InterDB, a Prediction-Oriented Protein Interaction Database for *C. elegans*

Nicolas Thierry-Mieg and Laurent Trilling

Laboratoire LSR-IMAG,
38402, Saint-Martin-d'Hères cedex, France
{Nicolas.Thierry-Mieg,Laurent.Trilling}@imag.fr

Abstract. Protein-protein interactions are critical to many biological processes, extending from the formation of cellular macromolecular structures and enzymatic complexes to the regulation of signal transduction pathways. With the availability of complete genome sequences, several groups have begun large-scale identification and characterization of such interactions, relying mostly on high-throughput two-hybrid systems. We collaborate with one such group, led by Marc Vidal, whose aim is the construction of a protein-protein interaction map for *C. elegans*. In this paper we first describe WISTdb, a database designed to store the interaction data generated in Marc Vidal's laboratory. We then describe InterDB, a multi-organism prediction-oriented database of protein-protein interactions. We finally discuss our current approaches, based on inductive logic programming and on a data mining technique, for extracting predictive rules from the collected data.

1 The Biological Problem: Protein-Protein Interactions

Protein-protein interactions are critical to many biological processes, extending from the formation of cellular macromolecular structures and enzymatic complexes to the regulation of signal transduction pathways.

With the availability of complete genome sequences, several groups have begun large-scale identification and characterization of such interactions [11], [22], [25]. These groups rely mostly on high-throughput two-hybrid systems [23]. Although such approaches significantly increase the rate at which interaction data is produced, they will require several years to produce full interaction maps for modest-sized organisms, whereas the "working draft" of the human genome has been available since June 2000. It is therefore enticing and promising to develop computational methods that could predict protein-protein interactions, be it in a rough and approximate manner. Ideally, the data produced by those high-throughput projects could suffice to develop prediction algorithms that could then be applied to genome sequence as fast as it is being released. More reasonably, the high-throughput projects themselves could benefit from predictions to speed up the discovery of interesting protein-protein interactions (see part 4).

In this paper we concentrate on a preliminary step to study protein-protein interactions in *Caenorhabditis elegans*. *C. elegans* is the first multi-cellular organism whose genome has been completely sequenced [5], as well as being a choice

O. Gascuel, M.-F. Sagot (Eds.): JOBIM 2000, LNCS 2066, pp. 135–146, 2001.

model organism for many functional genomics projects, from cDNA microarrays [9] to systematic knock-outs [10] and protein-protein interaction mapping. It is also a convenient model organism for classical genetic studies [24].

We first describe our efforts to set up WISTdb (Worm Interaction Sequence Tag database) [25], a database which gives access to interactions in *C. elegans*. These interactions are the result of the *C. elegans* interaction mapping project led by Marc Vidal at the Dana Farber Cancer Institute, Harvard Medical School, Boston, Massachusetts. Our goal consisted in setting up the informatics platform to annotate and store the interactions, and make them freely available through Internet. This has been implemented in the form of a database using the ACeDB database management system [21] and the AceBrowser [20] interface.

We then describe InterDB, a prediction-oriented database of protein-protein interactions, which we have built. The goal is to have access in a homogeneous way to as many known protein interactions present in available databases as possible, with the aim of predicting interactions in *C. elegans*.

We finally explain our current approaches to using this data. The aim is to extract rules that could explain the observed interactions of InterDB, and perhaps generalize to other unknown interactions. We report on our efforts at using an inductive logic programming technique, namely the Progol system [16], and another data mining technique based on association rules [2], [14].

2 The WISTdb Database

As said before, Marc Vidal's group at MGH is working on producing a protein interaction map for *C. elegans*, based on a high-throughput two-hybrid system. WISTdb is a database designed to store the data produced by this project. It is based on ACeDB, an object-oriented database management system developed initially to manage and distribute *C. elegans* genetic and genomic data. It also uses the Acembly sequence assembly and edition package [1], which functions on top of ACeDB. Both ACeDB and Acembly are freely available over the Internet. Since all the data currently available for *C. elegans* is distributed in the ACeDB format, the choice was a natural one. We first summarize the main ideas behind two-hybrid systems, then describe the database schema adopted for WISTdb and briefly discuss the algorithms involved.

