Linköping Electronic Articles in
Computer and Information Science
Vol. 5(2000): nr 31

# Logic and the Automatic Acquisition of Scientific Knowledge

Ross D. King
Andreas Karwath
Amanda Clare
Luc Dehaspe

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/2000/031/

*Published on December 21, 2000 by*
*Linköping University Electronic Press*
*581 83 Linköping, Sweden*

**Linköping Electronic Articles in**
**Computer and Information Science**
*ISSN 1401-9841*
*Series editor: Erik Sandewall*


*Typeset by the author using LaTeX*
*Formatted using* étendu *style*

**Recommended citation:**
*<Author>. <Title>. Linköping Electronic Articles in Computer and Information Science, Vol. 5(2000): nr 31.* http://www.ep.liu.se/ea/cis/2000/031/. *December 21, 2000.*

*This URL will also contain a link to the author's home page.*

**Abstract**

This paper is a manifesto. It argues that:

- Science is experiencing an unprecedented explosion in the amount of available data.
- Traditional data analysis methods cannot deal with this increased quantity of data.
- There is therefore an urgent need to automate the process of refining scientific data into scientific knowledge.
- Inductive logic programming (ILP) is the data analysis framework best suited for this task.

We describe an example of using ILP to analyse a large and complex bioinformatic database which produced unexpected and interesting scientific results. We then point a possible way forward to integrating machine learning with scientific databases to form intelligent inductive databases.

**Authors' affiliations:**

**Ross D. King, Andreas Karwath, and Amanda Clare:**
Department of Computer Science
University of Wales
Aberystwyth, Penglais, Aberystwyth
Ceridigion, SY23 3DB, Wales, U.K.

**Luc DeHaspe:**
PharmaDM
Celestijnenlaan 200A
B-3001 Brussels, Belgium

*"If we trace out what we behold and experience through the language of logic, we are doing science"*

*"The grand aim of all science [is] to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms"*

A. Einstein

# 1 Introduction

## 1.1 The need for automation of reasoning in science

Perhaps the most characteristic feature of science at the start of the third millennium is the "data explosion" (Reichart, 1999). From particle-physics to ecology, from neurology to astronomy, almost all the experimental sciences are experiencing an unprecedented increase in the amount and complexity of available data. Scientific databases with Terabytes ($10^{12}$ bytes) of data are now common place, and soon Petabyte ($10^{15}$bytes) database will be so. In some fields plans are already underway to deal with Exabytes ($10^{18}$ bytes) and Yottabytes ($10^{21}$ bytes) of data. To take a number of high profile cases:

- In physics, the new CERN Large Hadron Collider (LHC) is expected to generate ~100 Petabytes of data over its 15 year lifetime.
- In Geology, the US Geological survey is midway through photographing the entire US from the air at high resolution. This will form a ~12 Terabyte database.
- In Biology, transcriptome (microarray) data from gene knockouts of the human genome is expected to generate ~1 Petabyte of raw data - this data is also extremely complicated, as it has semantic links with many other biological databases.

These databases are made possible by increasingly sophisticated instrumentation, and by ever more powerful information technology. They represent a great achievement for science, and within them lies an abundance of new scientific knowledge. However, full analysis of such databases presents huge difficulties, and it is obvious that traditional approaches to scientific data analysis will not be sufficient. *New more automated data analysis methods are essential to refine the vast quantities of scientific data into communicable scientific knowledge.*

## 1.2 Scientific Discovery

The branch of Artificial Intelligence devoted to developing algorithms for acquiring scientific knowledge is known as "scientific discovery". Work has proceeded in this area for over thirty years and much has been achieved (e.g. Buchanan *et al.,* 1969; Langley *et al.,* 1987; Sleeman *et al.,* 1989; Gordon *et al.*, 1995; King *et al.,* 1996; Valdes-Perez, 1999). However, the field has not made a significant impact on science generally. *There have been no great scientific discoveries directly attributable to machine intelligence.*

