Role-similarity based functional prediction in networked systems: application to the yeast proteome

Petter Holme and Mikael Huss

J. R. Soc. Interface 2005 **2**, 327-333
doi: 10.1098/rsif.2005.0046

JOURNAL OF THE ROYAL SOCIETY Interface

J. R. Soc. Interface (2005) **2**, 327–333
doi:10.1098/rsif.2005.0046
Published online 23 May 2005

Role-similarity based functional prediction in networked systems: application to the yeast proteome

Petter Holme[1,†] and Mikael Huss[2]

[1]*Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA*
[2]*Department of Numerical Analysis and Computer Science, Royal Institute of Technology, 100 44 Stockholm, Sweden*

We propose a general method to predict functions of vertices where (i) the wiring of the network is somehow related to the vertex functionality and (ii) a fraction of the vertices are functionally classified. The method is influenced by role-similarity measures of social network analysis. The two versions of our prediction scheme are tested on model networks where the functions of the vertices are designed to match their network surroundings. We also apply these methods to the proteome of the yeast *Saccharomyces cerevisiae* and find the results compatible with more specialized methods.

Keywords: functional prediction; complex networks; prediction algorithm; yeast proteome; protein functions

1. INTRODUCTION

Systems made up of entities that interact pairwise can be modelled as networks. To comprehend the emergent properties of such systems—the objective of the study of complex systems and systems biology—one approach is to investigate the global properties of the corresponding networks (Buckley & Harary 1989; Wasserman & Faust 1994; Albert & Barabási 2002; Newman 2003). In many cases, the individual entities (or vertices) have distinct functions in the system. In such cases, one can predict these functions from the vertices' position in the network, provided the wiring of the edges relates to the function of vertices. For example, a corporate hierarchy may be topped by a CEO, followed by a CFO and COO, so a chart of who reports to whom is enough to identify these positions. Another problem of much recent interest in this category is predicting protein functions (Hodgman 2000) from the networks of protein interactions (Hishigaki *et al.* 2001; Deng *et al.* 2002; Maslov & Sneppen 2002; Letovsky & Kasif 2003; Samanta & Liang 2003; Vazquez *et al.* 2003; Yook *et al.* 2004; Leone & Pagnani 2005). These methods, like other methods based on protein sequences, are important because one needs function-specific and possibly hard-to-design *in vivo*, genetic or biochemical tests to confirm a protein function, while interaction and sequence data can be obtained fairly easily.

In this paper, we propose a general method of predicting the functions of vertices in networked systems where the functions are partly mapped out. The rationale of our algorithm is to match unknown vertices with the most similar (judging from the network structure) categorized vertex and take the functions of the latter vertex as our forecast. The network similarity concept we ground our method on is related to the notion of regular equivalence (Everett 1985; Wasserman & Faust 1994) or role similarity (Luczkovich *et al.* 2003) of social network theory. Roughly speaking, two vertices are similar, in this sense, if the network looks alike from their respective perspectives. We evaluate our method on model networks where the categories of vertices reflect their placement in the network. We also apply the method to *Saccharomyces cerevisiae* protein data obtained from the MIPS database (Pagel *et al.* 2004; data extracted 23 January 2005).

2. ROLE SIMILARITY AND DEFINITION OF THE PREDICTION SCHEME

Our starting point is networked datasets where some of the vertices are functionally categorized. Between a pair (i, j) of classified vertices we can assign a functional similarity $\sigma_f(i, j)$ based on the number of functions they have and how many they share

$$\sigma_f(i,j) = J(F_i, F_j) - \langle J \rangle, \tag{2.1}$$

where F_i is the function set of i (we assume a finite number of functions), $J(\cdot)$ denotes the Jackard index $J(A, B) = |A \cap B|/|A \cup B|$ and the average is over all pairs of categorized vertices. We will later need $\sigma(i, j) = 0$ to represent neutrality, which is why we subtract the mean. The crucial thing is, of course, to assess the similarity between vertices of unknown function. We continue this section by giving an overview of the role

[†]Author for correspondence (pholme@umich.edu).

