



Generating protein interaction maps from incomplete data: application to fold assignment

Michael Lappe^{1,*}, Jong Park^{1,3}, Oliver Niggemann² and Liisa Holm¹

¹Structural Genomics Group, The European Bioinformatics Institute, EMBL Outstation, Cambridge CB10 1SD, UK, ²Department of Mathematics and Computer Science, University of Paderborn, Warburgerstr. 100, 33098 Paderborn, Germany and ³Present address: MRC-DUNN, Human Nutrition Unit, The European Bioinformatics Institute, Hills Road, Cambridge CB2 2XY, UK

Received on February 6, 2001; revised and accepted on April 2, 2001

ABSTRACT

Motivation: We present a framework to generate comprehensive overviews of protein-protein interactions. In the post-genomic view of cellular function, each biological entity is seen in the context of a complex network of interactions. Accordingly, we model functional space by representing protein-protein-interaction data as undirected graphs. We suggest a general approach to generate interaction maps of cellular networks in the presence of huge amounts of fragmented and incomplete data, and to derive representations of large networks which hide clutter while keeping the essential architecture of the interaction space. This is achieved by contracting the graphs according to domain-specific hierarchical classifications. The key concept here is the notion of induced interaction, which allows the integration, comparison and analysis of interaction data from different sources and different organisms at a given level of abstraction.

Results: We apply this approach to compute the overlap between the DIP compendium of interaction data and a dataset of yeast two-hybrid experiments. The architecture of this network is scale-free, as frequently seen in biological networks, and this property persists through many levels of abstraction. Connections in the network can be projected downwards from higher levels of abstraction down to the level of individual proteins. As an example, we describe an algorithm for fold assignment by network context. This method currently predicts protein folds at 30% accuracy without any requirement of detectable sequence similarity of the query protein to a protein of known structure. We used this algorithm to compile a list of structural assignments for previously unassigned genes from yeast. Finally we discuss ways forward to use interaction networks for the prediction of novel protein-protein interactions.

Availability: www.ebi.ac.uk/~lappe/FoldPred/

Contact: lappe@ebi.ac.uk

OPERATIONALIZING THE NOTION OF FUNCTION

As more experimental data on protein interaction becomes available, it will be of critical importance to integrate and compare the data derived from different sources. The analysis of interaction data aims to reveal the organizational principles of cellular networks and to describe the architecture of biochemical and genetic networks. A key difficulty on the way is the incomplete experimental characterization of most biological systems. To fill the information gaps, we need to find ways of generalization from individual experimental evidence (e.g. protein-protein interactions) to higher level biological entities (e.g. protein families) in order to generate structural and functional annotations. The notion that cellular functions are the outcome of molecular interactions among biological entities is reflected in the post-genomic approach to cellular function, where molecular biological entities are seen as nodes in a complex network of interactions (Eisenberg, 2000). This view helps to operationalize the notion of function, since it allows to build models of the cellular circuitry as undirected graphs $G = (V, E)$. Here the set of vertices V represents proteins connected by a set of edges E , which represent the interactions.

A big problem with available interaction data is that it is fractionated and the data files come in the form of binary sets. Each data record represents a single edge $e = (v, w) \subseteq E$ in our graph and denotes that protein v interacts with protein w . Experimental techniques such as yeast-two-hybrid experiments are suitable for genome-wide screening but yield large numbers of both false positives and false negatives. Therefore, our knowledge of functional space in terms of protein interactions is still

*To whom correspondence should be addressed.

far from being complete. Clearly, in order to generate a comprehensive interaction map from fractionated and incomplete data, the input sets of binary interactions have to be merged in a meaningful way by joining their nodes. Within a given organism, it is easily possible to link the given set of edges (interactions) via identical node labels, as was done for yeast (Schwikowski et al., 2000). However, for many genomes we know only the gene complement (set of nodes) and need to infer the interaction network by mapping information from other sources and organisms.

We were motivated to the present work by the observation that even for such a simple model organism as yeast, with ~ 6000 genes, the current interaction network is far too complex (densely connected) to be perceived as a whole (Mayer & Hieter, 2000). Therefore, especially for the upcoming amounts of experimental data from yeast-two-hybrid, gene expression, co-immunoprecipitation, TAP and protein array experiments etc., it is important to have means to condense the interaction data into functional modules in a comprehensive way. It is intuitively clear that, like in aerial archaeology, where the structure of an ancient settlement is invisible from the ground and only becomes apparent from aerial photographs, we have to take a step back to get an idea of the bigger picture. In this paper, we present a general framework that is able to integrate data at different levels of abstraction coming from different sources and different species, and is able to condense the amount of data in a meaningful way. As a result, we get a glimpse at the overall architecture and topology of cellular networks.

