The Knowledge Engineering Review, Vol. 9:3, 1994, 287–300

Artificial intelligence and knowledge based systems in molecular biology*

JOHN FOX[1] and CHRISTOPHER J. RAWLINGS[2]

[1]*Advanced Computation Laboratory,* [2]*Biomedical Informatics Unit, Imperial Cancer Research Fund, PO Box 123, Lincoln's Inn Fields, London WC2A 3PX, UK*

Abstract

Over the last ten years, molecular biologists and computer scientists have experimented with various artificial intelligence techniques, notably knowledge based and expert systems, qualitative simulation, natural language processing and various machine learning techniques. These techniques have been applied to problems in molecular data analysis, construction of advanced databases and modelling of biological systems. Practical results are now being obtained, notably in the representation and recognition of genetically significant structures, the assembly of genetic maps and prediction of the structure of complex molecules such as proteins. The paper outlines the principal methods used, surveys the findings to date, and identifies promising trends and current limitations.

1 Introduction

It is now generally accepted that modern molecular biology research needs many different types of software to support the management, analysis and interpretation of data. It is therefore not surprising that the inherent complexities of biology, with its many ill-defined problems, would attract Artificial Intelligence (AI) practitioners in the hope that their technologies can provide a fresh outlook on many of the difficult scientific problems that are facing molecular biology today. Indeed, AI and molecular biology is now clearly emerging as a distinctive interdisciplinary subject, with a rapidly growing research community (Hunter, 1993; Hunter et al., 1993). In this paper, we review some of the more prominent recent developments in the field, and consider a range of AI techniques and applications. The discussion is generally confined to areas where there are sufficient results to draw some conclusions, if provisional ones.

AI is, of course, a diverse field which is developing rapidly on a number of fronts, not all of which are of much immediate interest to molecular biologists. AI research aimed at developing theories of animal and human reasoning, planning, learning, vision, hearing and natural language understanding, for example, are so far of little direct relevance. Similarly, work on developing AI as a formal discipline (e.g. theories of reasoning and problem solving grounded in mathematical logic) are not as yet having a direct impact, though arguably, they are having an indirect effect through their influence on the development of AI tools, programming languages and development methods. The areas where AI is having its strongest impact are where technologies developed in AI, such as knowledge based and expert systems and neural networks, are sufficiently mature to be used pragmatically to address recognized practical problems, without concern for theoretical or philosophical subtleties.

Molecular biology also has many aspects, ranging from understanding the mechanisms of genetics, modelling the structure and interactions of complex molecules such as proteins, to the causes and processes of diseases. AI techniques are being applied in all of these areas, though to

*This paper is based on a previous article published in the *Philosophical Transactions of the Royal Society of London*, Series B, 1994.

ANIMAL-GENES
  PROTEIN-CODING-GENES
    CONTRACTILE-PROTEIN-GENES
    GLOBIN-GENES
    HEAT-SHOCK-GENES
    HISTONE-GENES
PLANT-GENES ... etc.

Figure 1 A portion of the class hierarchy in the GENESIS genes knowledge base

date mostly in the first two areas. In fundamental genetics they are being used to create integrated knowledge bases with encyclopaedic coverage of topics in molecular genetics; to simulate biochemical processes and the lifecycles of simple organisms; to identify structures in DNA sequences and learn new rules for recognizing active genes in DNA sequence data. In molecular modelling they are being used to predict three-dimensional molecular structures from simpler biochemical data, and on learning rules for predicting how molecules such as proteins fold in three dimensions.

We have selected five AI topics for more detailed presentation here, on the basis that they have matured sufficiently to yield practical techniques for use in molecular biology. These are knowledge based systems, qualitative simulation, machine learning, language processing and search. After describing each of these areas, illustrating the use of AI techniques with example applications, we identify general themes and areas of success.

2 Advanced databases and knowledge bases

A feature of molecular biology is its capacity to generate prodigious quantities of data. The human genome project, which aims to analyse the human genetic "blueprint", will yield a database of raw DNA sequence data of the order of 10^9 items, for example. The management and interpretation of genetic and other data is therefore becoming an acute problem for modern biology.

The daunting scale of the human genome project is not perhaps the most problematic issue for the field of bioinformatics. The difficulties of mere storage and retrieval of information are likely to be satisfactorily addressed by advances in conventional computer and database technology. The most significant factors will probably be the combined effects of the increasing performance/cost ratio of computer hardware, improved software technologies able to exploit parallel computer hardware, and advanced data networking. However, the field also faces a deeper problem, to do with the interpretation of such data as scientific understanding in molecular biology advances. Where conventional databases have been primarily concerned with supporting efficient storage and retrieval, knowledge based systems are being used to represent, manage and maintain, over time, knowledge obtained by interpreting the raw data stored in databases.

GENESIS (Friedland et al., 1982) was an early influential attempt to build an integrated knowledge based system for genetics; it was intended for use in laboratory data management and experiment planning. GENESIS provided a substantial database of detailed DNA sequence data, extended with documentation and derived data such as maps of larger scale genetic features. The GENESIS knowledge base was organized into frames representing biological concepts and their interrelationships. For example, a fragment of the GENESIS knowledge base dealing with genes is shown in fig. 1. Each name in this hierarchy refers to a frame which defines the attributes which characterize each class of genes.

Organizing a knowledge base as a frame hierarchy is intuitively natural from a biologist's point of view, but it also offers a number of technical benefits as well. The most important of these is the inheritance of knowledge over concept classes. The frame-based representation and the inheritance mechanisms used are closely related to those used in object-oriented (OO) database systems (Cattell, 1991), but the knowledge based systems used in molecular biology differ from standard OO databases in the provision of AI programming languages such as Lisp, Prolog or special rule-

based reasoning systems. GENESIS provided a rule-based programming language, GENGLISH, which permitted a knowledge base designer to attach data manipulation rules to particular concepts in the hierarchy. Operators in these rules permitted the user to simulate the activity of key enzymes used as reagents in genetic engineering (see later).