Received 23 March 2005
Accepted 23 April 2005

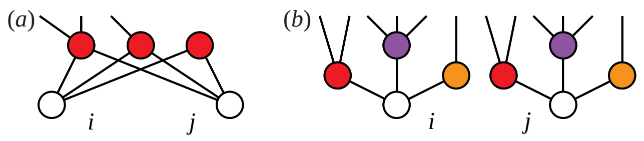


Figure 1. Illustration of structural and regular equivalence. (*a*) *i* and *j* are structurally equivalent since they have the same neighbourhoods and (*b*) *i* and *j* are regularly equivalent since regularly equivalent vertices between the neighbourhoods match. Vertices of the same colour are regularly equivalent in (*b*).

similarity concept and our iterative scheme to assign a similarity score to a pair of functionally unclassified vertices; we then describe how such a similarity measure can be combined with the functional similarity measure to form a functional prediction algorithm.

Role similarity refers to a rather broad set of concepts and related measures. Basically, the role of a vertex is determined by the characteristics of the vertices to which it is connected (Wasserman & Faust 1994).[1] Consider two vertices, i and j. If their neighbourhoods are similar, we say i and j have high role similarity. The question of how to define the similarity of the neighbourhoods Γ_i and Γ_j leads to two different concepts. One choice matches the *identity* of vertices in the neighbourhood. This leads to the *structural equivalence* relation (figure 1) which is true if $\Gamma_i = \Gamma_j$. Another way to compare neighbourhoods is to match the *similarity* of vertices in the neighbourhood which gives the concept of *regular equivalence* (figure 1)—if one can pair the vertices in Γ_i with the vertices in Γ_j such that each pair is regularly equivalent, then i and j are also regularly equivalent. Since vertices with the same functions need not, in general, be close, we will need a similarity score measuring how close to regular equivalence two vertices are. Following Jeh & Widom (2002) and Blondel *et al.* (2004) we define a similarity score based on iterating the regular equivalence principle: 'two vertices are similar if they are pointed to, or point to, vertices that are themselves similar.' In a general networked system, one may have many distinct kinds of edges. Assuming we have $R \in [1, \infty)$ edge types, one implementation of the regular equivalence principle is to sum the similarities between vertices of the neighbourhoods

$$\sigma^{\mathrm{I}}_{n+1}(i,j)=\sum_{r=1}^{R}\left[\sum_{i'\in\Gamma^{\mathrm{in}}_{i,r}}\sum_{j'\in\Gamma^{\mathrm{in}}_{j,r}}\sigma^{\mathrm{I}}_{n}(i',j')+\sum_{i'\in\Gamma^{\mathrm{out}}_{i,r}}\sum_{j'\in\Gamma^{\mathrm{out}}_{j,r}}\sigma^{\mathrm{I}}_{n}(i',j')\right], \tag{2.2}$$

where $\sigma^{\mathrm{I}}_n(i, j)$ is the similarity between i and j after the nth iteration and $\Gamma^{\mathrm{in}}_{i,r}$ is the in-neighbourhood of i with respect to r-edges. Whenever a pair of classified vertices (i, j) appear in equation (2.2) we use the $\sigma_f(i, j)$ value of equation (2.1). That is, we assume the pre-existing functional classification is more accurate than the role similarities and, hence, do not update the former. To avoid overflow problems we rescale all similarities so that $\max_{ij}|\sigma^{\mathrm{I}}_n(i,j)| = S$ after each iteration. S will be a parameter setting the relative importance of the functional similarities compared with the subsequent assessments of σ. We break the iteration when the sum, before the normalization, has not changed by more than a 10^{-8}th of its previous value. Typically this happens after three or four iterations. During the first iteration of equation (2.2), $\sigma(i, j)$ is determined by the similarity of neighbourhoods of i and j. As the iterations proceed, vertices further and further away are taken into account. For details on the rationale, existence and uniqueness of a fixed point of the iterations, see Jeh & Widom (2002).

By the equation (2.2) definition, vertices of high degree (number of neighbours) will appear more similar to the average other vertex than low-degree vertices. To compensate for this effect one may divide by the appropriate degrees (numbers of neighbours) to obtain

$$\sigma^{\mathrm{II}}_{n+1}(i,j)=\sum_{r=1}^{R}\left[\frac{1}{k^{\mathrm{in}}_{i,r}k^{\mathrm{in}}_{j,r}}\sum_{i'\in\Gamma^{\mathrm{in}}}\sum_{j'\in\Gamma^{\mathrm{in}}}\sigma^{\mathrm{II}}_{n}(i',j') + \frac{1}{k^{\mathrm{out}}_{i,r}k^{\mathrm{out}}_{j,r}}\sum_{i'\in\Gamma^{\mathrm{out}}}\sum_{j'\in\Gamma^{\mathrm{out}}}\sigma^{\mathrm{II}}_{n}(i',j')\right], \tag{2.3}$$