The goal of conducting a two-hybrid experiment, or screen, is to identify proteins that physically interact with a given protein, called the bait. This bait can be expressed as a hybrid protein, fused to a specific DNA binding domain (coming from the yeast GAL4 transcription factor). Using a library of plasmids capable of expressing potential interactors fused to the GAL4 transcriptional activation domain, a custom yeast strain is co-transformed with two plasmids: the bait hybrid and one random plasmid from the library. The specifically engineered yeast cell is designed such that under certain conditions, its survival depends on the bait interacting with the unknown protein, thereby reconstituting the GAL4 activity. At this point, surviving yeast colonies contain a plasmid in which the cDNA insert codes for an interactor of the bait. Those interactors are called

fishes. To identify them, they are amplified by PCR, then sequenced. The data produced is therefore a set of ABI automatic sequencer traces, for each bait screened. Our task is to align those traces on the *C. elegans* genomic sequence, so as to identify the fishes, and to store the newly discovered interactions in a database along with all relevant information. The main conceptual difficulties lay in the definition of a schema for the data. This is discussed below.

The data schema is derived from the schema distributed with the Acembly package. This software system is designed to handle ABI trace files and perform sequence assembly. It is well suited to the alignment of cDNA sequences on genomic sequence, and includes a versatile graphical interface to visualize and edit the traces. The new schema has been stripped of unnecessary classes and attributes, and enriched with the new classes IST and ISTScreen to describe the interactions found.

The main remaining standard classes are the following: locus, sequence, cdna_clone, and transcribed_gene. The locus class represents genetic loci, including gene names and other genetic information, as well as links to relevant sequence objects when available. The sequence class contains all nucleic acid sequences, be they genomic cosmids, expressed sequence tags (ESTs), predicted open reading frames, or ABI sequence traces. Cdna_clone objects store information on the clones from which the ABI traces come. Finally, when the traces are aligned on genomic sequence, they can be clustered into overlapping subsets of traces. The transcribed_gene objects are created by the system to reconstruct the genes corresponding to these clusters.

The new classes can be described as follows. An ISTScreen object is created for every gene used as a bait in a two-hybrid screen. It contains links to every IST object that represents an interaction uncovered by this screen. It also contains links to the gene (genetic locus) and sequence (physical locus) of the bait. An IST object is basically a link between two proteins that interact in the context of the two-hybrid system, a bait and a fish. It contains additional information such as the observed strength of the interaction, in terms of what two-hybrid phenotypes are observed. Both bait and fish may be referred to by gene name and/or sequence identifier. In fact, given a bait and a transcribed_gene, an IST linking them is generated if the transcribed_gene's construction relies on an ABI trace corresponding to a cdna_clone which scored positive in the two-hybrid screen with that bait.

The current version of WISTdb stores interaction data for 22 baits, therefore containing 22 ISTScreen objects. It also contains 1195 cdna_clone objects, which represent 117 different transcribed_gene objects, and correspond to 146 interactions (IST objects).

3 Construction of InterDB from Current Databases

The first step in studying protein-protein interactions is the construction of a database of such interactions. The main difficulties encountered are the scarcity

of available data, and the problem of constructing an integrated database from several independent heterogeneous databases, as described below.

First, computationally useful protein interaction data is hard to find. Still today, biologists studying particular proteins search and identify interacting partners for their protein of interest. That information is eventually published, sometimes inconspicuously, along with other information on the protein. Reading biology papers can be a daunting task for non-specialists like ourselves, let alone reading thousands of them looking for specific data that rarely stands out. One approach to this problem is to apply natural language processing techniques, in order to extract protein-protein interaction information from the scientific literature. For example A. Valencia and colleagues, from Madrid University, have initiated such studies based on Medline abstracts (personal communication). But although this strategy appears promising, it can only give *predictions* of protein-protein interactions, since the natural language processing involved is not completely accurate. The only reasonable scenario for obtaining experimental data lay in either finding protein interaction databases compiled by third parties, or teaming up with functional genomics projects that produce such data. As said before, a collaboration was set up with the Vidal laboratory in Boston, which gives us access to the interactions that they detect. But more data was desirable. The following databases containing information on protein-protein interactions were found.

