$\ln\left(\Pr(X_a = x) / \Pr(X'_a = x)\right) = -\frac{nk_i \langle K \rangle_L}{2M} + \sum_{y=0}^{x-1} \ln\left(\frac{nk_i}{(n-y)(k_i-y)}\right)$$

The ratio can reach several orders of magnitude for large values of x. In this paper we use the likelihood P of an edge-count variable to sort vertex lists, e.g. to answer questions such as "what is the best connected GO category to my protein of interest?". A relevant question is thus whether sorting edge sets based on P_a or P'_a is equivalent. This was tested with the following experiment. Lists of size $n = 5, 10, \ldots, 50$ were

generated. For each size nine lists were randomly created, imposing minimal degrees of values $1, 2, \ldots, 10$. Then, for nine vertices of degrees $2, 3, \ldots, 10$, the values of P_a and $P'a$ were computed for all possible attachments to the lists, at the exception of $x = 0$. This gave a total of 5,250 points. The points were then sorted based on P_a or P'_a and given ranks between 0 and 1. Figure 2C shows that, sorting with P_a or P'_a is not equivalent. There is however a strong correlation between the two vectors of ranks ($c = 0.9989$).

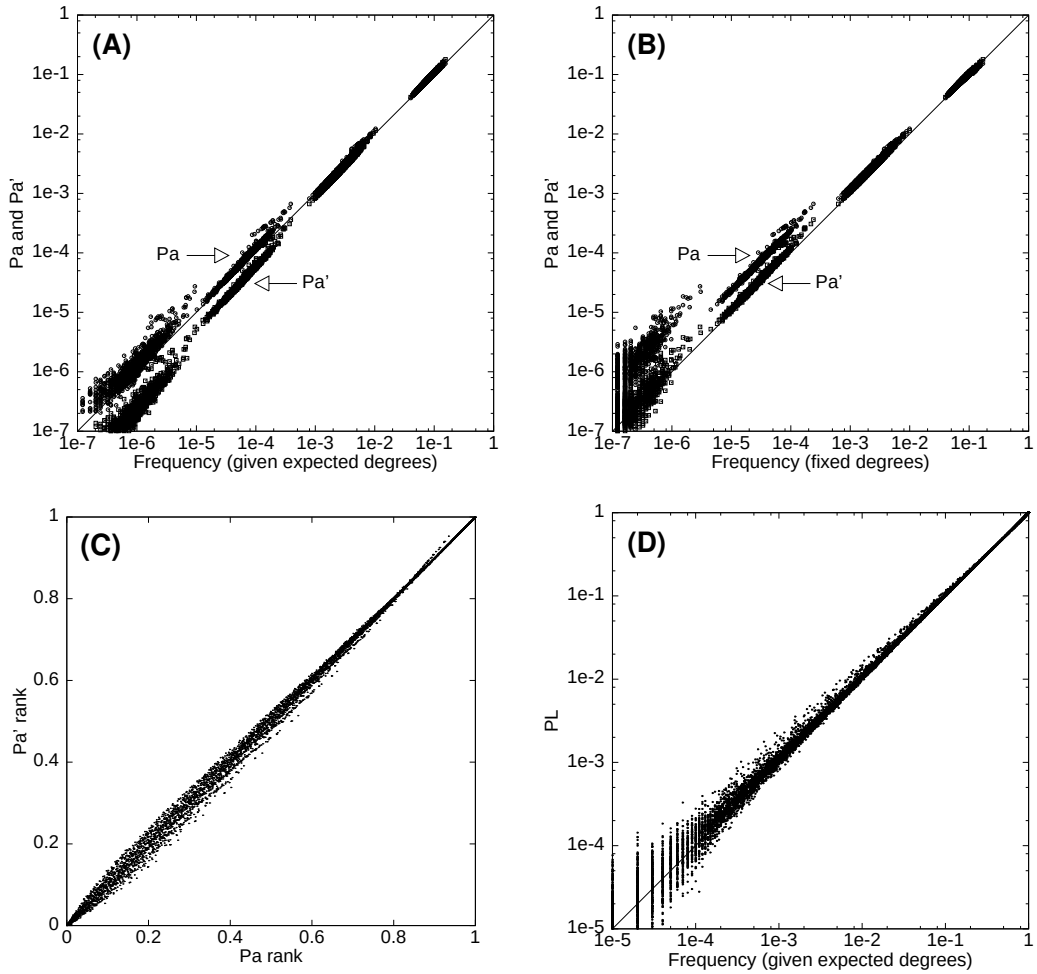

FIG. 2. **(A)** Comparison of the values of P_a and P'_a ($k_i = 5, n = 50$, 1,000 lists) to the frequencies estimated with 10^5 random graphs with given expected degrees. **(B)** Same as (A), but for 10^5 random graphs with fixed degree sequence. **(C)** Sorting edge-sets based on their values of P_a or P'_a is strongly correlated, $c = 0.999$ with 5,250 points. **(D)** Comparison of the values of P_L (1,000 lists of size $n = 50$) to the values estimated with 10^5 random graphs with prescribed expected degrees.

Next, we use P_a with a simple greedy algorithm to extract from the network sets of vertices that, given their degrees, induce subgraphs of high connectedness. There have been several recent publications presenting methods to extract such structures from large networks (Bader and Hogue, 2002; Spirin and Mirny, 2003; Krause *et al.*, 2003; Samanta and Liang, 2003), some with the broader goal of organizing the structures into a hierarchy (Rives and Galitski, 2003; Newman, 2004; Girvan and Newman, 2002; Ravasz *et al.*, 2002). P_a is an interesting measure to perform such mining as it is conditional to the degrees of all considered vertices and does not require Monte Carlo simulations. The algorithm is seeded with a list consisting of the two end vertices of an edge. The vertex with the smallest P_a value for this list is then added. The list is grown in this way as long as the P_a value of the added vertex decreases.