where $k^{\mathrm{in}}_{i,r}$ is the in-degree of i with respect to r-edges. From now on we call $\sigma^{\mathrm{I}}(i, j) = \sigma^{\mathrm{I}}_{\infty}(i, j)$ of equation (2.2) and $\sigma^{\mathrm{II}}(i, j)$ of equation (2.3) the I- and II-similarity between i and j, respectively.[2] One can further develop this measure by adding different weights to the different kinds of vertices; adding stronger weights to the more reliable edges. Since the choice of weights is highly problem-dependent and since the results for the model and protein interaction networks are quite robust to this choice we will henceforth only discuss the case of equal weights.

In general we can now define our prediction scheme as follows.

(i) For vertex pairs with at least one unclassified vertex initialize $\sigma_0(i, j)$ to 0 if $i \neq j$ and to $1 - \langle J \rangle$ otherwise.

(ii) Calculate the similarity scores for all pairs of unique vertices such that at least one is unclassified.

(iii) For an unclassified vertex i, predict the function set $F_{\hat{i}}$ where $\hat{i}$ is the classified vertex with the highest similarity to i. If $\hat{i}$ is not unique, but a set $\hat{I} = \{\hat{i}_1, \ldots, \hat{i}_m\}$ has the highest similarity to i, then let the set G of functions present in more than half of the set of js be your guess. If G is empty, let F_j for a random $j \in \hat{I}$ be the guess.

The diagonal elements will have maximal functional similarity (which is why we set them to $1 - \langle J \rangle$ in step

[1]Note that the nomenclature is somewhat ambiguous. Another use of 'role' is to say that vertices with similar values to vertex-specific structural measures have the same role (Lusseau & Newman 2004; Guimerà & Nunes Amaral 2005).

[2]There are more possible implementations of the mentioned similarity principle. We choose these because they perform better on our two test cases. For other problems, it might be the case that another implementation listed in Jeh & Widom (2002) performs better.

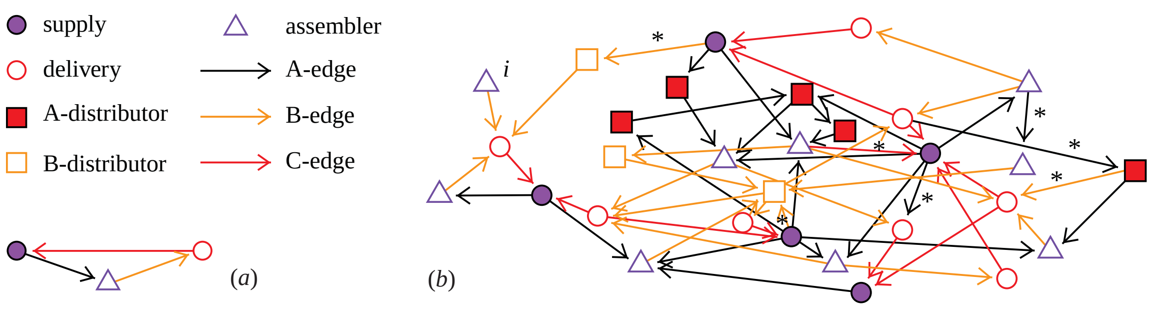


Figure 2. Model networks where vertex function and position are related. (*a*) shows the initial network. (*b*) shows a realization with 30 vertices and rewiring probability $r=0.1$. An asterisk indicates a rewired edge.

(ii)), otherwise we assume neutrality. The back-up selection rules in step (iii) will typically be needed when unclassified vertices are structurally equivalent to classified vertices and the use of the majority rule, instead of a random guess, will compensate for occasional errors in the assignment of functions to classified proteins. As mentioned above, the functional classification is assumed to be more accurate than the role-similarities, and it is thus sensible to choose S in the interval $[0,1-\langle J\rangle]$. The appropriate value of S, the parameter setting the relative importance of the functional similarities, is problem dependent. We will use $S=0.8$, which is within the chosen interval for both our two test cases. To summarize, we have proposed two versions of our prediction scheme, scheme I and II, corresponding to I- and II-similarity.