The DIP database (Database of Interacting Proteins, [12]) is a collection of interactions in a variety of organisms. Until recently, it used exclusively the PIR (Protein Information Resource, [26]) unique identifier to identify interacting proteins. The current release also provides SwissProt [3] identifiers when possible. DIP is freely available and downloadable, and contains around 2290 interactions at the time of writing.

Another source of interaction data is YPD (Yeast Protein Database, [27]). YPD is a general database on *Saccharomyces cerevisiae* proteins. It is the result of manual curation and annotation based on a review of the literature. YPD is a proprietary database, although access is freely granted to academic users. A lot of functional information is present, including protein-protein interactions, but it is in the form of free text in English, and scriptable access to YPD is forbidden. This would preclude using it in our context, but an agreement was negotiated with Proteome Inc., leading to our having access to a table containing all YPD protein-protein interactions. The table comprises 1115 interactions, and interacting partners are referred to by gene name.

Finally, FlyNets [19] contains to date 53 protein-protein interactions in the fly *D. melanogaster*. This database was considered but not used.

The second problem encountered is a classical one: starting with several heterogeneous databases, construct a single database integrating the knowledge stored in the initial ones. In our case, a unique framework was needed to identify proteins. The protein databases SwissProt and TrEMBL [3] were chosen. They are available as a non-redundant flat database, and respect a reasonably

tight syntax. They contain links to corresponding PIR identifiers, as well as gene names.

For each source of protein-protein interaction data, a Perl script was written to parse the data files, and ace files were generated for all interactions in which both partners could be identified in SwissProt or TrEMBL. This guarantees the coherence of InterDB, but led to discarding some of the available interactions. Out of 1115 YPD interactions, 247 (22%) were discarded because at least one interactor could not be identified in SwissProt or TrEMBL, by the gene name given in YPD. Similarly, 1010 of the 2290 DIP interactions (44%) were discarded because they involved at least one protein whose PIR identifier was not referenced in SwissProt or TrEMBL.

A new database, called InterDB, was built to store the interaction data collected. It uses ACeDB as a database management system. The schema is designed to fulfill two main goals:

- the quick construction of the database from new releases of the source databases,
- the optimization of queries necessary to construct training sets of interactions, and to predict protein-protein interactions using predictive rules.

The schema contains the three main classes Protein, Interaction and Predictive_rule. Protein objects correspond to the SwissProt and TrEMBL entries. The information contained in the following SwissProt fields is stored: identification, including gene names and organisms concerned; database cross-references to PIR, ProSite and Pfam; keywords; and Sequence. Interaction objects are basically links between two protein objects. Predictive_rule objects are used to store predictive rules, as generated by the approaches described below.

InterDB contains to date 307199 protein objects, and 2245 interaction objects involving 1891 proteins. It should be noted that although only 1891 proteins are involved in interactions, InterDB must store all 307199 proteins from SwissProt and TrEMBL in order to identify interacting partners. 5% of the interactions are between *C. elegans* proteins, 75% between *S. cerevisiae* proteins, 10% between H. *sapiens* proteins, and the remaining 10% are spread over 40 various organisms. 45% of the 1891 proteins involved in at least one interaction are actually involved in two or more interactions.

4 Protein Interaction Prediction

As stated before, protein-protein interactions are fundamental to a wide range of biological processes, and several large-scale projects are under way to identify them experimentally. Yet the prediction of such interactions by computational methods remains a highly attractive goal, for reasons evoked earlier. Even if the predictions are not reliable enough to be useful to the final user, i.e. the biologist, they could still prove valuable, in the sense that they could guide the high-throughput projects to speed up the discovery of interesting protein-protein interactions.

Indeed, two-hybrid experiments can be performed in two different manners: the first approach is to screen against a library, as described in part 2; the second is to clone some genes into both DNA binding domain and transcriptional activation domain vectors, and to test all the cloned genes against one another. The main advantage of the first strategy is that each cloned gene can be tested and yield interesting results. Its main drawback is that every fished clone must be sequenced. The second approach requires that many genes be cloned beforehand, but the expensive sequencing step is avoided since the scientist knows what protein-protein interactions he is testing. In this context, cloning the genes in a favorable order can yield positive interactions rapidly. For example, if the aim is to map interactions concerning proteins involved in DNA repair, a possible strategy is to clone the fifty or so genes known to be involved in this process, and another few hundred genes suspected, or predicted, to interact with them.