CLUSTERING OF INTERACTION INFORMATION (CONTRACTION)

Here we apply a graph-theoretic framework to generate protein interaction maps from a given set of binary interactions obtained from experimental data. Such a set can be seen as a graph $G = (V, E)$. Initially, each node $v \in V$ is connected to just one other node $w \in V$, representing experimental evidence that the protein represented by v interacts with the protein represented by w . So how do we go about joining these binary interactions in order to get an overview?

It is fairly obvious that for interaction data derived from the same species it is possible to assign a finite set of protein names L to the interacting partners represented as the set of nodes V . This defines a labeling function $l : V \rightarrow L$, where l represents our knowledge about the identity of proteins within the proteome. Then it is straightforward to link the given interactions via nodes with identical labels. The method described above does not work across species, unless we have a way for identifying the same (homologous) proteins in different species.

Biologists distinguish two kinds of homology; for practical purposes, orthologues have exactly the same biological function (including interactions) while paralogues may have evolved new properties or lost some of the ancestral properties. One method would be to link the interaction networks of different species by orthologous genes (COGs) (Tatusov et al., 1997) which would be an extension of the above mentioned identity by a lookup-table defining the same node across different species. A more general way is via sequence similarity, using the assumption that proteins above a certain sequence similarity undergo similar interactions. So in this case, the labeling function would be based on a threshold t for sequence similarity.

We are going about these obvious notions in this rather awkward way to see that building a genome-wide interaction map is formally the same as any other contraction of a graph (clustering of the nodes) and differs only in the way the identity function l is defined.

There are two general classes of clustering approaches, which can be used to further condense the interaction graph. The first approach is unsupervised clustering, which means treating the network as an arbitrary graph and clustering it according to its distribution of edges, without any domain-specific knowledge. Decomposition of the graph into highly connected subcomponents reveals communication hubs and functional modules (sets of genes that are involved in a cellular process). The second approach is supervised clustering, whereby one assigns a set of known biological properties to all the nodes and clusters the nodes based on these attributes. This is exemplified by the clustering according to subcellular localization done by Schwikowski et al. (Schwikowski et al., 2000) to see how much crosstalk there is between different cellular compartments, i.e. edges that link the given clusters.

We follow here the second approach and use a given clustering from the biological domain in order to analyze the underlying interaction network under the aspect of the knowledge encoded in this clustering. Since the set of binary interactions is now seen as interactions within and between higher-level entities (alike subcellular compartment) this leads us to the definition of induced interactions at a higher level of abstraction.

INDUCED INTERACTIONS AND THE LEVEL OF ABSTRACTION

Let $G = (V, E)$ be the graph abstraction of the biological system under consideration (the interaction network). Then $C(G) = (C_1..C_n)$ is a decomposition of G into n subgraphs induced on the C_i , if $\cup_{C_i \in C} C_i = V$ and $C_i \cap C_{j \neq i} = \emptyset$. The induced subgraphs $G(C_i)$ are called clusters. The set of edges $E_c \subseteq E$ consists of the

set of edges between the clusters. Then the contraction $H = \langle C(G), E_c \rangle$ is called the connectivity structure. For a given (interaction) graph $G = (V, E)$ and a given (hierarchical) clustering C , let's consider an edge $e = (v, w) \in E$, where v and w are nodes from the cluster C_i and $C_{j,j \neq i}$, respectively. Then e induces an edge in the contraction H between the two nodes representing the clusters C_i and C_j .

Translating this to more biological terms, an interaction $e = (v, w)$ induces an interaction between two higher level biological entities (clusters, or inner nodes in a hierarchical classification) A and B , if and only if, v and w are descendants (or members of the clusters) of A and B . Then we call $i = (A, B)$ an induced interaction of e given the classification (or clustering) C . More verbally, an induced interaction at a given level of abstraction is the interaction between higher-level biological entities (e.g. subcellular compartments, protein families) induced by a clustering of lower-level entities given their interactions.