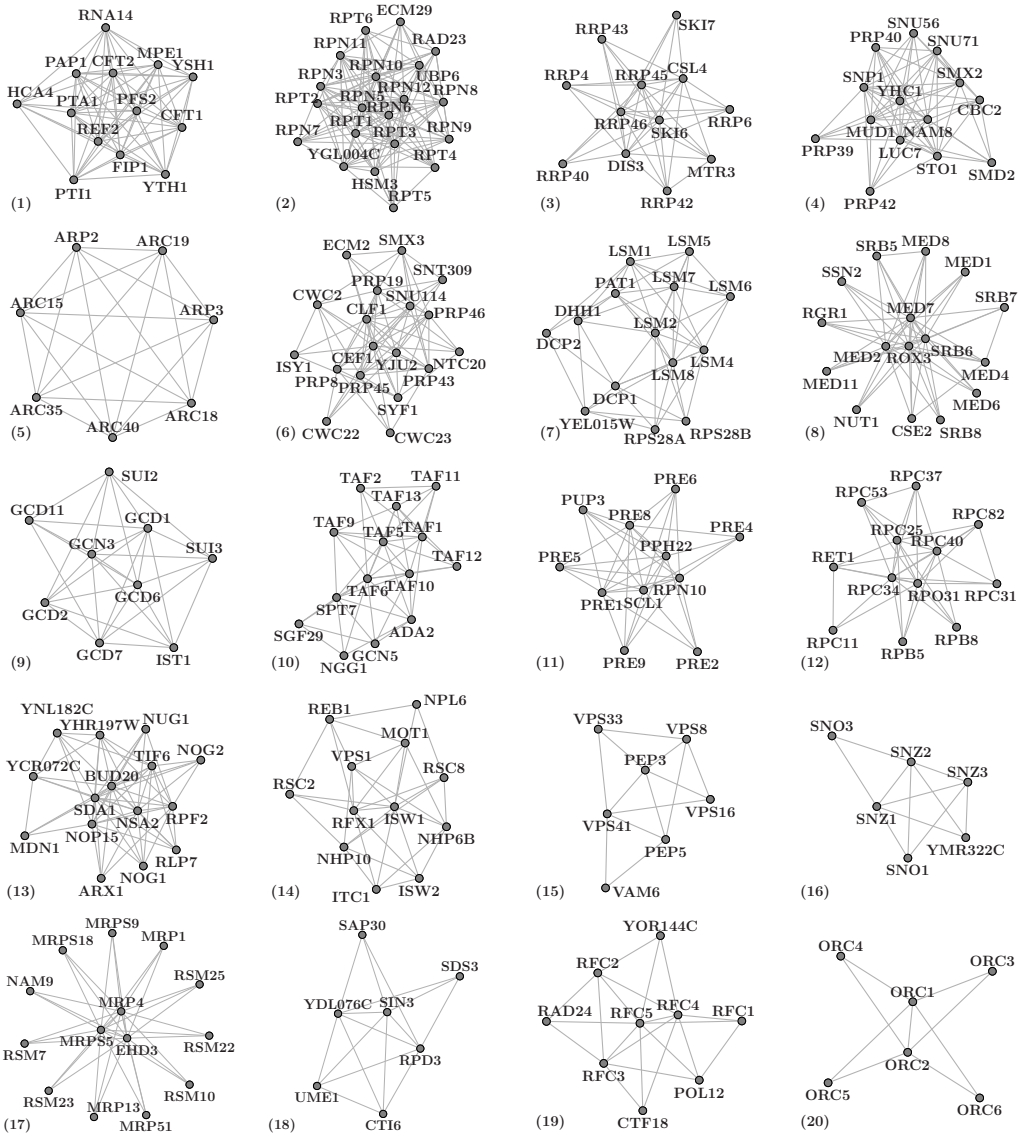

FIG. 3. Twenty structures extracted from the protein interaction network with a greedy algorithm based on P_a. The final values of P_a range from 10^{-17} to 10^{-9}. (1) Cleavage and polyadenylation factor. (2) 19S proteasome regulatory particle. (3) Exosome. (4) Small nuclear ribonucleoproteins U1. (5) Arp2/3 complex. (6) Spliceosome proteins. (7) LSM complex. (8) Mediator complex. (9) Translation initiation factors EIF2 and EIF2B. (10) TAF-II complex. (11) 20S proteasome core complex. (12) RNA polymerase III. (13) Large ribosomal subunit assembly and maintenance. (14) Chromatin remodeling. (15) Vacuolar proteins. (16) Pyridoxine synthesis. (17) Mitoribosome small subunit. (18) Histone deacetylase complex. (19) Replication factor C. (20) Origin recognition complex.

We applied this algorithm to the protein interaction network, using each edge as a seed. If two obtained modules overlapped for at least half of the members of one of them, the module with the largest final P_a value was discarded. The subgraphs induced by the twenty lists of smallest final P_a values are displayed in Figure 3. These represent parts or whole of known protein complexes, or groups of functionally related enzymes such as list (16), a cluster of putative metabolic enzymes that might be involved in the synthesis of vitamin B6 (Braun *et al.*, 1996). At the exception of (5) and (20) the lists do not induce maximally

connected subgraphs. Lists (2), (6), (7), (9), (13), (17), (18) and (19) contain proteins (YGL004C, YJU2, YEL015W, IST1, YNL182C/YHR197W, EHD3, YDL076C and ELG1) that have been previously assigned putative functions based on their interactions (Samanta and Liang, 2003). The functions that were predicted for these proteins are consistent with the lists we extracted here. In addition, list (6) indicates that CWC23 might be involved in spliceosomal function (Sanders *et al.*, 2002). Also, list (18) contains the protein YCR072C, suggesting its potential involvement in ribosomal biogenesis (Bassler *et al.*, 2001). Our results suggest that it could be interesting to use P_a as a proximity measure of proteins to lists of annotated genes in an algorithm specifically designed to transfer annotation (Letovsky and Kasif, 2003).