It seems that bioinformatics has become interested in the question of predicting protein-protein interactions fairly recently. To our knowledge, two groups have published work in this direction [17]: Marcotte et al. [12] and independently Enright et al. [7] have developed methods to predict protein-protein interactions. They both rely on the hypothesis that when two proteins A and B are homologous to (a part of) a third protein in another organism, but are not homologous to each other, then they interact (the "fused domain" approach). Marcotte et al. also proposed a multiparadigm algorithm to predict functional links between *S. cerevisiae* proteins [13], which actually uses three prediction engines and two sources of experimental data. The first prediction engine is the fused domain algorithm discussed above. The second links proteins with related phylogenetic profiles [18], e.g. two proteins that have homologs in approximately the same subsets of genomes are predicted to interact. Finally, the third engine links proteins whose mRNA levels are correlated across various microarray experiments in *S. cerevisiae* [6]. This third method is not generalizable to other organisms without access to genome-wide expression data. However, the first two methods can be applied to any sequenced organism, and aim at predicting physical protein-protein interactions. But they are hypothesis-driven, meaning that their starting point is a biological hypothesis, which is implemented then validated.

5 An ILP Approach

Our method is different, in that we seek to discover rules that could explain protein-protein interactions, but don't have strong preconceptions as to what these rules should be. However, we do know that we need a formal language to express and manipulate the relevant biological background knowledge. The chosen formalism is first-order logic. The proposed predicates fall into two main categories: first, predicates that characterize individual proteins, specifically predicates that express the presence of a Pfam [4] or ProSite [8] domain in the protein considered, or its association with a keyword in SwissProt; second, predicates that express relationships between proteins or functional domains, for example predicates asserting that two proteins interact, or are homologs. We considered

including predicates of the first category to use raw structural data from the PDB. Indeed, the information content of a 3D structure is clearly much higher than that of a Pfam or ProSite entry. In practice though, this idea is not implementable due to the scarcity of structural data concerning *C. elegans* (there are only 15 structures for partial or complete *C. elegans* proteins in the PDB as of November 2000). This language is used in the framework of inductive logic programming [15], which is an automated learning method inspired by logic programming and machine learning. More specifically, the powerful inductive logic programming system Progol [16] is used.

Progol generalizes a set of examples, i.e. positive instances of the predicate interaction, by generating Horn clauses from which these examples can be deduced. However, the induced Horn clauses can be specified to abide by user-defined rules, most notably the so-called mode declarations. Mode declarations permit to restrict the form that induced rules may take, for example by specifying whether the variables that appear in the atoms should be pre-bound or not. Allowing too many unbound variables, i.e. output variables, greatly increases the search space of inducible rules. Typically we decided, by using modes, to restrict the induced rules to the form:

```
interaction(P,Q):-
descriptor(P,D1)∧descriptor(P,D2)∧...∧descriptor(Q,D3)∧...,
```

where `descriptor(P,D)` is true if protein P is described by descriptor D, i.e. P contains Pfam or ProSite domain D, or is described by keyword D in SwissProt. Note that P and Q are always bound when they appear in the `descriptor` predicate: they were bound in `interaction(P,Q)`.

We tried several experiments with Progol. Unfortunately it appears that this approach is not suitable for such large amounts of data, both in terms of number of interactions and number of attributes. In fact, we had to restrict the size of our training sets to a maximum of 80 positive examples to avoid running out of memory and to stay within reasonable runtimes. Also, this method is particularly well adapted for dealing with highly structured and abundant background knowledge. Alas, the biological knowledge currently included in InterDB is mostly flat. For these reasons, we shifted towards a data mining technique supposedly better adapted to our data, namely association rules.

6 An Association Rule Approach

The idea behind the association rule data mining technique [2] [14] is the following. Given a boolean matrix where each line is a transaction and each column is an item, the goal is to find sets of items which are frequently present in the same transactions. From these *frequent itemsets*, one can then derive rules that link items. For example, if A and B are two items, and {A,B} is a frequent itemset, then A=¿B is derived, provided that its confidence (the ratio of the frequency of {A,B} over the frequency of {A}) is high enough. To apply this technique to our problem, we proceeded as follows.