As discussed above, the clustering of the interaction network is not limited to a single step. Since different clustering methods can be applied consecutively, this is logically equivalent to a hierarchical clustering. Considering a hierarchical clustering, this leads to the concept of level of abstraction: The more we contract the interaction network, the more we abstract from the underlying experimental data. Higher abstraction trades a gain in generality for a loss of specificity. In analogy to moving upwards to more general classes in object-oriented hierarchies, we call moving towards the root of a hierarchical classification to a higher level of abstraction *upcasting*. Examples of upcasting are the abstraction of the underlying protein-to-protein network to interactions between cellular compartments, functional modules, or structural classes (depending solely on the hierarchy used for classification).

The complementary procedure of projecting a given network between higher level entities back down to lower levels of abstraction is what we call *downcasting*. This could be, for example, downcasting a network of protein family interactions back to the protein or domain level. Generalizing from individual protein interactions by upcasting them to protein family level and then downcasting them back to the protein level is a way to predict possible interactions. At the same time it is not reasonable to predict that all members of protein-families A and B interact based on the instance of a single observed interaction, so additional filters are needed in the downcasting procedure. A real-life analogy of the problem here is to claim that everybody likes all their in-laws, which is obviously a gross overprediction. The development of specific filters and descriptors for protein interactions is required before we will be able to generate accurate genome-wide predictions of protein interaction networks. However, it is already possible to embed a given

protein into a given network based on its interaction data alone.

In the next two sections, we demonstrate the application of the framework described above. Upcasting of different sources of interaction data to the SCOP superfamily level resulted in a comprehensive overview (Application & Experimental Data). Downcasting and embedding in network context resulted in a novel method for fold assignment (Fold Prediction by embedding in interaction context).

APPLICATION & EXPERIMENTAL DATA

We compared the recent Y2H-data published (Schwikowski et al., 2000) with the Database of Interacting Proteins DIP (Xenarios et al., 2001) using the Structural Classification of Proteins SCOP. The Schwikowski (Y2H) data represent a comprehensive overview of all known interactions in yeast and contain 2238 interactions, of which only 37 interactions were present in DIP. DIP (dip12102000.dat) contains 3602 interactions from different sources and organisms, including yeast. As a hierarchical clustering we mapped these interactions to the structural classification of proteins, SCOP version 1.53 (Murzin et al., 1995). SCOP classes 1-3 were used in the analysis, excluding multidomain, membrane or artificial proteins.

Mapping the Interaction data

In this experiment, we used SCOP for the contraction of the interaction data. SCOP classifies protein domains into a hierarchy with different levels of similarity, which are from the most general to the most specific: Class, Fold, Superfamily, Family and Protein.

The interactions from DIP and Y2H were linked to SCOP by assigning them to those proteins in SCOP which resulted in the best e-values using BLAST (Altschul et al., 1997). First we validated that $t = 10^{-5}$ is a reasonable cutoff for the assignment by comparing the entries in SCOP against themselves (data not shown). Subsequently, the assignment of all nodes in DIP and Y2H was performed using BLAST and the threshold t . An interaction was considered as fully assigned if both nodes constituting this edge could be mapped reliably to SCOP. This resulted in 667 fully assigned interactions from DIP (19 %) and 149 from Y2H (7 %).

For an additional 1252 interactions in DIP and 679 in yeast only one interaction partner could be given a structural assignment. These partially assigned edges were later used to embed the unassigned interaction partner in the context of the overall network, which yielded interaction-network-context derived structural assignments for those proteins BLAST could not assign based on sequence similarity.

There were 1683 interactions in DIP and 1410 in Y2H where both interaction partners remained unassigned,