4 Size of a subgraph induced by a list

Given two lists L and L' we would like to compare the sizes s_L and s'_L of the subgraphs they induce in G. In other words, given the degrees of the vertices in L and L' we wish to state about which list is more connected in G. Again, we use the random graph with given expected degrees to compare s_L and $s_{L'}$. We call S_L the random variable corresponding to s_L in the random graph. We want to find an approximation of

$$P_L(s) = \Pr(S_L \geq s) \tag{15}$$

As in the previous section, we model the edges between vertices of L with Bernoulli random variables, approximate the distribution of their sum with a Poisson law (11) and normalize by taking into account the maximal value of S_L

$$\lambda_L = \sum_{i<j} p_{ij}, \quad (v_i, v_j \in L); \quad \alpha_L = e^{-\lambda_L} \sum_{u=0}^{n(n-1)/2} \frac{\lambda_L^u}{u!} \tag{16}$$

Figure 2D compares values obtained with (11) and (16) to the frequencies estimated with 10^5 simulations of random graphs with given expected degrees. 1,000 lists of sizes ranging from 5 to 50 were used to generate the figure. The agreement between predicted and estimated values is quite good.

To illustrate the advantage of using the random graph with given expected degrees as a null model, we will compare the results obtained when scoring lists with P_L or with their relative connectedness in G. By relative connectedness we mean s_L/s_{Lm}, where s_{Lm} is the maximal size of the subgraph induced by L across the set of all graphs with the degree sequence of G. s_{Lm} is a function of the degrees of the elements of L in G. Let $K = (k_1, \ldots, k_n)$ be the nonincreasing sequence of these degrees. There are two obvious bounds for s_{Lm}

$$\min(k_1, n-1) \leq s_{Lm} \leq \left\lfloor \frac{1}{2} \sum_i \min(k_i, n-1) \right\rfloor \tag{17}$$

The lower bound is due to the absence of isolated vertices in the initial graph G. The upper bound takes into account the sparseness of the graph. Nevertheless, it is possible to compute the exact value of s_{Lm}.

For any graph with the degree sequence of G, the elements of L induce a subgraph of degree sequence $D = (d_1, \ldots, d_n)$. D has the following properties

$$0 \leq d_i \leq \min(k_i, n-1) \qquad (i = 1, \ldots, n) \tag{18}$$

$$\text{The integer sequence } D \text{ is graphical} \tag{19}$$

and the corresponding size is

$$s(D) = \frac{1}{2} \sum_{i=1}^{n} d_i \tag{20}$$

There are at least eight equivalent criteria to test for property (19) (Sierksma and Hoogeveen, 1991). Here, we use a theorem due to Havel (Havel, 1955) and Hakimi (Hakimi, 1962)

Theorem 1 (Havel-Hakimi) *A positive integer sequence $D = (d_1, \ldots, d_n)$ with $d_1 \geq \ldots \geq d_n$, $d_1 \geq 1$ and $n \geq 2$ is graphical if and only if the sequence $D^* = (d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$ is graphical.*

Recursive application of this theorem gives an algorithm to test whether an integer sequence is graphical or not (Hakimi, 1962). We adapt this algorithm to compute s_{Lm} from K.

Algorithm A

(A_0) $w \leftarrow 0$; $R \leftarrow K$.

(A_1) Order by decreasing values the elements of R.

(A_2) Suppress from R any element less than 1. Call m the number of remaining elements.

(A_3) For all elements of R, $r_i \leftarrow \min(r_i, m-1)$

(A_4) $w \leftarrow w + r_1$

(A_5) Create the sequence $R^* = (r_2 - 1, \ldots, r_{r_1+1} - 1, r_{r_1+2}, \ldots, r_m)$.

(A_6) If R^* contains less than two integers greater than 0, return w. Otherwise, $R \leftarrow R^*$ and go to (A_1).

Theorem 2 *Algorithm A returns s_{Lm}.*

Proof. We can think of algorithm A as the construction of a graph factorable into $H = St_1 \oplus St_2 \oplus \ldots \oplus St_l$ $(l \leq n-1)$. St_1 is the star of largest size $(\min(k_1, n-1))$ that can be built from K. St_2 is the star of largest size after removing St_1, etc. We now show that there is a graph of size s_{Lm} satisfying (18) and having the same factorization. Since the factors are edge disjoint this implies that algorithm A computes s_{Lm}.

Let $F = (f_1, \ldots, f_n)$ be a nonincreasing graphical sequence such that $s(F) = s_{Lm}$. We have $f_i \leq \min(k_i, n-1)$, for all elements of F. Let us assume that $f_1 < \min(k_1, n-1)$. This means that for a graph $g(F)$ realizing F the vertex v_1 has $\min(k_1, n-1) - f_1$ spikes left. There cannot be a vertex v_j not connected to v_1 and having one or more free spikes. If there were, an additional edge could be created, leading to a graph of size $s_{Lm}+1$. However, there must be in $g(F)$ at least $\min(k_1, n-1) - f_1$ edges whose vertices are not adjacent to v_1. Indeed, the absence of such edges combined to $f_1 < \min(k_1, n-1)$ would contradict the lower bound of s_{Lm} (17). We break some of these edges and rewire the vertices to v_1 until its degree is maximal. The rewiring does not change the total number of edges. It gives a new graphical sequence F' such that $f'_1 = \min(k_1, n-1)$ and $s(F') = s(F) = s_{Lm}$. If we had $f_1 = \min(k_1, n-1)$, then $F' = F$.