The first task was to find a way of representing our data in an appropriate boolean matrix. The proteins corresponding to the 2245 interactions of InterDB are described by 497 Pfam domains, 406 ProSite domains, and 357 keywords, adding up to a total of 1260 descriptors. We therefore built a matrix comprising 2520 (1260×2) columns and 4490 (2245×2) lines. Each line represents an interaction, and for each line the first 1260 columns represent the presence of descriptors in the first protein, while the last 1260 columns correspond to the second protein. Since this introduces a dissymmetry not present in the inherently symmetrical "interaction" relation, we chose to enter interactions twice, once in each orientation, hence the 4490 lines. This choice means that for every frequent itemset, there is a dual itemset that appears with exactly the same frequency, and actually represents the same link between descriptors.

We then used a program, which implements the classic Apriori algorithm, to find frequent itemsets in this matrix. The frequency cutoff was set at 0.5%, meaning that an itemset had to be present in at least 0.5% of the lines to be considered frequent. This step produced 98391 frequent itemsets, along with their respective observed frequencies.

We finally applied a series of custom filters, designed to extract significant frequent itemsets from this list.

1. The first filter discards itemsets that concern only one protein. These itemsets actually correspond to linked descriptors within a single protein. For example, the well-known SH2 domain is represented by three different descriptors: one Pfam domain, one ProSite domain, and one SwissProt keyword. These three descriptors will therefore naturally form a frequent itemset. Although such itemsets are not deprived of meaning, they are not useful to predict interactions.
2. A second filter discards itemsets that contain specific user-specified descriptors. Typically, we don't want to consider itemsets containing the SwissProt keyword "hypothetical protein". Indeed, this keyword is obviously irrelevant to protein-protein interactions. Conceptually, these "bad" descriptors, which are actually mostly keywords, could have been eliminated before running the Apriori algorithm. But checking every descriptor beforehand would have been much more time-consuming than just looking at the descriptors that occur frequently and checking that they make sense with regards to protein-protein interactions. In a first approximation, we introduced an upper limit on the number of proteins described by each keyword to consider the keyword valid.
3. A third filter assigns a significance score to each itemset. The itemsets whose score is below a user-specified threshold are discarded. This score is defined as follows. Consider a frequent itemset I occurring with frequency F. We can write I=(D1,. . . ,Dn,D'1,. . . ,D'p), where D1. . . Dn are descriptors for protein 1 and D'1. . . D'p are descriptors for protein 2. The itemsets (D1,. . . ,Dn) and (D'1,. . . ,D'p) are also frequent, although they have been filtered out by step 1, and occur with frequencies F1 and F2 respectively. Supposing that these

two itemsets are independent, I is expected with a frequency F1xF2. We define the score for I as F/(F1xF2).

4. A final filter is then applied, in order to favor large itemsets vis-a-vis their subsets when the score penalty is not too heavy. Specifically, whenever two itemsets I1 and I2, whose scores are S1 and S2 respectively, are such that I1⊂I2, I1 is discarded if S1/S2 is smaller than a user-specified value.

To summarize, these filters generate a set of hopefully significant frequent itemsets, parameterized by three user-defined variables: a list of "bad" descriptors, a score cutoff, and a set to subset score ratio cutoff. Note that a frequent itemset (D1,...,Dn,D'1,...,D'p) can be interpreted as the rule:

interaction(P,Q) :-
descriptor(P,D1)∧...∧descriptor(P,Dn)∧descriptor(Q,D'1)∧...
∧descriptor(Q,D'p).

Based on this idea, a set of rules is generated for every triplet of parameter values. In practice, the filters are implemented in Perl, and a range of promising values has been determined for each parameter. The filters have been run for every combination of values in these ranges.

The next step involves validating the rules, and finding the optimal values for each parameter. A new set of experimentally determined protein-protein interactions has been produced recently by Anne Davy from CRBM, Montpellier in collaboration with Marc Vidal's laboratory. This test set TS contains 103 interactions, involving 81 proteins which play a role in the *C. elegans* proteasome. Since these interactions have not been used to produce current predictive models, they constitute a nice test set for these models.