Why do scientific discovery programs not rival scientists in the way that the chess playing programs rival chess Grand masters? We believe the answer is that chess is an isolated world where, despite the almost limitless complications possible, everything that is relevant to the game is described in the rules. In contrast, scientific problems cannot be so easily isolated from general reasoning where human excel. To be successful scientific discovery programs do not need to know about everything - "Cabbages and Kings" (Boden, 1977); but they do need access to large amounts of background knowledge about the scientific field they are applied to. Humans still have the decisive edge over scientific discovery programs because scientific knowledge is difficult to encode, and we do not have a good theory of relevance. *However, as the focus of science moves towards problems involving very large and complex datasets, the balance of advantage must shift towards machine intelligence.*

### 1.3 Inductive Logic Programming

We believe that first-order predicate logic should be the basis for representing knowledge in scientific discovery programs as it has the best understood semantics and inference mechanisms (Russell & Norvig, 1995). Propositional logic is too weak to be generally used on scientific problems: it cannot represent much of our background knowledge about scientific problems, nor can it represent scientific theories in the concise way scientists expect. Similarly, we consider that it is computationally intractable to learn scientific theories in higher-order, temporal logics etc., especially when large databases are involved (however, see Bowers *et al.*, 2000).

Machine learning and data mining methods that employ first-order predicate logic to represent background knowledge and theories come from the field of Inductive Logic Programming (ILP) (Muggleton, 1990; Muggleton, 1992; Lavrac & Dzeroski, 1994). ILP is a relatively new and highly promising branch of machine intelligence. ILP has shown its value in many scientific problems in chemistry (e.g. King, *et al.,* 1992; King, *et al.,* 1996; Dehaspe, *et al.,* 1998; Helma *et al.*, 1998; Finn *et al.*, 1998; King & Srinivasan, 1999; Dzeroski *et al.*, 1999) and molecular biology (Muggleton, *et al.,* 1992; King, *et al.,* 1994; Sternberg *et al.,* 1994; Turcotte *et al.,* 2000) where it has found solutions not accessible to standard statistical, neural network, or genetic algorithms. The theories produced by ILP have also been generally more comprehensible than those using propositional methods.

## 2 An Application of ILP Scientific Discovery to Functional Genomics

In this second part of this paper we describe an example application of ILP to an important problem in functional genomics which illustrates how machine intelligence can be used to glean scientific knowledge from a large database.

### 2.1 Scientific Background

Molecular Biology is currently experiencing a data explosion. The sequencing of the first draft of the 3 billion bases of human genome is now complete `http://www.sanger.ac.uk/HGP/`), and the genomes of around 20 micro-organisms have now been completely sequenced (`http://www-fp.mcs.anl.gov/~gaasterland/genome.html`; Blattner, F. R. *et al.,* 1997; Cole, *et al.,* 1998; Goffeau, *et al.* 1996) as have those of the multicellular animals *Caenorhabditis elegans* (C. elegans Sequencing Consortium, 1998) and *Drosophila melanogaster* (Adams *et al.* (2000). The data from these sequencing projects is revolutionising biology. Perhaps the most important discovery from the sequenced genomes is that the functions of only 40-60% of the predicted genes are known with any confidence. For example in *Saccharomyces cerevisiae*, one of the most intensely studied organisms, of the ~6,000 predicted protein-encoding genes (Goffeau, *et al.* 1996), the function of only ~60% can be assigned with any confidence. The new science of *functional genomics* (Hieter & Boguski, 1997; Bussey, 1997; Bork *et al.,* 1998; Brent, 1999; Kell & King, 2000) is dedicated to determining the function of the genes of unassigned function, and to further detailing the function of genes with purported function.

Functional genomics needs better ways of being able to accurately predict a newly sequenced proteins function from its sequence. Currently this is done by using sequence similarity methods to find a similar (homologous) protein in the database that has a labelled function (Pearson & Lipman, 1988; Altschul, *et al.,* 1997). The function of the new sequence is then inferred to be the same as the homologous protein as it has been conserved over evolution. It is a kind of nearest-neighbour type inference(in sequence space).