3. APPLICATION TO MODEL NETWORKS

To test our prediction algorithm, we construct model networks where the assigned functions of the vertices correspond to their position in the network. We test the algorithm's size scaling and performance in subideal conditions by randomly perturbing the network.

3.1. Definition of the model networks

In defining our model, we will metaphorically use the flow of raw material, products and information in a manufacturing system. For our purpose we only need networks where the functions of vertices correspond to their position in their network surroundings—we will not further motivate its relevance as a model for manufacturing networks. We assign five distinct functional classes of the vertices: The *supply* vertices are the source of the raw material which flows along *A-edges* to *assembler* vertices. The assembled products are transported via *B-edges* to *delivery* vertices that dispatch the products. From the delivery vertices, informational feedback is sent to the supply vertices through *C-edges*. Furthermore, the A- and B-edges can fork at *A-* and *B-distributor* vertices.

The precise definition of the model is as follows: Start with the kernel shown in figure 2*a*, then grow the network vertex by vertex. At each iteration, assign, with equal probability, one of the above functions to the new vertex. Then, depending on the assigned function, form edges including the new vertex as follows.

Supply: add an A-edge to an assembler or A-distributor and a C-edge from a delivery vertex.

Assembly: add an A-edge from a supply vertex or A-distributor vertex and a B-edge to a delivery vertex or B-distributor.

Delivery: add a B-edge from an assembler or B-distributor and a C-edge to a supplier.

A-distribution: add an A-edge from a supply or A-distributor vertex and an A-edge to an assembler or A-distributor.

B-distribution: add a B-edge from an assembler or B-distributor vertex and a B-edge to a delivery vertex or B-distributor.

The vertex to be attached to the new vertex is chosen, given its functional category, with uniform randomness. Note that the number of edges will on average be twice the number of vertices (two edges are added per vertex).

From the definition so far, any vertex is identifiable from its neighbourhood—a vertex with incoming C-edges and out-going A-edges is a supplier, and so on. The wiring of the edges and the functional classification in real datasets are seldom perfect. To test the prediction scheme under more realistic circumstances we randomize the network as follows; after generating a network according to the above scheme, we sequentially go through all edges. With probability r, we detach the from-side of an edge and reattach it to a randomly chosen vertex so that self-edges or multiple edges (of the same type—A, B or C) are not formed. We rewire the to-side likewise with the same probability. A realization of the algorithm is displayed in figure 2*b*. After the rewiring there is not necessarily enough information to classify a vertex; i in figure 2*b* is an assembler but could just as well have been a B-distributor.

3.2. Prediction performance

To test our prediction scheme we mark a random set of aN, $a\in(0,N)$, vertices unclassified. Then we predict the function of these vertices and let the average fraction of correctly predicted vertices s be our performance measure. figure 3 shows s for $a=1/50$ and different network sizes as a function of the rewiring probability r. In the small-r limit, the I-similarity prediction scheme makes an almost flawless job with $s>99.9\%$ for $N\geq500$. Note that since we have five distinct functions random guessing could not do better than $s=1/5$. This value, $s=1/5$, is by necessity attained in the random limit $r=1$. For small r-values the scheme II performs best, but if $r \lesssim 0.2$, scheme I performs slightly better. The size convergence for scheme I is faster, so II may outperform I in the large network limit. To understand the performance of the different schemes we note that scheme I has a tendency to match an unknown vertex to a known vertex of high degree. When $r=0$, this effect leads to some prediction errors for scheme I. However, the redundant information about high degree vertices makes the more robust to minor perturbations. (If one edge is rewired, a fraction $1-1/k$ of the edges is still correct, which is large if the degree k is large.) This redundancy is probably the reason for the slower decay of the $s(r)$-curves of scheme I compared with the corresponding curves of scheme II.

We observe that the performance increases with the systems size for both schemes. This is an important effect since databases generally grow in size—our prediction scheme will thus be more accurate with time. We surmise that this is because the bigger the network gets, the more likely it is that there is a very good matching. This is an effect local methods (taking only the surrounding of a vertex into account) could not utilise. A full explanation of this effect lies beyond the scope of this paper.