In practice, the sets of rules are stored in InterDB and a Perl program has been developed to apply them to any input protein. The result is a list of potential interactors for that protein.

105 predictive models have been generated, using a promising range of values for each parameter. The upper limit on the number of proteins described by valid keywords has been set to 50, 100 and infinite (use all descriptors). The score cutoff has been set to 1, 2, 3, 5, 8, 10 and 50. The set to subset score ratio cutoff has been set to 1, 2, 3, 5 and infinite (keep all subsets). Using the most stringent values for these parameters, i.e. 50 for the upper limit, 50 for the score cutoff and 1 for the set to subset score ratio cutoff, we obtained 385 predictive rules. Using the most permissive values, i.e. infinite, 1 and infinite respectively, we obtained 83469 predictive rules.

We applied a representative subset of the predictive models to each TS protein, to obtain predicted interactions involving it. No interaction from TS was predicted successfully. This can be explained by the following observation: only 3 interactions from TS could possibly have been predicted by the most permissive model. We mean by this that those 3 interactions are the only ones in which each partner is described by at least one descriptor present in at least one predictive rule. We can propose two possible explanations for this. First, the initial frequency cutoff used in the Apriori algorithm seems too high, as only

79 descriptors out of 1260 are present in the predictive rules. Note that this is independent of the filters, since the most permissive model uses values which completely bypass filters 2 and 4, and reduce filter 3 to eliminating blatantly bad rules. Second, the information content of InterDB could be insufficient to produce pertinent rules for proteins from the proteasome.

7 Conclusion

Protein-protein interaction prediction is a difficult task, due to several reasons. First, it seems that there are not enough biological experiments to build training sets with enough coverage. Second, as always in bioinformatics, the data is never completely reliable. Third, counter-examples are not available, mostly due to the nature of the problem. This is particularly the case in a high-throughput setting, where it is vital to keep the generation of false positives as low as possible, therefore tolerating a higher rate of false negatives.

Three aspects of our work have been presented. We have first described the development of WISTdb, a platform designed to generate, store, annotate and make available the *C. elegans* protein-protein interaction data generated by the Vidal laboratory. Second, we have described InterDB, a prediction-oriented multi-organism protein interaction database. Finally, we have reported on our attempts to generate predictive rules for protein-protein interactions, using the Progol inductive logic programming system and an association rule data mining technique. We have not yet obtained satisfactory results with these approaches, perhaps for the reasons detailed above.

Work is under progress in two directions. On one hand, we plan to include richer and more structured biological knowledge in InterDB, from three sources. First, although the subcellular localization information from SwissProt is specified as free text, a close inspection reveals that most of the entries are chosen in a list of 66 localizations, which could be used as descriptors for our proteins. Second, we are replacing the Pfam and ProSite descriptors by InterPro descriptors (http://www.ebi.ac.uk/interpro/). InterPro is an integrated resource of protein families, domains and sites, and federates data from Pfam and ProSite, along with ProDom and Prints, two other databases with similar aims but different approaches. This resource will certainly provide more reliable information than the independent use of Pfam and ProSite, and also structures its entries by introducing relationships between them. Finally, we wish to incorporate data from the Gene Ontology Consortium (http://genome-www.stanford.edu/GO/), which features a hierarchical classification system for the functional annotation of proteins from Drosophila, Saccharomyces, Mus, Arabidopsis and Caenorhabditis. On the other hand, studies on fine-tuning the parameters used in the association rule approach are under way, especially by lowering the initial frequency threshold.

Acknowledgements. We wish to thank Jean-Francois Boulicaut and Baptiste Jeudy from INSA, Lyon for their help with the association rule data mining ap-

proach, and Anne Davy from CRBM-CNRS, Montpellier for sharing unpublished results concerning protein-protein interactions in the *C. elegans* proteasome.