## 2.2 Methodology

We have developed a complementary approach to sequence similarity methods for predicting a proteins function from sequence (King, *et al.*, 2000). This approach is based on learning symbolic rules to predict a proteins functional class. To test this approach we selected the *Mycobacterium tuberculosis* genome. This bacteria is the causative agent of tuberculosis which kills ~2 million people each year, and concern about the growing epidemic has led the World Health Organisation to declare tuberculosis a global emergency `http://www.who.int/inf-fs/en/fact104.html`. The genome of *M. tuberculosis* has been recently sequenced and has over 4 million base pairs, and 3,924 identified genes (Cole, *et al.,* 1998). After sequencing the genome was annotated and a database of proteins of identified function formed `http://www.sanger.ac.uk/Projects/M_tuberculosis/gene_list_full.shtm` (15/03/1999). This database contains details of all the proteins in *M. tuberculosis* of known function from experiment, as well as those with predicted function from sequence similarity searches. These assignments of function were organised in a strict hierarchy, where each higher level in the tree is more general than the level below it, and the leaf nodes are the individual functions of proteins - this is typical of current genome annotation. A subsection of the function hierarchy is shown in Figure 1. A typical protein in the genome is L-fuculose phosphate aldolase (Rv0727c fucA), its top-level class assignment is "Small-molecule metabolism", its second-level class is "Degradation", and its third-level class is "Carbon compounds". (Note that there are errors in annotation of function (Brenner, 1999), and proteins may have more than one function which adds "noise" to the assignments). *The organisation of functions into classes allows generalisation of the sequences over these classes using machine learning.*

To describe the sequences for generalisation we first formed a datalog database (Ullman, 1988) containing all the data we could find on these sequences. The most commonly used technique to gain information about a sequence is to run a sequence similarity search, and this was used as the starting point in forming descriptions. The basic data structure in the deductive database is the result of a PSI-BLAST sequence similarity search (Altschul, *et al.,* 1997). For each protein in the genome we formed an expressive description based on: the frequency of singlets and pairs of residues in the protein; the phylogeny ("family tree") of the organism from which each homologous protein was obtained - from SWISS-PROT (Bairoch A. & Apweiler (2000) (a standard protein database); SWISS-PROT protein keywords from homologous proteins; the length and molecular weight of the protein; and its predicted secondary structure (Ouali & King, 2000). In total 5,895,649 datalog facts were generated. Such a database is clearly too large to analyse manually.

We used the ILP data mining program Warmr (Dehaspe, *et al.*, 1998) to identify frequent patterns (conjunctive queries) in the database. Warmr is a general purpose data mining algorithm that can discover knowledge in structured data, where patterns reflect the one-to-many and many-to-many relationships of several tables. *This is not possible with standard data mining programs.* Warmr uses the efficient levelwise method known from the Apriori algorithm (Fayyad, *et al.,* 1996). This allows it to be used on very large databases. The Warmr levelwise search algorithm (Mannila & Toivonen, 1997) is based on a breadth-first search of the pattern space. This space is ordered by the generality of patterns. The levelwise method searches this space one level at a time, starting from the most general patterns. The method iterates between candidate generation and candidate evaluation phases: in *candidate generation*, the lattice structure is used for pruning non-frequent patterns from the next level; in the *candidate evaluation* phase, frequencies of candidates are computed with respect to the database. Pruning is based on the monotonicity of specificity with respect to frequency if a pattern is not frequent then none of its specialisation can be frequent. The application of Warmr can be considered as a way of identifying the most important structure in a database. In the *M. tuberculosis* database Warmr discovered ~18,000 frequent queries. These frequent patterns were converted into 18,000 Boolean (indicator) attributes for propositional rule learning, where an attribute gets value 1 for a specific gene if the corresponding query succeeds for that gene, and 0 if the query fails. The propositional machine learning algorithms C4.5 and C5 (Quinlan, 1993) were used to induce rules that predict function from these attributes. Good rules were selected on a validation set, and the unbiased accuracy of these rules estimated on a test set. Rules were selected to balance accuracy with unidentified gene coverage. The prediction rules were then applied to genes that have not been assigned a function to predict their functions.