These and other techniques have proved to have lasting value in the development of molecular biology applications of knowledge bases. Yoshida et al. (1992) are developing LUCY, a "human genome encyclopaedia", which is intended to provide a uniform structure for integrating a range of public and private laboratory databases together with over 40 different types of large-scale genetic maps. Like GENESIS, the LUCY system organizes knowledge as a frame structure and provides specialized languages appropriate for the representation of molecular biological data.

Unlike GENESIS, LUCY does not have a special-purpose inference language, but provides a Prolog interpreter (Clocksin & Mellish, 1981). Prolog's value as a knowledge representation and knowledge base query language was realized early in molecular biology (e.g. Lyall et al., 1984; Rawlings et al., 1985), and some have argued that it has additional advantages over conventional programming techniques for use in molecular biology (e.g. Barton & Rawlings, 1990). The Argonne laboratories have put considerable effort into using logic programming techniques in developing integrated knowledge bases to support research in human (Hagstrom et al., 1992) and bacterial genetics (Baehr et al., 1992). A number of studies have also looked at the combination of logic programming with frame and object-based data representations. Gray et al. (1990) apply this combination of techniques to protein structure analysis, and conclude that the two methods are complementary.

Other logic-based knowledge-based systems include the GeneSys system (Overton et al., 1990). This explores issues in the automation of biosequence analysis, notably with respect to relationships between molecular structure and function in the operation of genes. The PAPAIN system (Clark et al., 1990) is designed to automate the analysis of protein sequences and protein structure prediction, and makes use of a variety of logic techniques including deductive databases and constraint logic programming (discussed later).

3 Qualitative modelling and simulation

Because many aspects of molecular biology are not amenable to rigorous mathematical treatment, qualitative approaches for modelling biochemical processes have been developed. The MOLGEN project (Stefik, 1981; Friedland & Iwasaki, 1985) was one of the first experiments in the qualitative simulation of molecular biological processes. The motivation behind MOLGEN was to plan genetic engineering experiments automatically. To do this, it was necessary to simulate the activity of the reagents and processes of molecular genetics, e.g. cutting and joining DNA sequences and translating them into their equivalent amino acid sequences. The ideas of automated experiment planning have also been developed by Carhart et al. (1988) and Jiang et al. (1990).

In the MOLGEN-II project (Friedland & Kedes, 1985), the emphasis shifted to developing programs that could (re)discover scientific hypotheses and, in particular, the reasoning involved in elucidating how gene operation is controlled. This research has required the development of techniques for representing and reasoning about biological processes and experimental techniques. An important outcome of MOLGEN-II is a qualitative model and simulation of the operation of an important genetic mechanism, the feedback and control processes that regulate the production of an enzyme involved in protein synthesis, the trp operon (Karp, 1993). The approaches and technologies developed to support qualitative molecular biological simulations used in the MOLGEN project have also influenced simulations of genetic regulation (Meyers & Friedland, 1984) and DNA metabolism (Brutlag et al., 1991).

As fundamental biological processes, gene regulation and expression have also been studied in some detail (Weld, 1984; Koton, 1985), and in some cases, models of the lifecycle of simple organisms such as the lambda bacteriophage (Meyers & Friedland, 1984) and the human immunodeficiency virus (Koile & Overton, 1989) have been developed.

Many people have observed that for knowledge based systems to make a significant impact on molecular biology, then they should have a basis in the "common-sense biochemistry" as taught to every undergraduate biologist (e.g. Karp, 1992). The EcoCYC project (Karp & Riley, 1993) is proposing to do this in a comprehensive way for the genetics and biochemistry of the bacterium Escherichia coli. Others are concentrating on methods for representing the detailed aspects of cellular metabolism (Mavrovouniotis, 1993a, and further developed in Mavrovouniotis, 1993b, & Kazic, 1993).

4 Machine learning

Ever since scientists began to collect and store large collections of data on computers, they have been fascinated by the idea that it might be possible to develop techniques for "discovering" patterns in the data that would be hard for them to find unaided. The classical approach, of course, is for a scientist to guide a computer in searching for patterns in a database to confirm specific suspicions or hypotheses. But there is also a growing body of work on techniques whereby the computer searches for correlations and formulates hypotheses without the guidance of a human investigator. This work ranges from attempts to find statistical regularities automatically (e.g. Blum, 1982) to highly ambitious projects aimed at showing that a computer can formulate scientific conjectures and theories without human intervention (e.g. Lenat, 1983).

The value of unsupervised discovery methods is controversial, particularly among scientists who are sceptical about whether computers can originate new concepts or hypotheses. However, in a limited form, machine learning techniques for automatically finding regularities in large collections of data are promising to be useful in molecular biology, the most prominent of which are neural networks and rule induction.

Neural networks were developed in the 1950s by researchers interested in modelling brain mechanisms involved in perception. They developed artificial networks in hardware which could learn to recognize patterns, presented as sets of features, by progressively increasing the numerical weight associated with features which are typically present in specific categories of pattern, and decreasing the weight attached to features which are typically absent. Such a network can be trained to distinguish the different categories by providing feedback indicating whether an example is or is not a member of a particular category. Perceptrons, as the early networks were called, subsequently fell out of favour because of demonstrations by Minsky and Papert (1969) that they could not learn to discriminate some important types of pattern, although perceptron-like networks were used by Stormo et al. (1982) to recognize specific biologically important sites in DNA sequences. Recent technical advances, however, have overcome enough of these difficulties that neural network software now appear to be useful in many other practical pattern recognition applications.

Among the first applications in molecular biology was the prediction of the secondary structure of proteins. Proteins are made of long chains of amino acids which in their natural environment (in solution) fold up into simple "secondary" structures, like helices, and then by further folding into higher-order structures. Qian and Sejnowski (1988) trained a network by presenting it with amino acid sequences whose secondary structure was known, and tested its ability to correctly classify new sequences which were structurally different from the training set. The initial successful recognition rate with randomly assigned weights was at the chance level of 33%, but during training this rose to an average of 64.3% for the three states, a performance level that was superior to other methods available at the time. The reader is directed to a number of recent reviews that cover the use of artificial neural networks for predicting structure and function in both protein and DNA sequences, (Hirst & Sternberg, 1992; Presnell & Cohen 1993; Steeg, 1993; and Holbrook et al. 1993).