References

1. The Acembly sequence assembly package, `http://alpha.crbm.cnrs-mop.fr/acembly/`
2. Agrawal R., Srikant R. (1994): Fast algorithms for mining association rules. *Proceedings of the 20*th *VLDB Conference*, 487-499
3. Bairoch A., Apweiler R. (1999): The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Research* **27(1)**, 49-54
4. A. Bateman, E. Birney, R. Durbin, S. Eddy, R.D. Finn, E.L. Sonnhammer(1999): Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acids Research*, 27(1), 260-262
5. The C. elegans Sequencing Consortium (1998), *Science* **282**, 2012-2018
6. M. Eisen, P. Spellman, P. Brown, D. Botstein (1998): Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* **95**, 14863-14868
7. A. Enright, I. Iliopoulos, N. Kyrpides, C. Ouzounis (1999): Protein interaction maps for complete genomes based on gene fusion events. *Nature* **402**, 86-90
8. K. Hofmann, P. Bucher, L. Falquet, A. Bairoch(1999): The PROSITE database, its status in 1999. *Nucleic Acids Research*, 27(1), 215-219
9. The Kim laboratory, `http://cmgm.stanford.edu/~kimlab`
10. The *C. elegans* Gene Knockout Consortium, `http://www.cigenomics.bc.ca/elegans/`
11. Lecrenier N., Foury F., Goffeau A. (1998): Two-hybrid systematic screening of the yeast proteome. *BioEssays*, **20**, 1-5
12. E. Marcotte, M. Pellegrini, H. Ng, D. Rice, T. Yeates, D. Eisenberg (1999): Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751-753
13. Marcotte E., Pellegrini M., Thompson M., Yeates T., Eisenberg D. (1999): A combined algorithm for genome-wide prediction of protein function. *Nature* **402**, 83-86
14. Manilla H., Toivonen H., Verkamo A. (1994): Efficient algorithms for discovering association rules. *KDD-94: AAAI Workshop on Knowledge Discovery in Databases*
15. S. Muggleton, L. De Raedt(1994): Inductive logic programming: theory and methods. *Journal of logic programming*, 19,20:629-679
16. S. Muggleton(1995): Inverse entailement and Progol. *New generation computing*, 13, 245-286
17. A. Sali (1999): Functional links between proteins. *Nature* **402**, 23-26
18. Pellegrini M., Marcotte E., Thompson M., Eisenberg D., Yeates T. (1999): Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA* **96**, 4285-4288
19. C. Sanchez, C. Lachaize, F. Janody, B. Bellon, L. Röder, J. Euzenat, F. Rechenmann, B. Jacq(1999): Grasping at molecular interactions and genetic networks in Drosophila melanogaster using FlyNets, an internet database. *Nucleic Acids Research* **27(1)**, 89-94
20. L. Stein, J. Thierry-Mieg (1999): Scriptable Access to the Caenorhabditis elegans Genome Sequence and other Acedb Databases. *Genome Research* **8(12)**:1308-1315
21. J. Thierry-Mieg, D. Thierry-Mieg, L. Stein (1999): ACEDB: The ACE database manager. In S. Letovsky (ed.): *Bioinformatics, Databases and Systems*, Kluwer Academic Publishers, 265-278

22. Uetz *et al.* (2000): A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, **403**, 623-627
23. M. Vidal, P. Legrain (1999): Yeast forward and reverse 'n'-hybrid systems. *Nucleic Acids Research* **27(4)**, 919-929
24. A. Walhout, H. Endoh, N. Thierry-Mieg, W. Wong, M. Vidal (1999): A model of elegance. *American Journal of Human Genetics* **63(4)**:955-61
25. A. Walhout, R. Sordella, X. Lu, J. Hartley, G. Temple, M. Brasch, N. Thierry-Mieg, M. Vidal (2000): Protein interaction mapping in C. elegans using proteins involved in vulval development. *Science*, **287**, 116-122
26. Winona C. Barker, John S. Garavelli, Peter B. McGarvey, Christopher R. Marzec, Bruce C. Orcutt, Geetha Y. Srinivasarao, Lai-Su L. Yeh, Robert S. Ledley, Hans-Werner Mewes, Friedhelm Pfeiffer, Akira Tsugita and Cathy Wu (1999): The PIR-International Protein Sequence Database. *Nucleic Acids Research* **27(1)**: 39-43
27. The Yeast Protein Database, `http://www.proteome.com/`