## 2.3 Results

It was possible to find good rules that predict function from sequence at all levels of the functional hierarchies, as shown in Table 1. The test accuracy of these rules is far higher than possible by chance. *Of the genes originally of unassigned function class, the rules predicted 985 (65%) to have a function at one or more levels of the hierarchy.* The rule learning data, the rules, and the predictions, are given at: `http://www.aber.ac.uk/~dcswww/Research/bio/ProteinFunction/`. We illustrate the value of the rules by describing rule TB_C50_1_26 shown in Figure 2.

*The most important scientific result of this work was the unexpected discovery that it was possible to predict a protein's function in the absence of homology to a protein of known function.* To demonstrate this we carried out all-against-all sequence similarity searches using PSI-BLAST for those proteins correctly predicted by each rule. If all the proteins could be linked together by PSI-BLAST e-values $< 10$ (a very liberal definition) then the proteins were considered homologous. It was found that many of the predictive rules were more general than possible using sequence homology. Rules were found to correctly predict the function of sets of proteins that are not homologous to each other in the test set, and to correctly predict the function of proteins that are not homologous to any in the training data (Table 1). We speculate that such rules are caused by convergent evolution causing forcing proteins with similar function to resemble each other, or horizontal evolution has transferred functional related groups of protein into the organisms.

# 3 Intelligent Databases

We are working on extending our work on *M. tuberculosis* by developing an intelligent database (ID) for its functional genomics data. The ID will be designed to *add scientific value to the underlying data.* The rough design of the ID is given in Figure 3. The database will:

1. Store the large amount of new data being generated on the *M. tuberculosis* genome, e.g. from transcriptome (DeRisi *et al.*, 1997; Brown & Botstein, 1999; Alizadeh *et al.*, 2000), proteome (Humphery-Smith et al., 1997; Blackstock & Weir, 199), and metablome (Oliver & Baganz, 1998; Gilbert *et al.*, 1999) experiments.
2. Enable standard database queries of the data.
3. Incorporate background molecular biology knowledge.
4. Enable deductive inferences involving the data and the background knowledge.
5. Enable inductive inferences involving the data and the background knowledge.

To populate the database, data will be collected through our existing research links to the functional genomics projects for *M. tuberculosis*, and from primary bioinformatic databases. The basic data will be stored in a standard relational database management system. The background biological knowledge necessary for deduction will be stored in a connected module with the deduction engine. One important element that is missing from the use of logic programs to represent scientific knowledge is a way of representing uncertainty (Russell & Norvig, 1995). How best to do this is unclear, but we favour Bayesian probabilistic methods. Probability theory is often considered to be grounded on propositional logic (Jaynes, 1994), and until recently little work has been done combing logic programs with probabilities. However, there has been a recent upsurge of research in this area (e.g. Muggleton, 2000; Kersting & DeRaedt, 2000).

Induction will be carried out using conventional (propositional) data mining and Inductive Logic Programming (ILP). The ILP data mining will be connected with the background knowledge and deductive engine. A small section of the planned structure of data and background knowledge of ID is shown in Figure 4. The results of the induction will be stored with the basic data in the relational database - as in an inductive database (Mannila, 1997). The ID will be made accessible to the Scientific community using a variety of standard methods, including Web Browsers, ODBC, XML, CORBA, etc.

The aim of the Intelligent Database is to directly integrate scientific databases and scientific discovery tools to provide scientists with a tool which can help automate science. We believe that such Intelligent Databases are essential to analyse the large amounts of new scientific data that is being generated and to refine this data into scientific knowledge.

## Acknowledgements

We would like to thank Ugis Sarkans of the EBI and Nigel Hardy of the Department of Computer Science, University of Wales, Aberystwyth.