The use of neural networks to detect errors in biological sequences was suggested by Brunak et al. (1991). They trained a network to recognize genetic patterns in 33 human genes. During this

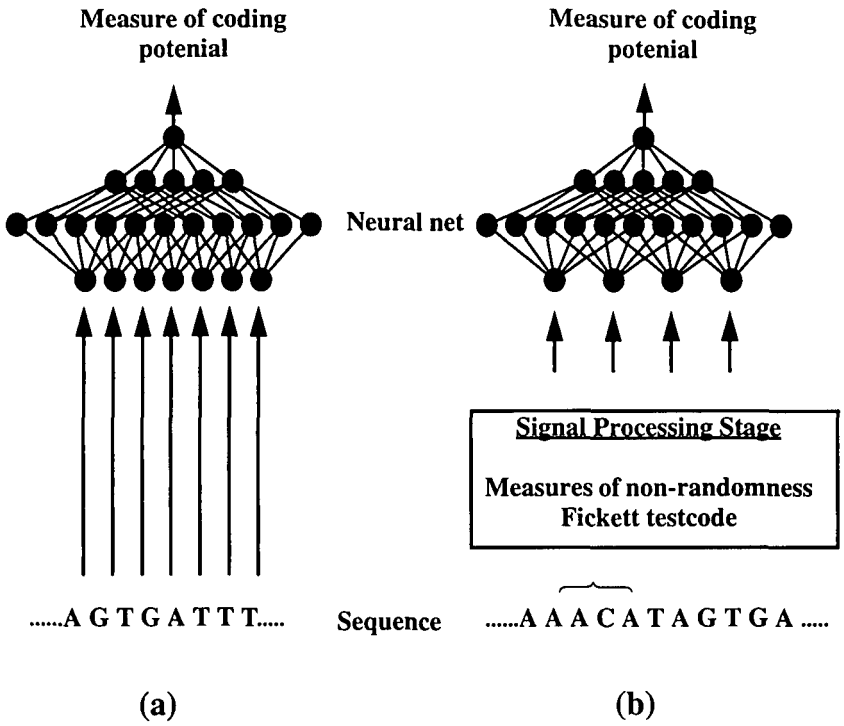


Figure 2 Use of neural nets for analysing sequences of nucleic acids A,C,G,T (Adenine, Guanine, Cytosine, Thymine) in DNA. (a) Learning based directly on the raw sequence data, and (b) based on statistical data obtained by preprocessing the sequence

study they noticed that some sequences appeared to disturb the learning in that the network weights did not stabilize on a specific pattern classification. Subsequent investigation revealed discrepancies from the original papers for three genes, due to misprints and other errors of interpretation. A further study on 241 sequences revealed nine new errors. They argue that neural networks could be used as "computerized proof readers" to detect possible errors before accepting data into a database.

A more recent development in the use of neural networks in molecular biology is to combine them with conventional analysis techniques. Mural et al. (1992) have examined the use of neural networks in finding regions of the genome which code for functional proteins. Computer-based recognition of DNA features can be difficult; statistical analysis can help, but in many cases the consensus sequence is insufficient to specify the feature of interest. They argued that results from "knowledge-free" methods (i.e. those which simply use the sequence, fig. 2a) are encouraging, but the networks are large (particularly for complex features such as protein coding regions), and training requires large amounts of supercomputer time. They therefore took a different approach, in which they preprocessed the sequence statistically, to first identify potentially meaningful biological patterns (fig. 2b), labelling the sequence along its length with seven different measures. These different labellings were then used as input to the neural network, rather than the raw DNA sequence. Using this method, they located 90% of the gene coding regions they were looking for, and correctly classified 96% of sequences coming from coding/non-coding sequences. Of the 1113 cases classified as coding, 92% were correct (8% false positives).

Craven and Shavlik (1993) compared neural network techniques with conventional statistical methods, predicting that neural networks should outperform simple statistical methods because they do not make the assumption that the statistics of neighbouring elements in a DNA sequence

are independent. They found that the best conventional techniques yielded 87.2% correct assignment of subsequences as coding/non-coding. The best neural network resulted in 87.45% correct assignment, but when statistical techniques were used to define the features of coding regions, the neural network learned to make 89.15% correct assignments.

While these learning techniques have promise for a variety of applications in molecular biology, they have weaknesses, notably that what they learn is difficult to understand (since this is merely a number of weights distributed within the network rather than a biologically meaningful structure). As Weiss and Kulikowski (1991) put it, ". . . unfortunately, invoking even simple computations . . . can make many noncomputational experts very wary and distrustful of the results generated by a computer. At best the human user will lack a sense of participation in the prediction. At worst, the computer's answer will be accepted on faith, which can lead to misunderstandings and worse when the results are misapplied."

A different family of learning techniques has been developed in addition to statistical and neural networks methods which offers to address this problem. "Induction" techniques can be used to generate decision trees or collections of inference rules from sets of examples rather than quantitative parameters (Weiss & Kulikowski, 1991). A particularly powerful form of induction is Inductive Logic Programming (ILP), a variant of rule induction in which rules expressing first-order as well as simple (propositional) logical relationships between objects can be derived by the learning process. For example, the ILP system GOLEM (Muggleton & Feng, 1990) takes input descriptions and generates output rules as terms in the language Prolog. ILP is interesting because it is being used productively in molecular biology, and also because of its potential for development into a formal, well understood learning theory (Bergadano & Gunetti, 1994).

A typical application of ILP is in predicting properties of the amino acids in a protein, for example whether each amino acid will be found within a helix, or some other secondary structure. Muggleton et al. (1992) generated a set of rules using GOLEM which predicted the correct assignment of residues to structures in 81% of examples in a test set.