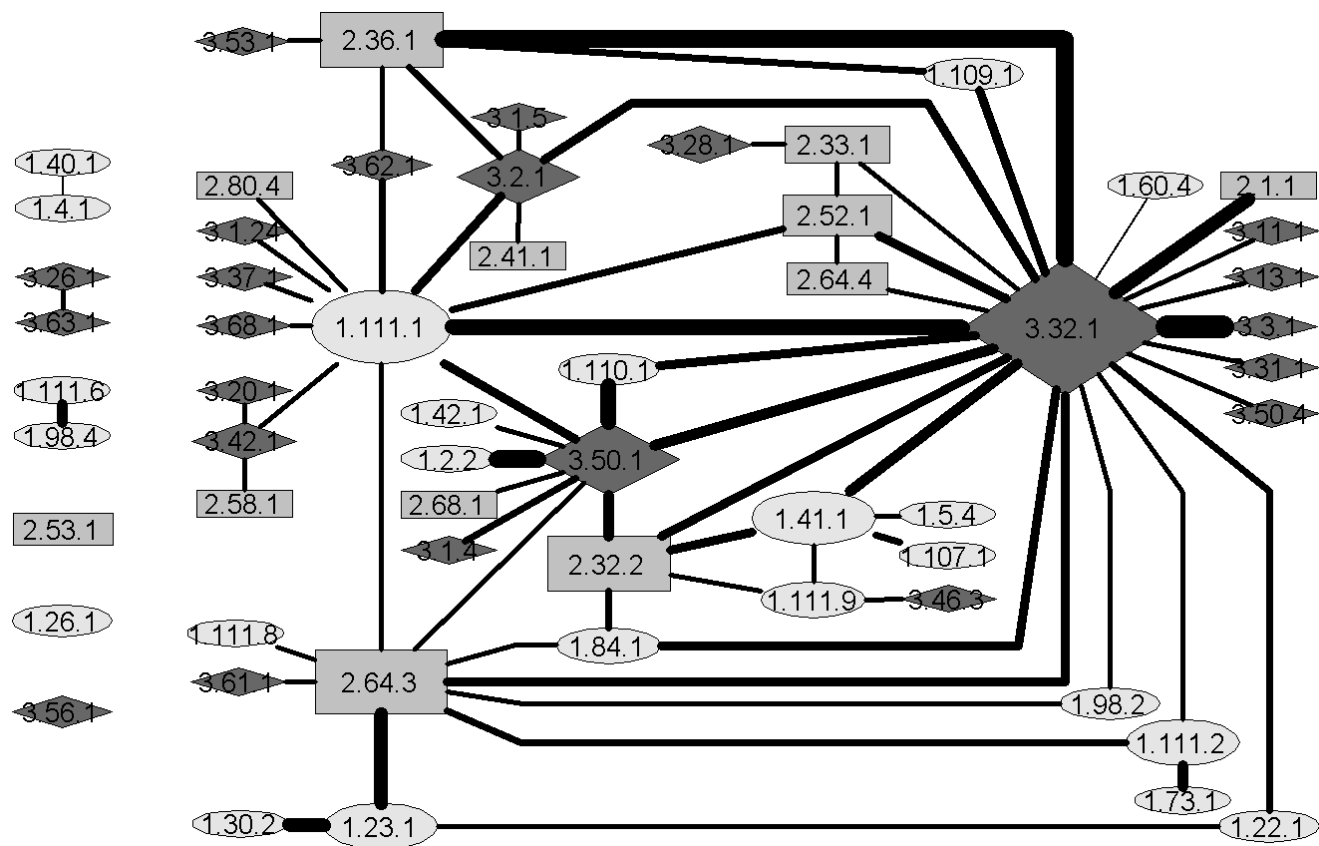


Fig. 1. Graphic representation of the intersection between the DIP and Y2H data at SCOP superfamily level.

which involved 47 % of all the nodes in DIP and 63 % of all the nodes in Y2H.

Overlaying DIP and Y2H

The SCOP superfamily (SF) level represents an unambiguous evolutionary relationship of its members. At the SF level proteins often show clear functional homology and this was chosen as the level of abstraction in our analysis. We used the upcasted interaction networks from both sources (DIP and Y2H) at superfamily level and then generated the intersection between the two sets (Figure 1).

An edge $e = (A, B)$ in the graph was created if there are at least two edges, one $d = (e, f)$ from DIP and another $y = (x, z)$ from Y2H given that e, x belong to SF A while f, z belong to SF B . Then this edge was weighted with the number of underlying interactions in both sources. So each edge in the resulting graph has at least 1 underlying interaction from each source, and therefore a weight from 2 up to 19 (denoting the number of experimental interactions that induced this edge in the graph). The weight is represented by the width of each

edge in Figure 1.

Note that for all nodes in the graph there are self-interactions linking the node with itself which have been omitted in the graphic representation for the sake of clarity.