4. PREDICTING PROTEIN FUNCTION IN YEAST

4.1. Functional prediction of proteins

Experimentally specifying protein functions requires demanding and potentially expensive tests. Obtaining good guesses of the functions of an unknown protein is very useful. During the last decade, a great number of methods have been suggested for protein functional prediction, including methods based on sequence or structure alignments (Pawlowski *et al.* 2000; Irving *et al.* 2001), attributes derived from collections of sequences or structures (Dobson & Doig 2003; Jensen *et al.* 2003), phylogenetic profiles (Pellegrini *et al.* 1999) and analysis of protein complexes (Gavin *et al.* 2002). Much recent work has concentrated on functional prediction based on protein–protein interaction data. Many of these are specialized methods that exploit specific features of protein–protein interaction data (Marcotte *et al.* 1999*a*,*b*; Hodgman 2000; Schwikowski *et al.* 2000; Hishigaki *et al.* 2001; Deng *et al.* 2002; Letovsky & Kasif 2003; Samanta & Liang 2003; Strong *et al.* 2003; Vazquez *et al.* 2003), such as the empirical observation that those vertices which interact physically are likely to share some functionality. The more general approaches (Hishigaki *et al.* 2001; Deng *et al.* 2002) are local, in the sense that they are based on pairwise statistics only. For this reason, they may not share the advantageous size scaling properties of our method.

4.2. Applying the method to protein data

There are two types of large-scale network data available for *S. cerevisiae*: 'physical' and 'genetic' protein–protein interactions. The terms 'physical' and 'genetic' refer to the type of experiment used to deduce the interaction. The genetic experiments are based on mutation studies and the evidence from them is of a more indirect nature. We therefore distinguish between physical and genetic edges. All edges are undirected. Our dataset, derived from the MIPS database, has $N=4580$ linked together by 5129 genetic regulation edges and 7434 physical interaction edges. We removed duplicates, self-edges and interactions where one or both of the interacting substances were not proteins. The assigned functions are arranged in a hierarchical fashion, according to the FunCat categorization scheme (Ruepp *et al.* 2004) used by the MIPS database. The first level contains the coarsest description of a protein's function (such as 'metabolism'), the second level is more specified (such as 'amino acid metabolism') and so on. We tested our algorithm on the first- and second-level of this hierarchy and, thus, treat functions that differ in a finer classification as equal. There are three categories with no substantial functional information—'ubiquitous expression', 'classification not yet clear-cut' and 'unclassified proteins'. We considered vertices with no other assigned categories than those listed above.

In figure 4, we show a small example of scheme II in action on the yeast data. Suppose YJL191w is to be classified (we know it has the level-1 functions 'protein with binding function…' and 'protein synthesis'). The classified protein with highest similarity is YOR133w. This is because YNL041c, which interacts physically with YJL191w, is functionally identical (at level one of the hierarchy) to YBR068c, which is physically linked to YOR133w. Similarly, YJL191w is genetically linked with YCR031c, which shares one functional category with YDR385w, which is genetically linked with YOR133w. These two features give a high similarity score to the pair YJL191w and YOR133w, so scheme II guesses that YJL191w has the functional category 'protein synthesis' but misses the 'protein with binding function…' category.

4.3. Performance of the scheme

For the previously described test networks we know a priori that the number of functions to be predicted is one. The same may be true for a variety of systems but not for proteins. With the number of functions as one variable in the prediction problem we proceed to replace the success rate s with the two measures *precision* s_+ and *recall* s_- (the names are borrowed from corresponding quantities in the text-mining literature, see Raghavan *et al.* (1989) and references therein),

$$s_+ = \left\langle \frac{n_c}{f_*} \right\rangle \quad \text{and} \quad s_- = \left\langle \frac{n_c}{f} \right\rangle, \qquad (4.1)$$

where n_c is the number of correctly predicted functions, f is the real number of functions and f_* is the number of predicted functions. $1-s_+$ is thus the expected fraction of false-positive predictions (and similarly for s_-). Both these measures take values in the interval [0, 1] with 0 meaning that no function is predicted correctly and 1