Sternberg et al. (1994) applied GOLEM to derive rules that relate the chemical structure of proteins into information about three-dimensional structure and function. However, the aim is not simply to obtain predictive rules, but also to express these rules using relationships which are understandable to the molecular biologist. In their work the input to the learning program is a set of Prolog clauses which describe the arrangements of secondary structures called strands, in terms of their "topological" relationships. The topology concerns whether strands are adjacent, connected, parallel, etc. for which information is available (in the form of a logic database (Rawlings et al. 1985)). Sternberg et al. successfully showed that their learning program could learn rules for predicting protein structure which were consistent with earlier analyses, and which may suggest new protein features that biologists might not identify in informal examination. Sternberg et al. (1994) also describe applications of ILP to "drug design".

The overall conclusion appears to be that rule induction in general, and inductive logic programming in particular, are promising techniques for identifying biologically significant features of complex molecules and, unlike conventional statistical methods and neural network techniques, produce their results in a form which is intelligible to biological scientists.

Purely symbolic methods may be open to the criticism that they cannot fully exploit quantitative features of the training material. Some workers have tried to obtain the best of both worlds by combining rule based and neural network techniques. For example, Shavlik et al. (1992) propose a scheme whereby a biological theory is expressed as a set of if . . . then . . . rules, such as a theory which relates raw DNA sequence data to higher-order genetic structures. These rules can be translated into an equivalent neural network structure, which can then be trained on examples in the usual way. The trained network can then be translated back into a set of rules, yielding a more refined theory. They carried out two experiments using this scheme, and compared the results with other machine learning and conventional techniques. They found the method to be both superior to other methods and, unlike normal neural networks, the results are in the form of explicit rules which are intelligible to biologists.

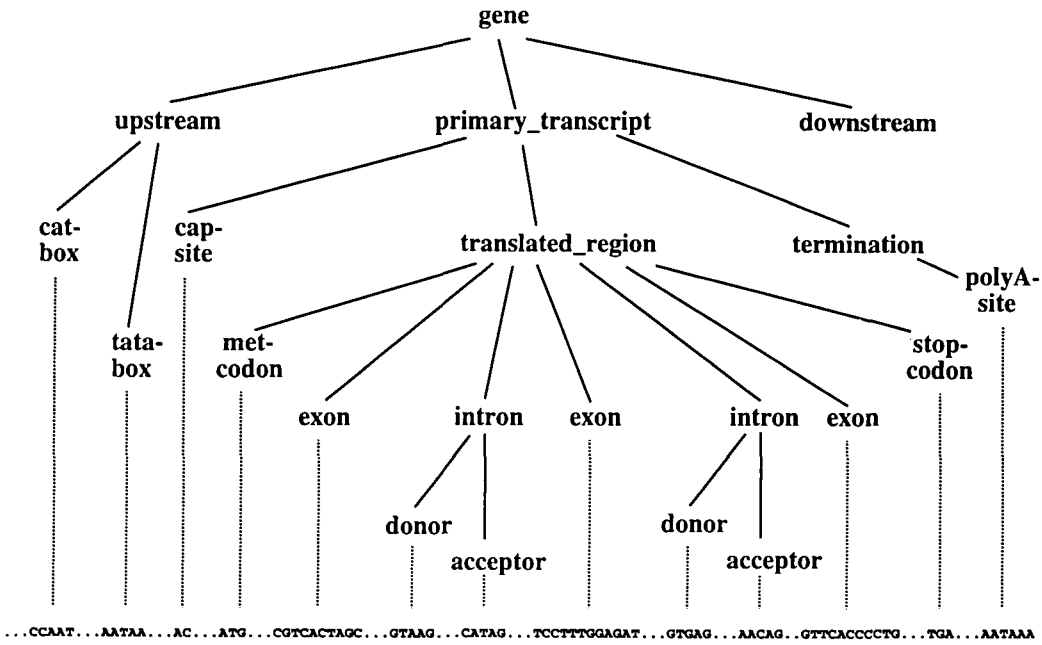


Figure 3 Analysis of a nucleic acid sequence by constructing a parse tree. This shows the "phrase structure" corresponding to the hierarchical organisation of biologically functional elements in the sequence

5 Techniques from language processing

It is a commonly held view that genetic sequences are similar to natural languages; as written languages rely on the sequence of letters and punctuation to convey meaning, so biological sequences carry meaning (descriptions of structure and function) in the linear order of their elements—nucleic acids in DNA; amino acids in proteins. The long term goal of computational genetic linguists is to develop accurate automatic methods to identify the biological meaning of the features encoded in molecular sequences. Searls (1993) provides an excellent overview of language theory and its application to molecular genetics, including the different types of grammatical systems and the classification of genetic grammars.

The practical value of formalising a language as a grammar is that it facilitates the construction of a program that can parse sentences from that language. The most useful outcome of a parser is the parse tree, and fig. 3 shows how a genetic "sentence" can be decomposed into other genetic "phrases" or elements described by a genetic grammar.

Many genetic sequence patterns associated with biological functions can be expressed using regular expressions, but most sequence pattern recognition programs (e.g. QUEST, Abarbanel et al., 1984; ARIADNE, Lathrop et al., 1987) use a pattern language that has been extended beyond pure regular expressions in order to accommodate some of the less regular features of biological sequences. Furthermore, while extended regular languages can capture many genetic features (e.g. PROSITE, Bairoch, 1991) and have resulted in some important practical programs, other aspects of nucleic acid and protein structure and function are not encoded in local sequence patterns, but are a consequence of local spatial interactions mediated by "long range" (in terms of molecular distance) and higher-order sequence structures. Consequently, to extend the range of biological meanings that can be recognized by linguistic methods, it is necessary to consider more complex languages.

In his discussion of the linguistic classification of genetic grammars, Searls (1993) takes as his starting point a Definite Clause Grammar (DCG) for representing genetic structures. DCGs have a close association with the Horn Clause Logic employed by Prolog (Pereira & Warren, 1980), and most Prolog systems are able to interpret and transform a DCG into an executable Prolog program which can be used as a top-down parser for the language described by the DCG. Prolog support for

DCGs is made flexible by the expedient of allowing grammar rules to be augmented with native Prolog code. In what Searls names a String Variable Grammar (SVG), further extensions to the DCG take advantage of the logic programming paradigm to provide features necessary to describe higher-order interactions among genetic sequences, and give SVG properties necessary to describe some of the context-sensitive features of nucleic acids (Searls & Liebowitz, 1990).