Discussion of the graph architecture

Figure 1 shows the essential structure of all the interaction networks upcasted to SCOP we have seen so far (Park et al., 2001). Scale-free networks are commonly observed in biology, and the protein-protein interaction networks are no exception (Albert et al. 2000; Jeong et al., 2000). Obviously the scale free property is retained in the contraction, as there are few hubs with high connectivity keeping the network together while most nodes have a relatively low connectivity. The most highly connected nodes (top down from the graph) are listed in Table 1, where the SCOP classes are '1' - all alpha proteins (containing 128 folds), '2' - all beta proteins (87) and '3' - alpha and beta proteins (93). The breakdown of the 12 most populated hubs reveals 6 superfamilies involved in signal transduction or immunity, 4 widespread catalytic domains and 2 compo-

Table 1. Hubs in the protein-protein interaction graph.

SCOP SF	Description	Cellular function
1.2.2	Chaperone J-domain	Protein folding
1.41.1	EF-hand, calcium binding	Signal transduction
1.110.1	Ras GEF	Signal transduction
1.111.1	ARM repeat	Signal transduction
2.1.1	Immunoglobulin	Immune response
2.32.2	SH3-domain	Signal transduction
2.36.1	Sm motif of small nuclear ribonucleoproteins, SNRNP	Spliceosome
2.64.3	Trp-Asp repeat (WD-repeat)	Signal transduction
3.2.1	NAD(P)-binding Rossmann-fold domains	Enzymes
3.3.1	FAD/NAD(P)-binding domain	Enzymes
3.32.1	P-loop containing nucleotide triphosphate hydrolases	Enzymes
3.50.1	Actin-like ATPase domain	Enzyme domain found in metabolic, heat shock and cytoskeletal proteins

Table 2. Graph properties at different levels of abstraction.

DIP SCOP-level	$ E $	$ V $	$\langle k \rangle$	Lambda
Fold	303	116	2.61	0
Superfamily	341	148	2.30	0
Family	383	195	1.96	8
Protein	466	319	1.46	34
Y2H SCOP-level	$ E $	$ V $	$\langle k \rangle$	Lambda
Fold	78	59	1.32	0
Superfamily	83	69	1.20	0
Family	88	87	1.01	2
Protein	114	126	0.90	18

nents involved in the assembly of multisubunit molecular machines (spliceosome and chaperones). The strong representation of signaling pathways is natural since these are based on specific protein-protein interactions (which may be localized to adaptor modules like the SH3-domain). The enzyme domains may have been co-opted in evolution to power proteins involved in a variety of cellular functions. For example, the actin-like ATPase domain is part of actin (cytoskeleton), several sugar kinases (metabolism) and heat shock protein 70 (Table 1).

It is not clear at the moment, whether the catalytic domains (e.g. 3.32.1 assigned to SCOP) are interacting with other proteins or whether the experimentally observed interaction involves multidomain proteins which have a specific 'interaction domain' in addition to the domain that we were able to assign to SCOP. We plan to clarify the domain issue in future work.

Upcasting to higher levels of abstraction

Table 2 shows how the resulting graphs behave under upcasting to higher levels of abstraction. $|E|$ and $|V|$ denote the number of edges and nodes, respectively. The average connectivity $\langle k \rangle = |E|/|V|$ is a measure of the density of the graph at that particular level. The lambda-value is a measure developed to evaluate the results of graph clustering algorithms (see (Stein, 1999) for a complete description). The lambda value for a given cluster is defined as the minimum number of edges that have to be removed from this cluster to make the cluster internally unconnected, multiplied by the number of nodes in this cluster. For example, a cluster containing n nodes and being internally connected has a lambda value of at least n . In this table, lambda is the sum over the lambda-values of all clusters in the graph. A total lambda value of zero means that all clusters are internally unconnected.

Surprisingly $\lambda/|V|$ is far smaller than the density $\langle k \rangle$ of the graph. This means that the clusters induced by SCOP are more densely connected externally than are the nodes contracted within each cluster. The observation that the clusters are less dense than the connectivity structure H is also confirmed in Table 2 by a much greater decrease of $|V|$ than $|E|$ when moving to higher levels of abstraction. This illustrates that most of the interactions are retained even on higher levels of abstraction (using SCOP), which is a desirable result.