## References

1. Adams *et al.* (2000) The genome sequence of Drosophilia Melanogaster. *Science*, **287**, 2185-2195
2. Alizadeh, A. *et al.*. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403,** 503-511.
3. Altschul, S. F. *et al.,* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acid Res.*, **25,** 3389-3402.
4. Bairoch A. & Apweiler R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL *Nucleic Acids Res.*, 28, 45-48.
5. Blackstock, W. P. & Weir, M. P. (1999). Proteomics: quantitative and physical mapping of cellular proteins. *Tibtech* **17,** 121-127.
6. Blattner, F. R. *et al.,* (1997) The complete genome sequence of *Escherichia coli* K-12. *Science* **277**, 1453-1461.
7. Boden, M. (1977) *Artificial intelligence and natural man.* Brighton, Sussex: The Harvester Press
8. Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M. & Yuan, Y. P. (1998). Predicting function: From genes to genomes and back. *Journal of Molecular Biology* **283,** 707-725.
9. Bowers, A. F., Giraud-Carrier, C., & Lloyd, J. W. (2000) Classification of Individuals with Complex Structure In *Proceedings of the Seventeenth International Conference on Machine Learning* (ICML'2000), pages 81–88. Morgan Kaufmann.
10. Brenner, E. 1999. Errors in gene annotation. *Trends in Genetics* **15:** 132-133.
11. Brent, R. (1999). Functional genomics: Learning to think about gene expression data. *Current Biology* **9,** R338-R341.
12. Brown, P. O. & Botstein, D. (1999). Exploring the new world of the genome with DNA microarrays. *Nature Genetics* **21,** 33-37.
13. Buchanan, B.G., Sutherland, G.L., & Feigenbaum, E.A. (1969) Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry in *Machine Intelligence 4* (eds. B. Meltzer & D. Michie) Edinburgh University Press. pp 209-254.
14. Bussey, H. (1997). 1997 ushers in an era of yeast functional genomics. *Yeast* **13,** 1501-1503.

15. elegans Sequencing Consortium (1998) Genome Sequence of the Nematode *C. elegans*: A Platform for Investigating Biology *Science* **282**, 2012-2018.

16. Cole, S. T. *et al.,* (1998) Deciphering the biology of *Mycobacterium tuberculosis* from the complete genome sequence. *Nature* **393**, 537-544.

17. Dehaspe, L., Toivonen, H. & King, R. D. (1998). Finding frequent substructures in chemical compounds. In *The Fourth International Conference on Knowledge Discovery and Data Mining.*, AAAI Press, Menlo Park, CA., pp. 30-36.

18. DeRisi, J. L., Iyer, V. R. & Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278,** 680-686.

19. Dzeroski S., Blockeel H., Kompare B., Kramer S., Pfahringer B., Van Laer W. (199) Experiments in Predicting Biodegradability, in*: Proceedings Ninth International Workshop on Inductive Logic Programming*, Springer, 1999.

20. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press., Boston.

21. Finn, P., Muggleton, S., Page, D., and Srinivasan, (1998) A. Pharmacophore discovery using the inductive logic programming system Progol. *Machine Learning*, **30**, 241-271.

22. Gilbert, R. J., Johnson, H. E., Winson, M. K., Rowland, J. J., Goodacre, R., Smith, A. R., Hall, M. A. & Kell, D. B. (1999). Genetic programming as an analytical tool for metabolome data. In *Late-breaking papers of EuroGP-99* (ed. W. B. Langdon, R. Poli, P. Nodin and T. Fogarty), Software Engineering, CWI. pp. 23-33.

23. Goffeau, A. *et al.,* (1996) Life with 6000 genes. *Science* **274**, 546-567.

24. Gordon, A, Sleeman, D & Edwards, P (1995) Informal Qualitative Models: A Systematic Approach to their Generation R Valdes-Perez (Ed), *Proceedings of AAAI 1995 Spring Symposium on Systematic Methods of Scientific Discovery*, SS-95-03, AAAI Press, 18-22, 1995.