A practical demonstration of the use of these techniques is the processing of gene structure information in the GenBank DNA sequence database. In a paper focused on globin genes, Aaronson et al. (1993) were able to reveal a significant number of errors in the database. These included a small number of incorrectly specified features and, more significantly, a larger number of genetic elements that were missing completely from the GenBank feature table. By exploiting the gene grammar structure and the ability to reason by analogy among structurally similar sequences, it was possible to propose the existence of 30% more coding regions and 40% non-coding regions than were listed in the feature tables. As well as providing a new method for data quality control in GenBank, the linguistic approach suggests a possible way of dealing with the important problem of consistency and completeness of the genetic feature tables.

DNA and protein sequence databases are important primary data resources in molecular biology, but higher order databases, such as catalogues of sequences with known biological properties ("motifs"), are becoming increasingly important. Perhaps the most well known protein sequence motif collection is the PROSITE database (Bairoch, 1991). In PROSITE and other similar databases, extended regular expressions are used to represent the sites and sequence patterns. The coverage of the pattern description language is an important factor in the usefulness of the database, and affects the potential for the database to grow to cover more sophisticated (i.e. long-range) patterns as they are discovered.

In recent releases of PROSITE there have been a number of patterns identified that cannot be completely described by the PROSITE pattern language. Helgeson and Sibbald (1993) have addressed these problems using a formal linguistic approach, and have developed the PALM pattern language. PALM has many features in common with an SVG, but is focused on the requirements of representing protein sequence patterns—in particular frequentistic patterns using an operator which counts occurrences of particular patterns in a sequence segment.

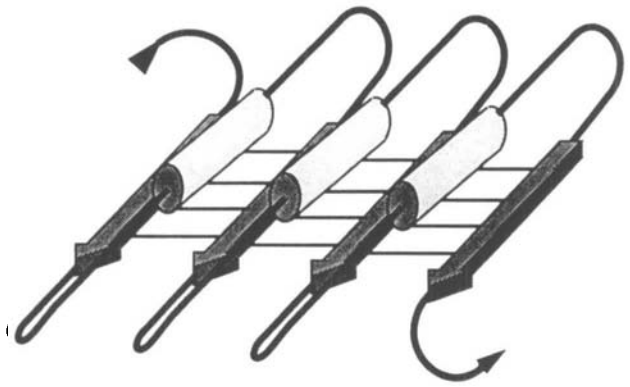
6 Constraint-based systems

Many problems of interpreting data from molecular biology experiments have combinatorial complexity (or worse), and computationally efficient methods are important if competing interpretations are to be evaluated in reasonable time. It can also be argued that many aspects of scientific reasoning can naturally be characterized as the search for consistent interpretations among data and the hypotheses that constrain the possible solutions. Constraint-based systems promise solutions to both these problems.

The constraint-based approach represents the dependencies among all the objects in a problem as constraints, and uses a problem solver that will prune illegal solutions and their consequents (constraint propagation) when a constraint is violated (Kumar, 1992).

The first example of an AI application in the molecular sciences, the DENDRAL system for interpreting mass spectrometer data (Lindsay et al., 1980) used a generate and test approach, which can be considered as the most straightforward constraint-based search method. The determination of protein structure from Nuclear Magnetic Resonance (NMR) spectra has also been addressed using constraint-based problem solving techniques. These and other AI approaches to protein structure determination from NMR spectra are reviewed by Edwards et al. (1993).

The prediction of the structure of ribonucleic acid (RNA) molecules is another problem that has been the subject of constraint-based approaches due to its combinatorial properties. In Heuze (1989), a constraint-based implementation of an established combinatorial method for predicting RNA structure (Gouy, 1989) uses parallel constraint logic programming (see later). Major et al.



(a)

No. of strands	2	6	10
No. of topologies	3	87,4800	3.57×10^{10}

(b)

Figure 4 The number of possible topological structures a protein can have increases exponentially with the number of components in the protein. (a) Typical topological structure of a protein which is made up of helices (cylinders) and strands (arrows) connected by coils; (b) growth in the theoretical number of possible topologies with increasing numbers of strands

(1991) used a hybrid approach in their MC-SYM system for predicting the three-dimensional structure of RNA. MC-SYM combines the use of a symbolic programming language (the functional language Miranda) to implement a constraint satisfaction algorithm which prepares a partial model of the RNA structure for a numerical program, which then computes the detailed energetic and structural constraints on RNA folding.

Protein topology prediction (hypothesizing the most plausible spatial organization of protein elements) has a known combinatorial complexity (fig. 4) for some classes of protein (Clark et al., 1991). By formulating this problem in constraint satisfaction terms and using qualitative folding rules as constraints, Clark et al. were able to test the accuracy and coverage of a number of protein folding rules published in the scientific literature. This was possible because the representation of constraints and the query language for their database of protein topological structures (Rawlings et al., 1985) were rules written in the Prolog language.

More recently, by analysis of a protein topology database and the use of inductive learning methods (see above and Sternberg et al., 1994), Clark et al. (1993) extended the number of constraints used during prediction, and re-implemented their method in the parallel constraint logic programming system, ElipSys. Constraint logic programming is a recent development in AI languages, extending logic programming languages such as Prolog with special purpose problem solving techniques from operations research and algebraic constraint propagation (Van Henteryk, 1991). Clark et al. (1993) demonstrated that CLP is well suited to solving these types of problems, both from the point of view of providing a concise and comprehensible representation language, and computational efficiency. They showed that they could achieve considerable performance gains (approximately 60 fold) over the original Prolog implementation. Furthermore, because the ElipSys system is also a parallel logic language, it was possible to increase performance

further on a parallel computer system as an almost linear function of the number of processing elements available.