Since F' is graphical, we can apply Theorem 1 and obtain a graphical sequence F'^* such that $s(F') = s\left(F'^*\right) + \min(k_1, n-1)$. We have created the first factor St_1. Removing the edges of St_1 gives a remaining sequence K^* of spikes. Because $s_{Lm} = s\left(F'^*\right) + \min(k_1, n-1)$, F'^* is a graphical sequence of maximal size when generating a graph from K^*. Applying steps (A_2) and (A_3) of algorithm A to K^* does not change the value of such maximal size. Consequently, we can reiterate the reasoning performed for (F, K) with $\left(F'^*, K^*\right)$. This proves that a graph of size s_{Lm} can be rewired to have the factorization constructed by algorithm A. □

We now use P_L and s_{Lm} to characterize GO categories with the protein interaction network. The distribution of P_L values obtained for all GO categories (1,212 categories having at least two proteins in the network) is strongly bimodal: 43% of lists with $P_L < 0.05$ and 54% with $P_L = 1$. For comparison, the average distribution of P_L over 1,000 rewired network gives 1% of lists with $P_L < 0.05$ and 90% with $P_L = 1$. Therefore, because of the graph sparseness, we cannot make any statistically meaningful statement about the absence of edges in a category. In contrast, small values of P_L indicate lists whose members are significantly related in the network. A small value of P_L does not necessarily correspond to a list inducing a subgraph of large relative connectedness s_L/s_{Lm}. This is shown in Figure 4. In the upper part of the figure GO categories are represented by dots. The first coordinate is the rank of $(1 - P_L)$ normalized between 0 and 1, and the second coordinate is s_L/s_{Lm}. The vertical line indicates the smallest P_L value (10^{-6}) obtained for the same lists with 1,000 rewired networks. One can see that lists with significantly small values of P_L can have very different relative connectedness. Two categories with significantly small P_L values are marked in Figure 4A. The subgraphs they induce in the network are displayed underneath. Gray nodes are list members and white nodes their adjacent proteins in the network. The "Arp2/3 protein complex" induces a clique. The "Transcription elongation factor complex" induces a subgraph that is

not maximally connected with $s_L/s_{Lm} \simeq 0.3$. However, given the size of the list and the degrees of its members, the induced subgraph is significantly connected.