Direct comparison of the two sources

As mentioned before, the procedure described above by using a given hierarchy to contract an interaction network is not the only possible way to compare interaction data from different sources. The comparison of the Y2H and DIP resulted in virtually identical graphs using two routes of upcasting: (1) map proteins in Y2H to DIP on the sequence level, then upcast equivalent interactions

using SCOP; (2) upcast proteins in either Y2H or DIP individually using SCOP, then identify equivalent interactions from their SCOP labels. Although this seems obvious given that BLAST does fairly well on detecting homology and homology is a transitive function, this is a cross-validation of the resulting overlap. That both methods generate fairly identical results means that using SCOP as a an identity-function is equivalent to using sequence homology (in this case BLAST) for integrating interaction networks.

The important practical implication is that this allows the strategy to assign each new genome and its interactome to SCOP and then subsequently compare the resulting interaction graphs within SCOP rather than comparing all genomes crosswise. Obviously, since the resulting upcasted graphs are smaller, a comparison within SCOP is much faster and easier to compute.

FOLD PREDICTION BY EMBEDDING IN INTERACTION CONTEXT

As a second application, we implemented an algorithm to embed a given node into the set of all interactions from DIP and Y2H in order to generate a fold assignment with SCOP. For this embedding and subsequent fold assignment, only the interactions known for this node were used together with the available SCOP assignment of their direct interacting partners.

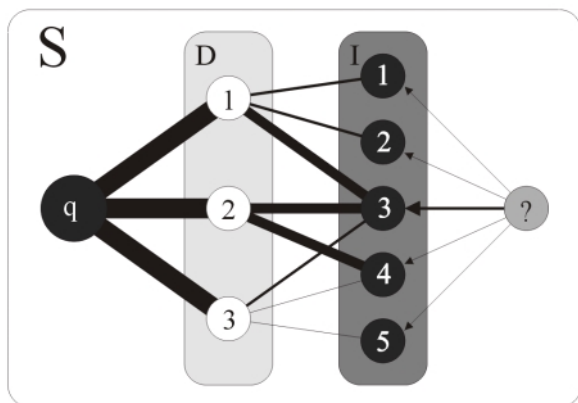


Fig. 2. Illustration of the fold assignment algorithm, given a query q based with its known direct interaction partners $D[1-3]$. The algorithmic problem is the identification of the node within the set of indirect neighbors $I[1-5]$ which has the most similar interaction pattern to q . The query q is then given the same fold class label as the best hit in I .

Table 3. Evaluation of the assignment algorithm using three different random test sets.

Set	Nodes	Assigned	Rank=1	Rank ≤ 5
A1	56	52	15	23
B1	49	36	9	12
C1	39	30	10	16
Total	144	118	28.8 %	43.2 %

Embedding Algorithm

We implemented the following embedding algorithm which is based on the idea of the computing maximum flow in a weighted graph (Figure 2).

Let's consider the complete interaction network $N = (V, E)$ which in our case consists of the induced interactions from DIP and Y2H on all levels of SCOP excluding the class level at the top of the hierarchy. The class level was omitted because there are only three classes and it helped to keep the graph at a reasonable size. All edges are weighted with the number of interactions from DIP and Y2H that are subsumed under this induced interaction.

For a given query node q , we generate a weighted subgraph S using the following method: First we initialize S by inserting q and its known interactions. The edges from q into the set of direct neighbors, D , are weighted with maximum capacity. Then for all nodes $d \in D$ the interactions derived from N are added to S , using the SCOP assignments of the nodes in D only. At this stage we consider all induced interactions at all levels of abstraction from the nodes in D . This results in the set of all nodes indirectly interacting with the query, called I . All nodes in I now represent a SCOP assignment and not an individual protein, all edges between D and I are induced interactions, each weighted with the number of experimental interactions subsumed. Given the observation that, by definition, every node in D is a direct neighbor of q as well as a direct neighbor of any node in I , all nodes in I are potential fold assignments for our query node q .

In order to work out the best assignment we have to determine the node in I where the interaction pattern is most similar to that of q . Therefore we add the node s as a source into S , connecting it to all nodes in I with edges of maximum capacity. Then, by using the working assumption that hitting an edge between D and I with a higher weight, i.e. with more experimental evidence, is a better choice, we have reduced the assignment problem down to compute the maximum flow from s as a source and q as a sink within S , using the weights as the capacities of the edges. Due to the construction of S , choosing a node in I which optimizes the flow is therefore the best known assignment for q given the underlying interaction data.

Table 4. Fold assignment predictions for hypothetical yeast proteins.