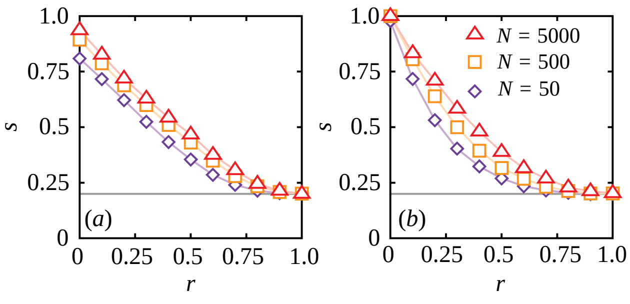
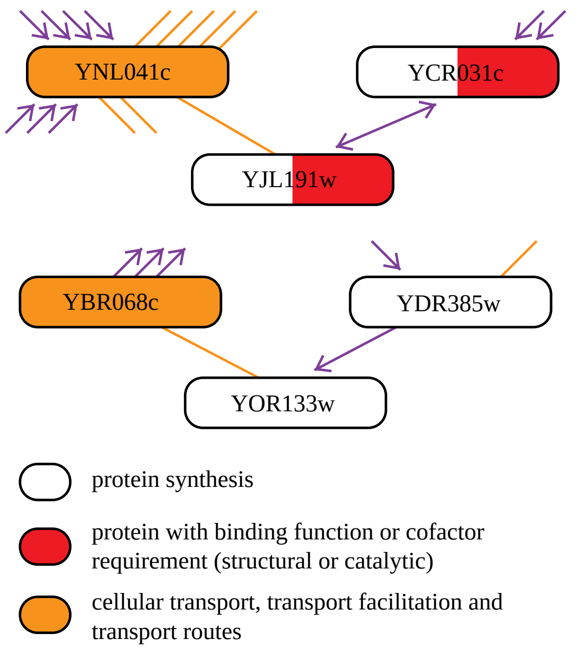


Figure 3. The fraction of correctly predicted functions s for our model networks as a function of the rewiring probability r; (a) shows the results based on I-similarities and (b) is the corresponding plot for II-similarities. The points are averaged over approximately 1000 runs of the network construction and prediction scheme with $a = 1/50$. Error bars are smaller than the symbol size. The horizontal line marks the limit of random guessing (0.2).



YNL041c
YCR031c
YJL191w
YBR068c
YDR385w
YOR133w
protein synthesis
protein with binding function or cofactor requirement (structural or catalytic)
cellular transport, transport facilitation and transport routes

Figure 4. Example of yeast protein prediction from the first-level functional data with scheme II. When YJL191w is marked unknown it gets matched with YOR133w because their surroundings look similar. The arrowed lines mark genetically determined interactions, while other lines represent physically determined interactions.

representing perfect prediction. The averages are over the set of predicted functions in the same kind of leave-one-out estimates as performed for the test networks.

We follow Deng *et al.* (2002) and Vazquez *et al.* (2003) and use the neighbourhood counting method (NCM) of Schwikowski *et al.* (2000) for reference values. This method assigns the f_* most frequent functions among the neighbours of the physical interaction network to the unknown protein (i.e. f_* is a parameter of this model). Considering its simplicity compared with the more elaborate procedures listed above, this is a remarkably efficient method. In our implementation, if the f_*th function is not unique we select it randomly. Thus, proteins with no neighbours are randomly assigned f_* functions. Precision and recall values are displayed in table 1. We use $f_* = 2$ for the NCM which is the closest value to the average number of functions per protein for both levels one and two in our dataset. We note that our methods have 10–45% higher precision values, but 2–18% lower recall values compared with the NCM. As the distinction becomes finer between the functions with the level of the FunCat hierarchy, it is not unexpected that the values of s_+ and s_- decrease from level 1 to level 2. The values may look low compared with similar tables in other papers on protein prediction, but these often use other performance measures (such as counting the fraction of proteins with at least one correctly predicted function, and so on) or do not include low-degree vertices. We note that, like the more disordered test networks, scheme II gives better performance in general (typically having better recall values, but slightly worse precision values).