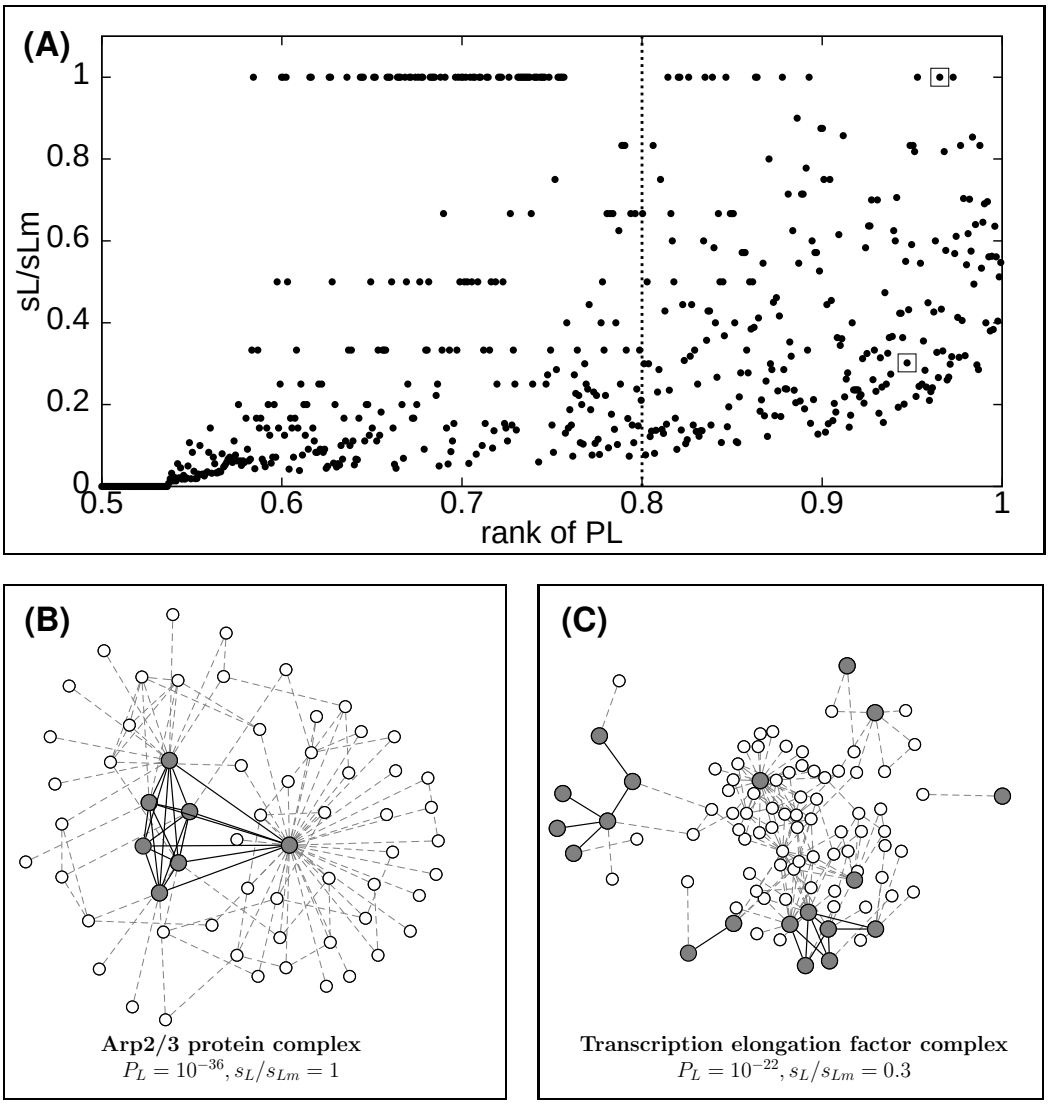

FIG. 4. (**A**) Comparison of the relative connectedness S_L/s_{Lm} and P_L for GO categories. The vertical dotted line corresponds to $P_L = 10^{-6}$, the smallest value obtained with 1,000 rewired networks (fixed degree sequence). Two GO categories with significantly small values of P_L are marked: "Arp2/3 protein complex" and "Transcription elongation factor protein complex". (**B**) and (**C**) Subgraphs induced by the two selected lists. Gray nodes are list members and white nodes their adjacent proteins in the network. Solid lines are edges between list members and dashed lines edges involving at least one protein not in the list.

In conclusion, P_L allows one to identify lists of vertices that induce significantly connected subgraphs, without requiring Monte Carlo simulations. A potential application of P_L is the post-processing of gene sets obtained from microarray data analysis. If one has prior knowledge about functional relationships between proteins encoded in the form of a graph, then P_L can be used to detect lists of significantly related genes.

5 Size of a bipartite subgraph induced by two lists

Let L_1 and L_2 be two lists of respective sizes n_1 and n_2 and such that $L_1 \cap L_2 = \emptyset$. We call s_b the number of edges bridging L_1 and L_2 in G and S_b the corresponding random variable in the random graph with given expected degrees. To compare the sizes of bipartite subgraphs induced by pairs of lists we can use

$$P_b(s) = \Pr(S_b \geq s) \tag{21}$$

Again, we approximate the distribution of S_b with a truncated Poisson law (11) of parameters

$$\lambda_b = \sum_{i,j} p_{ij}, \quad (v_i \in L_1, v_j \in L_2); \quad \alpha_b = e^{-\lambda_b} \sum_{u=0}^{n_1 n_2} \frac{\lambda_b^u}{u!} \tag{22}$$

In order to compare the usage of P_b to that of a direct measure of the inter-list connectedness in G we need to compute the largest possible size s_{bm} of a bipartite graph of vertex sets L_1 and L_2. s_{bm} is a function of the degree sequences K_1 and K_2 of L_1 and L_2 in G. For convenience, we index the vertices of the lists in order to have nonincreasing degree sequences, i.e. $k_{i,j} \geq k_{i,j+1}$, $(i = 1, 2)$. For any graph with the degree sequence of G, L_1 and L_2 induce a bipartite graph of degree sequence $D = (d_{1,1}, \ldots, d_{1,n_1}; d_{2,1}, \ldots, d_{2,n_2})$, such that

$$d_{1,i} \leq \min(k_{1,i}, n_2), \quad d_{2,j} \leq \min(k_{2,j}, n_1) \tag{23}$$

The maximal size s_{bm} is bounded as follows

$$\max\{\min(k_{1,1}, n_2), \min(k_{2,1}, n_1)\} \leq s_{bm} \leq \min\left\{\sum_i \min(k_{1,i}, n_2), \sum_j \min(k_{2,j}, n_1)\right\} \tag{24}$$

The lower bound is due to the absence of isolated vertices in G. It is possible to compute s_{bm} with the following algorithm

Algorithm B

(B_0) $w \leftarrow 0$; $R_1 \leftarrow K_1$; $R_2 \leftarrow K_2$.

(B_1) Order by decreasing values the elements of R_1. Do the same with R_2.

(B_2) Suppress from R_1 and R_2 any element less than 1. Call m_1 and m_2 the numbers of remaining elements.

(B_3) For all elements of R_1, $r_{1,i} \leftarrow \min(r_{1,i}, m_2)$. For all elements of R_2, $r_{2,j} \leftarrow \min(r_{2,j}, m_1)$.

(B_4) If $r_{1,1} \geq r_{2,1}$, $s = 1$ and $t = 2$, otherwise $s = 2$ and $t = 1$.

(B_5) $w \leftarrow w + r_{s,1}$

(B_6) $R_s^* = (r_{s,2}, \ldots, r_{s,m_s})$. $R_t^* = (r_{t,1} - 1, \ldots, r_{t,r_s+1}, r_{t,r_s+2}, \ldots, r_{t,m_t})$

(B_7) If R_s^* or R_t^* has no element greater than 0, return w. Otherwise, $R_s \leftarrow R_s^*$, $R_t \leftarrow R_t^*$ and go to (B_1).