Swissprot id	Length	PDB evalue	SCOP assignment	SCOP description
YEX6_YEAST	93	1e-4	2.36.1.1	SNRNP
YHW1_YEAST	637	0.5	1.111	alpha-alpha superhelix
YES2_YEAST	125	100	3.32.1	P-loop containing NTP hydrolases
PAF1_YEAST	445	-	1.4.5.15.1	DNA-binding domain from rap30
SED4_YEAST	1065	-	3.17	flavodoxin-like
YNI6_YEAST	102	-	3.56.1.1.13	phosphoribosyltransferase
YG33_YEAST	275	-	2.52.1.2	molybdenum cofactor biosynthesis protein
YHX1_YEAST	630	-	3.32.1.13.3	P-loop containing NTP hydrolase

Obviously the maximum flow in this case is equivalent to the maximum flow across the cut between D and I . Therefore the sum of the weights of all adjacent edges for any node in I is used as the scoring function for the fold assignment of q . Since we consider all possible levels of abstraction, the resulting assignment can be from any SCOP fold, SF, family or protein level.

Evaluation

In order to assess the performance of the algorithm described above, we generated 3 random sets of testnodes for which a SCOP classification is known, each set containing roughly 50 nodes (approximately 1%) of the set of overall 4556 nodes from both sources, DIP and Y2H. Then the overall interaction data was split using these nodesets, so all edges which have at least one node in the testset were written into a set of testedges, all other edges were copied to the set of trainingedges. Finally the nodes from the test sets were used as query nodes together with their known interactions from testedges to compute a SCOP assignment for q using the set of trainingedges. For evaluation, these assignments computed with our algorithm were compared to the known SCOP assignments produced by BLAST.

For the 3 test sets, between 25-30 % of the top assignments were correct. Also, a correct assignment could be found among the top 5 scores in 43% of the test-cases (Table 3). Since all predictions made were at least on the SCOP fold level (or on the more specific SF, family or protein level), and there are a total of 308 different folds within the SCOP classes 1-3, these predictions are well beyond random hits.

The advantage of the method described here is the ability to produce assignments for proteins where BLAST could not detect any clear homology. Since this method relies on context-information rather than homology, this approach can be characterized as a non-homology method (Marcotte, 2000). Homology is only needed to determine where the interaction partners of the query protein are

located within the overall network, but not for the query protein itself. The information required is the interaction pattern of the query protein, i.e. the (some of) the proteins the query is known to interact with. So we applied the algorithm to assign a fold classification to all the proteins from both DIP and Y2H that could not be reliably (with an evalue better than 10^{-5}) assigned to SCOP using BLAST alone. The complete list of fold assignments for previously uncharacterized proteins is available at <http://www.ebi.ac.uk/~lappe/FoldPred/>.

A sample of assignments obtained for hypothetical yeast proteins is shown in Table 4. YEX6_YEAST matches the sequence conservation pattern of SNRNPs (small nuclear ribonucleoproteins) and is clearly a remote family member. YHW1_YEAST has weak sequence similarities to cytoskeletal assembly proteins, and structural proteins are often helical so the alpha-alpha superhelix fold assignment is not an unreasonable prediction. Both YES2_YEAST and YHX1_YEAST hit the P-loop superfamily, but there is no identifiable sequence similarity to support a fold prediction. The target SCOP class is also extremely large, so we think this hit is spurious. The assignment of PAF1_YEAST is a possible discovery, which is supported by functional similarities. PAF1_YEAST is involved in transcriptional regulation, and the SCOP assignment is to a DNA-binding domain in the “winged helix” superfamily.

DISCUSSION

In summary, we have introduced the concepts of induced interactions, level of abstraction, upcasting and downcasting.

The concept of upcasting is similar to ‘information transfer by homology’ (Sander & Schneider, 1991), which is routinely used in functional annotation and structure assignment to genomes and in metabolic reconstruction (Overbeek et al. 2000; Teichmann et al., 2000). We have here phrased this principle in a formal framework which applies to any classification (not limited to ho-

mology) and any source of interaction information. In particular, we note that SCOP is one of several structural classifications available that could be applied here. However, CATH (Orengo et al., 1997) and the FSSP (<http://www.ebi.ac.uk/dali/fssp/>), agree well for most of the protein folds (Hadley & Jones, 1999), and the differences in these classifications should not affect the resulting overview qualitatively.