5. SUMMARY AND DISCUSSION

We have proposed methods for predicting the function of vertices in networked systems where the function of a vertex relates to its position. The principle behind our scheme is role equivalence as related to the regular equivalence concept of social network analysis, that is, vertices are similar if the network, as seen from the respective vertices, looks similar. We make two extensions to the method proposed by Jeh & Widom (2002) and Blondel *et al.* (2004) for networks where some of the vertices are functionally categorized. An uncategorized protein is predicted by copying the functions of the vertex with highest role similarity. Our schemes, corresponding to our two role similarities, are tested on model and protein interaction networks. These two test cases are intentionally different to test the general applicability of our approach. A summary of the different network structural properties is found in table 2. The model networks are designed to have a correspondence between the function of the vertex and their network surrounding. This correspondence can be tuned by a randomization parameter. We find that the performance of both schemes increases with the system size (the fraction of unknown vertices and rewired edges is fixed), which makes the applicability of our methods increase with time (as databases, in general, tend to grow). The differences between scheme I and II can be described by the fact that scheme I gives (compared with scheme II) a higher similarity to vertex pairs containing a high-degree vertex. In addition to the model networks we also apply our method to the *S. cerevisiae* proteome. We use the networks of protein–protein interactions and obtain results that compare well with standard methods designed solely with protein functional prediction in mind. We do not claim that our method outperforms the best specialized protein prediction methods—our aim is to construct a global method for general functional prediction and most protein functional prediction schemes would perform poorly on our test networks. The ideas of this paper might, however, contribute to future, more elaborate methods for prediction of protein functions.

Table 1. The performance of our methods (scheme I and scheme II) compared with the neighbourhood counting method (NCM) of Schwikowski *et al.* (2000).

(s_+ is the average fraction of correct predictions among the predicted functions averaged over all the classified proteins. s_- is the average fraction of correct predictions among the actual functions. For the NCM predictions we use the parameter value $f_*=2$ (i.e. two predicted functions per protein). Level 1 and 2 refers to the cut-off levels of the FunCat scheme (Ruepp *et al.* 2004)—'level 1' means that functions of the same first class are considered equal and so on. The numbers in the brackets are given the standard error in units of the last decimal.)

	level 1			level 2		
	NCM	scheme I	scheme II	NCM	scheme I	scheme II
s_+	0.269(6)	0.392(6)	0.337(6)	0.199(5)	0.238(6)	0.220(6)
s_-	0.354(6)	0.291(5)	0.346(7)	0.252(6)	0.199(5)	0.231(6)

Table 2. Qualitative comparison between the network structure of the model and yeast protein networks.

(For the degree distribution we assume the edges to be undirected and of the same type. The degree of distribution of protein interaction networks is known to be heavily right-skewed (Jeong *et al.* 2001). This is true for our dataset as well. (We do not dwell further on the exact functional form of the distribution.) The exponential functional form of the model networks follows from the fact that the model reduces to the 'uniform attachment model' of Barabási *et al.* (1999) when the edge and vertex categories are disregarded. The degree–degree correlations refer to whether high-degree vertices tend to be connected to other high-degree vertices (which would give a positive correlation) or to low-degree vertices. Negative degree–degree correlations for protein networks were first observed by Maslov & Sneppen (2001); see also Newman (2003).)

	model	yeast protein
directed	yes	no
number of functional classes	5	18 (level 1), 96 (level 2)
number of edge-type, R	3	2
degree of distribution	exponential	fat-tailed
degree–degree correlations	neutral	negative

The basic advantage of our method, as we see it, is that it is a very general method that should apply to functional prediction in many systems. Indeed, our model networks and the protein interaction networks are very different, both in the network structure and the distribution of the functions. Moreover, it makes use of global network information, giving a performance that does not decrease as the systems gets larger. The fact that it is a truly global algorithm—the prediction of every vertex's functions takes wiring of the whole network into account—makes it rather slow (compared with, for example, specialized protein functional prediction methods, such as the one proposed by Schwikowski *et al.* (2000)). The execution time-scales are $O(M^2)$ (where M is the total number of edges), but datasets of 10^4–10^5, which cover, for example, the size of proteomes of known organisms, should be manageable for present day computers. We believe the problem of functional prediction in different types of networked systems remains unresolved—in terms of utilising the characteristics in both general and more specific systems.

The authors thank Micha Enevoldsen, Elizabeth Leicht and Mark Newman for comments.