A major problem in the use of a logic representation of protein folding rules is providing a means to deal with uncertain or partial constraints. Using in-built optimization operators in the ElipSys system, Clark et al. (1993) developed an uncertainty management scheme based on the number of times a constraint (rule) was found to be true (or false) in the protein topology database. Using this scheme, the most plausible topological structure(s) were those that violated the fewest or weakest constraints.

Protein topology prediction is a good example of knowledge-intensive constraint satisfaction where there is a relatively large number of high level constraints that can be used to prune the hypothesis space. At the other extreme are data-intensive problems, where there are few general rules but many individual constraints coming from experimental data. The assembly of ordered genetic maps from measures of distance between identifiable features is highly data-intensive, for example. The assembly of restriction maps (maps of sites where various enzymes cut the sequence) was identified early on as amenable to AI techniques, and constraint-based search in particular (Stefik, 1978, 1981). In the CPROP program (Letovsky & Berlyn, 1992), the construction of a genetic map is dealt with as a constraint satisfaction problem. All information about the order of features in the map is represented as constraints, and CPROP uses rules for combining local or partially ordered maps into larger maps. As regions are combined, new distance constraints are created and a constraint propagation phase is initiated to ensure that the derived information is kept consistent.

In recent work by Clark et al. (1994), the ElipSys parallel CLP system was used to program a constraint-based approach to assembling long-range genetic maps. Their program combines heuristic data reduction methods and constraint satisfaction to derive the optimal ordering of DNA features. Doursenot et al. (1993) were able to show the power of introducing different constraints to prune the hypothesis space and reduce program execution times. They also showed that the ability to build maps was as good (by the same criteria) as those generated by conventional means (Mott et al., 1993), and it was demonstrated, for the first time, that for a given set of data the result was optimal.

7 Discussion and assessment

In the early days of applied AI in molecular biology (and many other areas), there was great optimism that the leverage provided by AI technology would overcome some of the practical difficulties of using complex, and sometimes unwieldy AI programming environments. Furthermore, there was a general belief that the development of cost-effective symbolic computing hardware that could efficiently run AI programs written in languages such as Lisp and Prolog would ensure that such systems would eventually become widely accepted. For many reasons, however, this was not what happened, and increasingly the applied AI community has become more pragmatic in its approach, so that the emergence of hybrids of AI and conventional software techniques are now common. This trend is evident in AI applications in molecular biology, and perhaps most clearly seen in the area of advanced knowledge based systems.

One of the developments enabling AI methods to be applied to molecular biology problems is the convergence of AI and database technologies, which is leading to the new class of "deductive databases". These systems have the ability to manage large-scale data together with providing logical reasoning and other inference capabilities. These deductive and other knowledge based systems are being made the basis for building what are often referred to as "encyclopaedic systems" (e.g. Yoshida et al., 1992; Karp, 1992) that bring together not just data from a wide variety of sources into a common framework, but also simulations of biochemical processes, experiment planning, data interpretation and other functions. If these projects are successful, then they will greatly ease the problems encountered by many scientists when trying to assemble information from the many different molecular biology and genetics databases.

A major difficulty in earlier large-scale knowledge-based projects such as GENESIS (Friedland & Kedes, 1985) was the huge amount of biological detail that had to be encoded before significant results were possible. The rapidly changing nature of AI software technology and the absence of a consensus on programming environments has also meant that much work has had to be repeated. A challenge to the new projects involved in this research will be to ensure that as well as addressing the scientific issues (computer science and biological science), the necessary procedures are put in place to enable these digital encyclopaedias to be used by any computational biologist—not just those that are prepared to program in Lisp or Prolog.

Early applications of AI in molecular biology drew heavily on the capabilities of the Lisp programming language. Although Lisp is still popular in many centres, particularly in the USA, there is a trend developing for the adoption of logic programming languages, such as Prolog, for molecular biological applications. One advantage that Prolog brings is its close relationship with a relational style of programming and the facility with which it is possible to link to relational database systems. The Prolog language also provides the structures necessary to build higher-level representations such as object-oriented (Gray et al., 1990) and frame-based representations (e.g. Overton et al., 1990; Yoshida et al., 1992). The most active users of Prolog are principally the knowledge based systems development community (e.g. Clark et al., 1990; Hagstrom et al., 1992), but Prolog is also the language of choice for genetic grammar and linguistics research (Searls, 1993), where the close relationship between definite clause grammars and Prolog is a clear advantage.

The appealing notion that computer programs can be built that can hypothesize new relationships (e.g. rules for predicting protein structure from amino acid sequence) by learning them directly from molecular biological data has yielded mixed results. The most popular and successful approach for machine learning of molecular biological concepts has been artificial neural networks. However, not all commentators are wholeheartedly optimistic about their future (e.g. Hirst & Sternberg, 1992). One problem that this research area has helped to bring into focus, and that pervades computational molecular biology, is selecting the correct training and test data sets and having accepted criteria for success. There is an urgent need for scientists in computational molecular biology to establish a well documented and easily accessible set of reference data that can be used to compare competing methodologies.

Machine learning research in molecular biology has shown two further areas where convergence between technologies are yielding clear results. In the first, Shavlik et al. (1992) developed a method for automatically extracting a comprehensible description of what has been learned from a trained artificial neural network. This means that in future, the rules learned by a neural network will be easier to understand and incorporate into knowledge-based systems. The second example of hybrid or converging technologies is the development of combinations of signal processing (e.g. statistical methods) and machine learning techniques. This is how the GRAIL system of Mural et al. (1992) predicts protein coding regions in DNA. Using a similar approach, Craven and Shavlik (1993) showed that a neural network could learn how to combine the results from the best pattern-based and statistical gene recognition programs to achieve a better result than any of them did individually.

An important goal for researchers who cross the boundaries of machine learning and natural language understanding is to be able to learn a grammar from examples of the language. Clearly, the possibility of learning detailed genetic grammars directly from molecular sequences must be a very long-term goal. However, the use of a combination of machine learning and genetic grammars to identify undocumented genetic elements in databases (Aaronson et al., 1993) is an important result for grammar induction techniques that complements the continued developments of more sophisticated grammars for representing biological structure and function in sequence data.