Theorem 3 *Algorithm B returns s_{bm}.*

Proof. The proof is similar to that of Theorem (2). Algorithm B constructs a bipartite graph factorable into $St_1 \oplus St_2 \oplus \ldots \oplus St_l$, $(l \leq \max(n_1, n_2))$. St_1 is the star of maximal size (lower bound in (24)) that can be built from (K_1, K_2). Let $F = (f_{1,1}, \ldots, f_{1,n_1}; f_{2,1}, \ldots, f_{2,n_2})$ be the degree sequence of a bipartite graph with vertex sets L_1 and L_2. Let us assume that $s(F) = s_{bm}$. We define the integers s and t as follows: if $k_{1,1} \geq k_{2,1}$, $s = 1$ and $t = 2$; otherwise, $s = 2$ and $t = 1$. We call $v_{s,1}$ a vertex of degree $f_{s,1}$ in a bipartite graph $g(F)$ realizing F. Let us assume that $f_{s,1} < \min(k_{s,1}, n_t)$. Combined to the lower bound in (24) this implies that, there must be at least $f_{s,1} - \min(k_{s,1}, n_2)$ edges between vertices that are not adjacent to $v_{s,1}$. We can rewire some of these edges to connect more vertices of L_t to $v_{s,1}$ until its degree is $\min(k_{s,1}, n_t)$. The rewiring does not change the total number of edges. It gives a new bipartite degree sequence F' of size $s(F') = s(F) = s_{bm}$ with $f'_{s,1} = \min(k_{s,1}, n_2)$. Removing the star St_1 (i.e. $v_{s,1}$ and its incident edges) gives two new spike sequences K_s^* and K_t^* as defined in (B_6). It also gives a new bipartite

degree sequence F'^* such that $s_{bm} = s\left(F'^*\right) + \min\left(k_{s,1}, n_t\right)$. This implies that F'^* maximizes the size of a bipartite graph generated from K_s^* and K_t^*. Such maximal size does not change if we apply steps (B_{1-3}) of algorithm B to K_s^* and K_t^*. Therefore, we can reiterate the reasoning performed for (F, K_s, K_t) with $\left(F'^*, K_s^*, K_t^*\right)$. This proves that $g\left(F\right)$ can be rewired to have the same factorization as the one generated by algorithm B, which implies that algorithm B constructs a graph of size s_{bm}. □

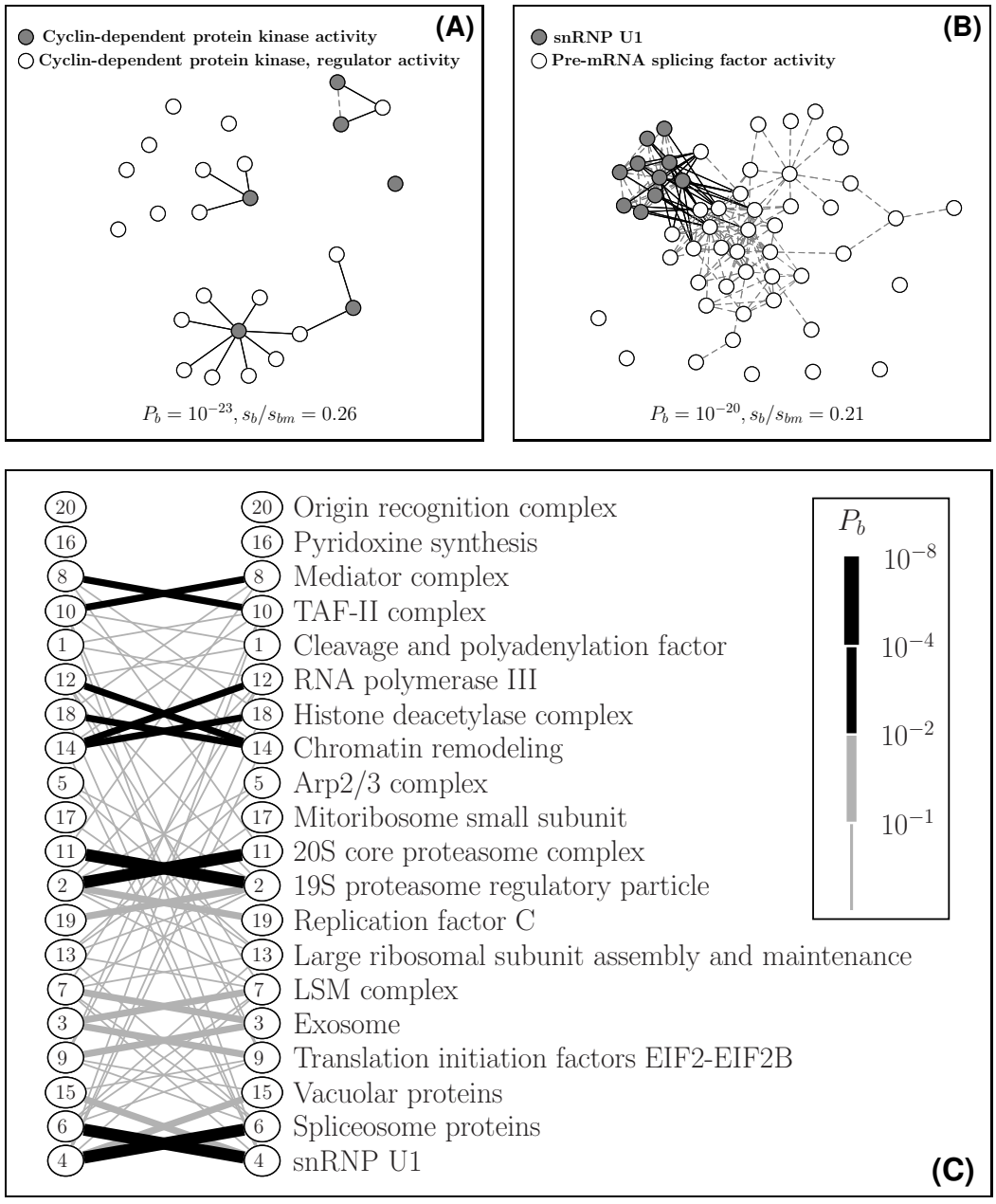

FIG. 5. (**A**) and (**B**) Two examples of bipartite subgraphs induced by pairs of GO categories and having small values of P_b. (**C**) Strength of the relations between twenty structures of high connectedness extracted from the protein network (Figure 3). A line between two lists means the existence of at least one interaction between two of their members. The color and thickness of the line reflect the strength of the relation as measured by P_b.