25. Helma C., Kramer S., & Pfahringer B. (1998) Carcinogenicity Prediction for Noncongeneric Compounds: Experiments with the Machine Learning Program SRT and Variou Sets of Chemical Descriptors, in: Proceedings 12th European Symposium on Quantitative Structure-Activity Relationships.

26. Hieter, P. & Boguski, N. (1997). Functional genomics: it's all how you read it. *Science* **278,** 601-602.

27. Humphery-Smith, I., Cordwell, S. J. & Blackstock, W. P. (1997). Proteome research: complementarity and limitations with respect to the RNA and DNA worlds. *Electrophoresis* **18,** 1217-1242.

28. Kell, D., King, R. D., On the optimization of classes for the assignment of unidentified reading frames in functional genomics programmes: the need for machine learning. *Trends in Biotechnology* **18**, 93-98 (2000).

29. Kersting, K. & De Raedt, L. (2000) Bayesian Logic Programs In *Machine Intelligence 17* (this volume).

30. King, R. D., Muggleton, S., Lewis, R. A. & Sternberg, M. J. E. (1992). Drug design by machine learning - the use of inductive logic programming to model the structure-activity-relationships of trimethoprim analogs binding to dihydrofolate-reductase. *Proc. Natl. Acad. Sci.* **89,** 11322-11326.

31. King, R. D., Clark, D. A., Shirazi, J. & Sternberg, M. J. E. (1994). On the use of machine learning to identify topological rules in the packing of beta-strands. *Protein Engineering* **7,** 1295-1303.

32. King, R. D., Muggleton, S. H., Srinivasan, A. & Sternberg, M. J. E. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. Natl. Acad. Sci.* **93,** 438-442.

33. King, R.D., Karwath, A., Clare, A., & Dehapse, L. (2000) Genome scale prediction of protein functional class from sequence using data mining. In: *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* (eds. R. Ramakrishnan, S. Stolfo, R. Bayardo, & I Parsa) The Association for Computing Machinery, New York, USA. pp. 384-389.

34. Jaynes, E. T. (1994). *Probability theory: The logic of Science.* `http://omega.albany.edu:8008/JaynesBook.html`.

35. Langley, P., Simon, H.A., Bradshaw, G.L., & Zytkow, J.M. (1987) *Scientific Discovery: Computational Explorations of the Creative Process.* Cambridge MA, MIT Press.

36. Lavrac, N. & Dzeroski, S. (1994). *Inductive logic programming: techniques and applications.* Ellis Horwood, Chichester.

37. Mannila H. (1997) Inductive database and condensed representations for data mining *Proc. International Logic Programming Symposium* (ed. J. Maluszynski) 21-30 MIT Press.

38. Mannila, H., & Toivonen, H. (1997) Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery* **1**. 241-258.

39. Muggleton, S. H. (1990). Inductive Logic Programming. *New Generation Computing* **8,** 295-318.

40. Muggleton, S. H. (1992). Inductive Logic Programming. Academic Press, London.

41. Muggleton, S., King, R. D. & Sternberg, M. J. E. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering* **5,** 647-657.

42. Muggleton, S. (2000) Learning Stochastic Logic Programs In *Machine Intelligence 17* (this volume).

43. Oliver, S. G. & Baganz, F. (1998). The yeast genome: systematic analysis of DNA sequence and biological function. In *Genomics: commercial opportunities from a scientific revolution* (ed. L. G. Copping, G. K. Dixon and D. J. Livingstone), pp. 37-51. Bios, Oxford.

44. Ouali, M., & King, R.D. (2000) Cascaded multiple classifiers for secondary structure prediction. *Prot. Sci.*, **9**, 1162-1176.