For AI techniques to be fully integrated into molecular biology computing, it is important that they be able to build systems that are efficient enough to reason with raw data and assist directly with the interpretation of scientific data. Constraint-based systems have many advantages when it comes to solving large scale application problems, and an important recent development in

constraint handling technology has been the convergence of logic programming, operations research and other problem solving techniques in constraint logic programming languages. Several recent examples of the application of CLP—protein topology prediction (Clark et al., 1993), prediction of RNA structure (Heuze, 1989) and the assembly of large scale physical genetic maps (Clark et al., 1994)—have demonstrated that this technology is sufficiently general and powerful to address a wide range of problems efficiently. Ongoing developments in CLP technology will link CLP languages with deductive databases to produce constrained deductive databases which should be well suited to many more problems in molecular biology.

To conclude, we have drawn on a selection of recent research to demonstrate the breadth of techniques and problems that have been addressed in AI and molecular biology. There are now signs that, through the adoption of an increasingly pragmatic approach and the integration with more traditional software technologies, important practical results are being achieved, and that AI has the potential to make a significant contribution to computational molecular biology.

Acknowledgements

We would like to thank Dominic Clark, Catherine Hearne and Simon Parsons for their comments on earlier drafts of this paper.

References

Aaronson, JS, Haas, J and Overton, GC, 1993. "Knowledge discovery in GenBank". In: L Hunter, D Searls and J Shavlik (eds.), *Proceedings of First International Conference on Intelligent Systems for Molecular Biology*, pp 3–11. AAAI Press, WA.

Abarbanel, RM, Wieneke, PR, Jaffe, DA and Brutlag, DL, 1984. "Rapid searches for complex patterns in biological molecules". *Nucleic Acids Research* **12** 263–280.

Baehr, A, Dunham, G, Ginsburg, A, Hagstrom, R et al., 1992. *An integrated database to support research on Escherichia coli*. Technical Report ANL-92/1, Argonne National Laboratory.

Bairoch, A, 1991. "PROSITE: a dictionary of sites and patterns in proteins". *Nucleic Acids Research* **19** 2241–2245.

Barton, GJ and Rawlings, CJ, 1990. "A PROLOG approach to analysing protein structure". *Tetrahedron Computer Methodology* **3** 739–756.

Bergadano, F and Gunetti, D, 1994. "Learning relations and logic programs". *The Knowledge Engineering Review* **9** (1) 73–77.

Blum, RL, 1982. "Discovery, confirmation and incorporation of causal relationships from a large time-oriented clinical database: the RX project". *Computers in Biomedical Research* **15** 164–187.

Brunak, S, Engelbrecht, J and Knudsen, S, 1991. "Neural network detects errors in the assignment of mRNA splice sites". *Nucleic Acids Research* **18** 4797–4801.

Brutlag, DG, Galper, AR and Millis, DH, 1991. "Knowledge-based simulation of DNA metabolism: prediction of enzyme action". *Computer Applications in the Biosciences* **7** 9–19.

Carhart, RE, Cash, HD and Moore, JF, 1988. "StratGene: object-oriented programming for molecular biology". *Computer Applications in the Biosciences* **4** 205–212.

Cattell, RGG, 1991. *Object Data Management: Object-oriented and extended relational database systems*. Addison-Wesley.

Clark, DA, Doursenot, S and Rawlings, CJ, 1994. "Genetic map construction with constraints". In: *Proceedings 2nd International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, WA (in press).

Clark, DA, Rawlings, CJ, Barton, GJ and Archer, I, 1990. "Knowledge-based orchestration of protein sequence analysis and knowledge acquisition for protein structure prediction". In: *Proceedings AAAI Spring Symposium*, 28–32.

Clark, DA, Rawlings, CJ, Shirazi, J, Veron, A and Reeve, M, 1993. "Protein topology prediction through parallel constraint logic programming". In: L Hunter, D Searls and J Shavlik (eds.), *Proceedings of First International Conference on Intelligent Systems for Molecular Biology*, pp 83–91. AAAI Press, WA.

Clark, DA, Shirazi, J and Rawlings, CJ, 1991. "Protein topology prediction through constraint-based search and the evaluation of topological folding rules". *Protein Engineering* **4** 751–761.

Clocksin, WF and Mellish, CS, 1981. *Programming in Prolog*. Springer-Verlag.

Craven, MW and Shavlik, JW, 1993. "Learning to predict reading frames in E. coli sequences". In: L Hunter (ed), *Proceedings of 26th Hawaii International Conference on Systems Science—Biotechnology*, pp 773–782. IEEE Computer Society.

Doursenot, S, Clark, DA, Rawlings, CJ and Veron, A, 1993. "Contig mapping using ElipSys". In: *Proceedings of AI and the Genome Workshop, 13th International Joint Conference on Artificial Intelligence* Chambery, France.

Edwards, P, Sleeman, D, Roberts, GCK and Yun-Lian, L, 1993. "An AI approach to the interpretation of the NMR spectra of proteins". In: L Hunter (ed), *AI and Molecular Biology*, pp 396–432. AAAI Press.

Friedland, P and Kedes, LH, 1985. "Discovering the secrets of DNA". *Communications of the ACM* **28** 1164–1186.

Friedland, P and Iwasaki, Y, 1985. "The concept and implementation of skeletal plans". *Journal of Automated Reasoning* **1** 161–208.

Friedland, P, Kedes, L, Brutlag, DL, Iwasaki, Y and Bach, R, 1982. GENESIS: a knowledge based genetic engineering simulation system for representation of genetic data and experiment planning". *Nucleic Acids Research* **10** 323–340.

Couy, M, 1989. "Secondary structure prediction of RNA". In: MJ Bishop and CJ Rawlings (eds), *Nucleic acid and protein sequence analysis Practical Approach*, pp 259–284. IRL Press.

Gray, PMD, Paton, NW, Kemp, GJL and Fothergill, JEF, 1990. "An object-oriented database for protein structure analysis". *Protein Engineering* **3** 235–243.