The application to fold assignment seems promising, but clearly is still lacking a statistical model to evaluate the predictions made. The scoring function is quite simple and straightforward, and there is still room to improve the assignment procedure by improving the scoring function.

We are not aware of any other prediction method that uses interaction information for fold assignment, although gene neighborhoods have been used to assign function (Huynen et al., 2000). Since the underlying ideas described in the framework are invariant to the source of interaction information as well as to the type of classification used, it can be applied beyond structural classifications. For example, to apply the described method to the prediction of subcellular localization or co-expression means employing a different form of clustering and classification that encodes such knowledge just as SCOP does encode knowledge about protein folds.

The concept of downcasting could be used to predict novel interactions between individual proteins, but this is crucially dependent on employing refined classifications, or efficient filters, that preserve the property of interaction. In other words, the development of descriptors for protein-protein-interfaces (Goh et al., 2000) would complement any non-homology method for predicting interactions, e.g. from gene fusion events (Enright et al., 1999). These descriptors of protein-protein interactions should be preferably designed to work on a sequence (like motifs) and not only on a structural level (like surface patches). Having such descriptors in place will enable genome-wide prediction of protein-interaction networks and clustering of proteins by interface similarity in functional space.

ACKNOWLEDGEMENTS

We thank Sven Meyer zu Eissen for providing the lambda values in Table 2, and Sabine Dietmann, Andreas Heger, Matthieu Louis, Sarah Teichmann and Caleb Webber for insightful discussions.

REFERENCES

- Albert, R., H. Jeong, et al. (2000) Error and attack tolerance of complex networks. *Nature*, **406**,378-82.
- Altschul, S. F., T. L. Madden, et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**,3389-402.
- Eisenberg, D. M., E.M. Xenarios, I. Yeates,T.O. (2000) Protein Function in the Post-genomic era. *Nature*, **405**, 823-826.
- Enright, A. J., I. Iliopoulos, et al. (1999) Protein interaction maps for complete genomes based on gene fusion events *Nature*, **402**, 86-90.
- Goh, C. S., A. A. Bogan, et al. (2000) Co-evolution of proteins with their interaction partners. *J. Mol. Biol.*, **299**, 283-93.
- Hadley, C. and D. T. Jones (1999) A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure Fold Des.*, **7**, 1099-112.
- Huynen, M., B. Snel, et al. (2000) Predicting protein function by genomic context: quantitative evaluation and qualitative inferences. *Genome Res.*, **10**, 1204-10.
- Jeong, H., B. Tombor, et al. (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651-4.
- Marcotte, E. M. (2000) Computational genetics: finding protein function by nonhomology methods. *Curr. Opin. Struct. Biol.*, **10**, 359-65.
- Mayer, M. L. and P. Hieter (2000) Protein networks-built by association. *Nat. Biotechnol.*, **18**, 1242-3.
- Murzin, A. G., S. E. Brenner, et al. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol.Biol.*, **247**, 536-40.
- Orengo, C. A., A. D. Michie, et al. (1997) CATH—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093-108.
- Overbeek, R., N. Larsen, et al. (2000) WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic Acids Res.*, **28**, 123-5.
- Park, J., M. Lappe, et al. (2001) Mapping Protein Family Interactions: Intramolecular and Intermolecular Protein Family Interaction Repertoires in the PDB and Yeast. *J. Mol. Biol.*, **307**, 929-938.
- Sander, C. and R. Schneider (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, **9**,56-68.
- Schwikowski, B., P. Uetz, et al. (2000) A network of protein-protein interactions in yeast. *Nat. Biotechnol.*, **18**, 1257-61.
- Stein, B., Niggemann, O. (1999) On the nature of structure and its identification. *25th International Workshop on Graph Theoretic Concepts in Computer Science*, **WG9**, Springer Verlag.
- Tatusov, R. L., E. V. Koonin, et al. (1997) A genomic perspective on protein families. *Science*, **278**, 631-7.
- Teichmann, S. A., C. Chothia, et al. (2000) Fast assignment of protein structures to sequences using the intermediate sequence library PDB-ISL. *Bioinformatics*, **16**, 117-24.
- Xenarios, I., E. Fernandez, et al. (2001) DIP: The Database of Interacting Proteins: 2001 update. *Nucleic Acids Res.*, **29**, 239-41.