The upper part of Figure 5 shows two examples of pairs of GO categories with significantly small P_b values (the smallest values obtained with 1,000 random graphs of prescribed expected degrees are of the order of 10^{-5}). Inter-list edges are drawn with solid lines and intra-list edges with dashed lines. In Figure 5A the subgraph induced by $L_1 \cup L_2$ is almost bipartite, as there is only one intra-list edge. Still, the size of the bipartite subgraph induced by the vertex sets L_1 and L_2 is not maximal with $s_b/s_{bm} = 0.26$. This means that vertices of L_1 and L_2 also connect to vertices of $V - (L_1 \cup L_2)$ (edges not displayed). It also illustrates that there is no simple relation between P_b and s_b/s_{bm}. In Figure 5B the subgraph induced by $L_1 \cup L_2$ is far from being bipartite as both lists have strong intra-connectedness. However, the pair has also a significantly small value of P_b. This illustrates that bipartite structures have small P_b values, but small P_b values are not necessarily synonymous of bipartite structures. A small value of P_b only implies a strong relation in the network between two lists.

Since P_b measures the strength of relations between vertex lists, we used it to group the twenty functional modules we previously extracted from the network (Figure 3). We computed the values of P_b for all pairs of the twenty structures. Proteins present in both lists were removed before computing P_b. The results are presented in Figure 5C. A line between two lists means the existence of at least one interaction between their members. The subgraph induced by all list members breaks down into two connected components. One small component consists of the enzymes involved in pyridoxine synthesis. In the large component, small values of P_b reveal the presence of five main groups: RNA polymerase II dependent transcription ((8) and (10)), chromatin remodeling ((12), (14) and (18)), proteasome ((2) and (11)), exosome ((3), (7) and (9)) and spliceosome ((4), (6) and (15)). More generally, P_b should provide interesting means of detecting strong functional relationships between gene lists. For instance, two microarray data analyses might give two gene sets that are sparse and weakly-overlapping gene samplings of a same functional module. Given a graph of known functional relationships between the gene products, the similarity of function can be detected with P_b.

6 Conclusion

We have presented a general approach to compare the sizes of subgraphs defined as functions of prescribed vertex lists. The comparison is performed by using the likelihood of an observed edge set in a model of random graph with given expected degrees. We have derived analytical approximations for such likelihoods. The approximations rely on the graph sparseness and use the stability of the Poisson distribution, i.e. the sum of independent random variables with Poisson distribution has Poisson distribution. We have shown that our approximations hold well by comparing their numerical values to those estimated with simulated random graphs with given expected degrees. We have also presented two algorithms that compute exact upper bounds for the size of the subgraph induced by a vertex list and the size of the bipartite subgraph induced by two lists. These algorithms solve two different versions of approximate degree sequence problems for undirected graphs.

Using a random graph with given expected degrees as a null model of a network is very convenient for analytical work. There are however associated costs. We have shown that probabilities derived for random graphs with given expected degrees or random graphs with fixed degrees can differ by orders of magnitude. Also, the results of ranking protein lists based on the likelihoods of their edge-count variables are strongly correlated between both random graph models, but clearly not equivalent. Another potential drawback of random graphs with given expected degrees is the deviation of their degree distribution from that of the original network. The proportions of vertices with degree 1, 2, 3 and 4 in the protein interaction network used here are 31.2%, 19.3%, 12.3% and 7.7%. These change to 22.3%, 18.2%, 13% and 9.1% in the random graphs. In other words, some vertices of small degrees have "disappeared", they now have degree 0; and there are more vertices of high degrees. The new degree distribution is not Poissonian, as it still has a thick tail, but it is different from that of the original network. Given these limitations, the random graph with given expected degrees is still a quite reasonable null model of a network because the fluctuations of a vertex degree around its mean value k_i scale as $\sqrt{k_i}$.

In this paper we have presented a few simple applications of our analytical approximations of edge-count variable distributions. P_a was used with a simple greedy algorithm to identify subgraphs of the network

having large connectedness. Our algorithm could certainly be improved. In section 4 we have presented a simple study of the connectedness of GO categories using P_L. A more exhaustive exploration of yeast functional categories using four different protein interaction networks was recently published by Yook *et al.* (2003) . These authors used two different measures to quantify the relations in the graph between members of a functional class: the clustering coefficient (Watts and Strogatz, 1998) of the subgraph induced by a class and the size of this subgraph. These quantities were then normalized by the average values obtained when randomly reassigning the annotation to the proteins. This takes into account the degree distribution of the interaction network, but does not provide measures conditional to the degree sequences of the considered functional classes. Therefore, our approach is different and complementary to that of Yook *et al.* (2003). In section 5 we have used P_b to quantify relations in the network between twenty functional modules. One could also use P_b as a proximity measure between vertex lists to perform hierarchical clustering of a network (Newman, 2004) or hierarchical clustering of protein functional classes based on a protein interaction network (Yook *et al.*, 2004). Finally, an obvious general use case for the likelihoods we presented here is the post-processing of results of functional genomic analyses. The most common output of microarray experiments is one or several lists of genes. One way to rapidly label these lists with a function is to test them for over-representation of known functional categories, for instance, GO categories. While large amounts of gene annotation are currently available, the bulk of the knowledge about protein function resides in their biochemical interactions. The analytical approximations we have presented here provide basic tools to use this knowledge for analyzing the outputs of microarray experiments.