45. Pearson, W. R., Lipman, D. J., Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444-2448 (1988).

46. Quinlan, R., *C4.5: Programs for machine learning* (Morgan Kaufmann, San Mateo, 1993).

47. Reichardt, T. (1999) Its sink or swim as a tidal wave of data approaches. *Nature.* **399**, 517-520.

48. Sleeman, D.H., Stacy, M.K., Edwards, P., & Gray, N.A.B. (1989). An Architecture for Theory-Driven Scientific Discovery. In *Proceedings of the Fourth European Working Session on Learning.* (ed. K Morik). London: Pitman, pp 11-23.

49. Srinivasan, A. & King, R. D. (1999). Feature construction with Inductive Logic Programming: A study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery* **3,** 37-57.

50. Sternberg, M. J. E., King, R. D., Lewis, R. A. & Muggleton, S. (1994). Application of machine learning to structural molecular biology. *Philosophical Transactions of the Royal Society of London Series B- Biological Sciences* **344,** 365-371.

51. Turcotte, M., Muggleton, S.H., &. Sternberg, M.J.E. (2000) The effect of relational background knowledge on learning of protein three-dimensional fold signatures. Machine Learning. (in press).

52. Ullman, J.D. (1988) *Principles of databases and knowledge-base systems*, vol 1. Rockville, MD: Computer Science Press.

53. Valdes-Perez R.E. (1999) Discovery Tools for Science Applications. *Communications of the ACM*, **42**, 37-41,

| | **Level 1** | **Level 3** | **Level 4** |
|---|---|---|---|
| Number of rules found | 25 | 20 | 3 |
| Rules predicting more than one homology class | 19 | 8 | 1 |
| Rules predicting a new homology class | 14 | 1 | 0 |
| Average test accuracy | 62% | 62% | 76% |
| Default test accuracy | 48% | 6% | 2% |
| New functions assigned | 886 (58%) | 60 (4%) | 19 (1%) |

**Table 1.** The number of rules found are those selected on the validation set. A rule predicts more than one homology class if there is more than one sequence similarity cluster in the correct test predictions. A rule predicts a new homology class if there is a sequence similarity cluster in the test predictions that has no members in the training data. Average test accuracy is the accuracy of the predictions on the test proteins of assigned function (if conflicts occur, the prediction with the highest *a priori* probability is chosen). Default test accuracy is the accuracy that could be achieved by always selecting the most populous class. "New functions assigned" is the number of ORFs of unassigned function predicted.

**Figure 1.** Please refer to the article's webpage for access to this figure. An example subset of the genes functional hierarchy in *M. tuberculosis*. The gene L-fuculose phosphate aldolase is in the Level 3 class "carbon compounds". This example has only three out of four possible classification levels.

```
If
  a protein's percentage composition of lysine is > 6.6%
Then
 its functional class is ''Macromolecule metabolism''
```

**Figure 2.** This top-level rule is 85% (11/13) accurate on the test set (the probability of this result occurring by chance is estimated at $1.2\text{x}10^{-5}$ as the class Macromolecule metabolism covers ~25% of examples). The rule correctly predicts the following proteins (rpsG (S7), rpsI (S9), rpsL (S12), rpsT (S20), rplJ (L10), rplP (L16), rplS (L19), rplX (L24), rpmE (L31), rpmJ (L36), infC (IF-3)). These proteins are all involved in protein translation. When the training data are included the rule covers 46 out of the 58 proteins known to be involved in ribosomal protein synthesis and modification. The rule predicts the function of fifteen genes of unknown function. It is also consistent with protein chemistry, as lysine is positively charged which is desirable for interaction with negatively charged RNA. The choice of lysine over arginine for the positively charged residue may be connected with the high GC content of the *M. tuberculosis* genome - lysine is coded by the codons AAA and AAG; while arg is coded by CGU, CGC, CGA, CGG; and his by CAT and CAC.

**Figure 3.** Design of the proposed Intelligent Database. Please refer to the article's webpage for access to this figure.

**Figure 4**. Description of part of background knowledge in *M. tuberculosis* intelligent database. Please refer to the article's webpage for access to this figure.