REFERENCES

Albert, R. & Barabási, A.-L. 2002 Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47–98.
Barabási, A.-L., Albert, R. & Jeong, H. 1999 Mean-field theory for scale-free random networks. *Physica A* **272**, 173–187.
Blondel, V. D., Gajardo, A., Heymans, M., Senellart, P. & van Dooren, P. 2004 A measure of similarity between graph vertices: applications to synonym, extraction and web searching. *SIAM Rev.* **46**, 647–666.
Buckley, F. & Harary, F. 1989 *Distance in graphs*. Redwood City: Addison-Wesley.
Deng, M., Zhang, K., Mehta, S., Chen, T. & Sun, F. 2002 Prediction of protein function using protein-protein interaction data. In *Proc. IEEE Computer Society Bioinformatics Conf. (CSB 02)*, pp. 197–207. Stanford, CA: IEEE Computer Society.
Dobson, P. D. & Doig, A. S. 2003 Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.* **330**, 771–783.
Everett, M. G. 1985 Role similarity and complexity in social networks. *Soc. Networks* **7**, 353–359.
Gavin, A. C. *et al.* 2002 Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**, 141–147.
Guimerà, R. & Nunes Amaral, L. A. 2005 Functional cartography of complex metabolic networks. *Nature* **433**, 895–900.
Hishigaki, H., Nakai, K., Ono, T., Tanigami, A. & Tagaki, T. 2001 Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast* **18**, 523–531.
Hodgman, T. 2000 A historical perspective on gene/protein functional assignment. *Bioinformatics* **16**, 10–15.
Irving, J. A., Whisstock, J. C. & Lesk, A. M. 2001 Protein structural alignments and functional genomics. *Proteins* **42**, 378–382.
Jeh, G. & Widom, J. 2002 SimRank: a measure of structural-context similarity. In *Proc. Eighth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 197–207. New York: Association for Computing Machinery.

Jensen, L. J., Staerfeldt, H. & Brunak, S. 2003 Prediction of human protein function according to Gene Ontology categories. *Bioinformatics* **19**, 635–642.

Jeong, H., Mason, S. P., Barabási, A.-L. & Oltvai, Z. N. 2001 Lethality and centrality in protein networks. *Nature* **411**, 41–42.

Leone, M. & Pagnani, A. 2005 Predicting protein functions with message passing algorithms. *Bioinformatics* **21**, 239–247.

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

Luczkovich, J. J., Borgatti, S. P., Johnson, J. C. & Everett, M. G. 2003 Measuring trophic role similarity in food webs using regular equivalence. *J. Theor. Biol.* **220**, 303–321.

Lusseau, D. & Newman, M. E. J. 2004 Identifying the role that animals play in their social networks. *Proc. R. Soc. B* **271**, 477–481.

Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O. & Eisenberg, D. 1999*a* Detecting protein functions and protein–protein interactions from genome sequences. *Science* **285**, 751–753.

Marcotte, E. M., Pellegrini, M., Thompson, M. J., Yeates, T. O. & Eisenberg, D. 1999*b* A combined algorithm for genome-wide prediction of protein function. *Nature* **402**, 83–86.

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

Newman, M. E. J. 2003 The structure and function of complex networks. *SIAM Rev.* **45**, 167–256.

Pagel, P. *et al.* 2004 The MIPS mammalian protein–protein interaction database. *Bioinformatics* **21**, 832–834.

Pawlowski, K., Jaroszewski, L., Rychlewski, L. & Godzik, A. 2000 Sensitive sequence comparison as protein function predictor. *Pac. Symp. Biocomput.*, 42–53.

Pellegrini, M., Marcotte, E., Thompson, M. J., Eisenberg, D. & Yeates, T. O. 1999 Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl Acad. Sci. USA* **96**, 4285–4288.

Raghavan, V. V., Jung, G. S. & Bollmann, P. 1989 A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.* **7**, 205–229.

Ruepp, A. *et al.* 2004 The FunCat a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.* **32**, 5539–5545.

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

Schwikowski, B., Uetz, P. & Fields, S. 2000 A network of protein–protein interaction in yeast. *Nat. Biotechnol.* **18**, 1257–1261.

Strong, M., Graeber, T. G., Beeby, M., Pellegrini, M., Thompson, M. J., Yeates, T. O. & Eisenberg, D. 2003 Visualization and interpretation of protein networks in *Mycobacterium tuberculosis* based on hierarchical clustering of genome-wide functional linkage maps. *Nucleic Acids Res.* **31**, 7099–7109.

Vazquez, A., Flammini, A., Martian, A. & Vespignani, A. 2003 Global protein function prediction in protein–protein interaction networks. *Nat. Biotechnol.* **21**, 697–700.

Wasserman, S. & Faust, K. 1994 *Social network analysis: methods and applications*. Cambridge: Cambridge University Press.

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