Acknowledgements

We thank R. Martin, J. Rees, C. Reich, D. Rose and A. Ruttenberg for very useful comments. We are grateful to T. Scarnecchia and J. Bolen for their support of this work.

References

Albert, R. and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 47–97.

Bader, G., Betel, D., and Hogue, C. 2003. BIND: the Biomolecular Interaction Network Database. *Nucl. Acids Res.* 31, 248–250.

Bader, G. and Hogue, C. 2002. Analyzing yeast protein-protein interaction data obtained from different sources. *Nat. Biotech.* 20, 991–997.

Bailey, J. 1999. Lessons from metabolic engineering for functional genomics and drug discovery. *Nat. Biotech.* 17, 616–618.

Bassler, J., Grandi, P., Gadal, O., Lessmann, T., Petfalski, E., Tollervey, D., Lechner, J., and E, H. 2001. Identification of a 60S preribosomal particle that is closely linked to nuclear export. *Mol. Cell* 8, 517–529.

Bender, E. and Canfield, E. 1978. The asymptotic number of labelled graphs with given degree sequences. *J. Combin. Theory (A)* 24, 296–307.

Braun, E., Fuge, E., Padilla, P., and Werner-Washburne, M. 1996. A stationary-phase gene in Saccharomyces cerevisiae is a member of a novel, highly conserved gene family. *J. Bacteriol.* 178, 6865–6872.

Chung, F. and Lu, L. 2002. The average distance in random graphs with given expected degrees. *Proc. Natl. Acad. Sci. USA* 99, 15879–15882.

Feller, W. 1970. *An introduction to probability theory and its applications*, volume I, chapter XI, pages 280–282. John Wiley & Sons.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* 99, 7821–7826.

Hakimi, S. 1962. On realizability of a set of integers as degrees of the vertices of a linear graph. *J. Soc. Ind. Appl. Math.* 10, 496–506.

Havel, V. 1955. A remark on the existence of finite graphs. *Časposis Pěst. Mat.* 80, 477–480.

Itzkovitz, S., Milo, R., Kashtan, N., Ziv, G., and Alon, U. 2003. Subgraphs in random networks. *Phys. Rev. E* 68, 026127.

Krause, R., Von Mering, C., and Bork, P. 2003. A comprehensive set of protein complexes in yeast: mining large scale protein-protein interaction screens. *Bioinformatics* 19, 1901–1908.

Letovsky, S. and Kasif, S. 2003. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics* 19, 197–204.

Maslov, S. and Sneppen, K. 2002. Specificity and stability in topology of protein networks. *Science* 296, 910–913.

Maslov, S., Sneppen, K., and Zaliznyak, A. 2003. Detection of topological patterns in complex networks: correlation profile of the internet. *arXiv:cond-mat/0205379* .

Milo, R., Kashtan, N., Itzkovitz, S., Newman, M., and Alon, U. 2003. Uniform generation of random graphs with arbitrary degree sequences. *arXiv:cond-mat/0312028* .

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 824–827.

Molloy, M. and Reed, B. 1995. A critical point for random graphs with a given degree sequence. *Rand. Struct. Algo.* 6, 161–179.

Newman, M. 2003a. Mixing patterns in networks. *Phys. Rev. E* 67, 026126.

Newman, M. 2003b. The structure and function of complex networks. *SIAM Review* 45, 167–256.

Newman, M. 2004. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* 69, 066133.

Newman, M., Strogatz, S., and Watts, D. 2001. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E* 64, 026118.

Pruitt, K. and Maglott, D. 2003. RefSeq and LocusLink: NCBI gene-centered resources. *Nucl. Acids Res.* 29, 137–140.

Ravasz, E., Somera, A., Mongru, D., Oltvai, Z., and Barabási, A. 2002. Hierarchical organization of modularity in metabolic networks. *Science* 297, 1551–1555.

Rives, A. and Galitski, T. 2003. Modular organization of cellular networks. *Proc. Natl. Acad. Sci. USA* 100, 1128–1133.

Samanta, M. and Liang, S. 2003. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc. Natl. Acad. Sci. USA* 100, 12579–12583.

Sanders, S., Jennings, J., Canutescu, A., Link, A., and Weil, P. 2002. Proteomics of the eukaryotic transcription machinery: identification of proteins associated with components of yeast TFIID by multidimensional mass spectrometry. *Moll. Cell. Biol.* 22, 4723–4738.

Shen-orr, S., Milo, R., Mangan, S., and Alon, U. 2002. Network motifs in the transcriptional regulation network of Escherichia coli. *Nat. Genetics* 31, 64–68.

Sierksma, G. and Hoogeveen, H. 1991. Seven criteria for integer sequences being graphic. *J. Graph Theory* 15, 223–231.

Spirin, V. and Mirny, L. 2003. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA* 100, 12123–12128.

Watts, D. and Strogatz, S. 1998. Collective dynamics of small world networks. *Nature* 393, 440–442.

Yook, S., Oltvai, Z., and Barabási, A.-L. 2004. Functional and topological characterization of protein interaction networks. *Proteomics* 4, 928–942.

Address correspondence to:
J.R. Pradines
Computational Biology, Informatics
Millennium Pharmaceuticals, Inc.
40 Landsdowne Street
Cambridge, MA 02139
Email: joel.pradines@mpi.com