Hagstrom, R, Michaels, GS, Overbeek, R, et al., 1992. *GenoGraphics for Openwindows*. Technical Report ANL-92/11, Argonne National Laboratory.

Helgeson, C and Sibbald, PR, 1993. "PALM—a pattern language for molecular biology". In: L Hunter, D Searls and J Shavlik (eds), *Proceedings of First International Conference on Intelligent Systems for Molecular Biology*, pp 172–180. AAAI Press, WA.

Heuze, P, 1989. *RS2P RNA secondary structure prediction in ElipSys*. Technical Report ElipSys/10, European Computer-Industry Research Centre, Ababellastrasse 17, D8000 Munich 81, Germany.

Hirst, J and Sternberg, MJE, 1992. "Prediction of the structural and functional features of protein and nucleic acid sequences by artificial neural network". *Biochemistry* **31** 7211–7218.

Hunter, L, 1993. *Artificial Intelligence in Molecular Biology*. AAAI Press/MIT Press.

Hunter, L, Searls, D and Shavlik, J, (eds), 1993. *Proceedings of First International Conference on Intelligent Systems for Molecular Biology*. AAAI Press.

Jiang, K, Zheng, J, Higgins, SB, et al., 1990. "A knowledge-based experimental design system for nucleic acid engineering". *Computer Applications in the Biosciences* **6** 205–212.

Karp, P, 1992. "A large knowledge-base of bacterial genes and metabolism". In: *Proceedings of AAAI Workshop on Communicating Scientific and Technical Thinking*, pp 133–137. AAAI Press.

Karp, P, 1993. "A qualitative biochemistry and its application to the regulation of the tryptophan operon". In: L Hunter (ed), *AI and Molecular Biology*, pp 289–323. AAAI Press.

Karp, P and Riley, M, 1993. "Representations of metabolic knowledge". In: L Hunter, D Searls and J. Shavlik (eds), *Proceedings of First International Conference on Intelligent Systems for Molecular Biology*, pp 207–215. AAAI Press, WA.

Kazic, T, 1993. "Representation, reasoning and the intermediary metabolism of Escherichia coli". In: L Hunter (ed), *Proceedings of 26th Hawaii International Conference on Systems Science – Biotechnology*, pp 853–862. IEEE Computer Society.

Koile, K and Overton, GC, 1989. "A qualitative model for gene expression". In: *Proceedings of the 1989 Summer Computer Simulation Conference*. Society for Computer Simulation.

Koton, PA, 1985. *Towards a Problem Solving System for Molecular Genetics*. MIT Laboratory of Computer Science, Technical Report MIT/LCS/TR-338.

Kumar, V, 1992. "Algorithms for constraint satisfaction problems: a survey. *AI Magazine* **13** 32–44.

Lathrop, R, Webster, TA and Smith, TF, 1987. "ARIADNE: Pattern-directed inference and hierarchical abstraction in protein structure". *Communications of the ACM* **30** 909–921.

Lenat, DB, 1983. "The role of heuristics in learning by discovery: Three case studies". In: RS Michalski, JG Carbonell and TM Mitchell (eds), *Machine Learning: an artificial intelligence approach*, pp 243–306. Tioga Press.

Letovsky, S and Berlyn, MB, 1992. "CPROP: A rule-based program for constructing genetic maps". *Genomics* **12** 435–446.

Lindsay, RK, Buchanan, BG, Feigenbaum, EA and Lederberg, J, 1980. *Applications of Artificial Intelligence for Organic Chemistry: the DENDRAL project*. McGraw-Hill.

Lyall, A, Hammond, P, Brough D and Glover, D, 1984. "BIOLOG–a DNA sequence analysis system in Prolog". *Nucleic Acids Research* **12** 633–642.

Major, F, Turcotte, M, Gautheret, D, Lapalme, G and Cedergren, R, 1991. "The combination of symbolic and numerical computation for three-dimensional modeling of RNA". *Science* **253** 1255–1260.

Mavrovouniotis, ML, 1993a. "Identification of qualitatively feasible metabolic pathways". In: L Hunter (ed), *AI and Molecular Biology*, pp 325–364. AAAI Press.

Mavrovouniotis, ML, 1993b. "Identification of localized and distributed bottlenecks in metabolic pathways". In: L Hunter, D Searls and J Shavlik (eds), *Proceedings of First International Conference on Intelligent Systems for Molecular Biology*, pp 275–283. AAAI Press.

Meyers, S and Friedland, P, 1984. "Knowledge-based Simulation of Genetic Regulation in Bacteriophage lambda". *Nucleic Acids Research* **12** 1–9.

Minsky, M and Papert, S, 1969. *Perceptrons*. MIT Press.

Mott, R, Grigoriev, A, Maier, E, Hoheisel, J and Lehrach, H, 1993. "Algorithms and software tools for ordering clone libraries: applications to the mapping of the genome of Schizosaccharomyces pombe". *Nucleic Acids Research* **21** 1965–1974.

Muggleton, S and Feng, C, 1990. "Efficient induction of logic programs". In: S Arikawa, S Goto, S Ohsuga and T Yokomosi (eds), *Proceedings 1st Conference on Algorithmic Learning Theory*, pp 368–381. Japanese Society for Artificial Intelligence.

Muggleton, S, King, RD and Sternberg, MJE, 1992. "Protein secondary structure prediction using logic". *Protein Engineering* **5**, 647–657.

Mural, RJ, Einstein, JR, Guan, X, Mann, RC and Uberbacher, EC, 1992. "An artificial intelligence approach to DNA sequence feature recognition". *Trends Biotechnol* **10** 66–69.

Overton, GC, Koile, K and Pastor, J, 1990. "GeneSys: A knowledge management system for molecular biology". In: G Bell and T Marr (eds), *Computers and DNA SFI Studies in the Sciences of Complexity*, pp 213–239. Addison-Wesley.

Pereira, FCN and Warren, DHD, 1980. "Definite clause grammars for language analysis". *Artificial Intelligence* **13** 231–278.

Presnell, SR and Cohen, FE, 1993. "Artificial neural networks for pattern recognition in biochemical sequences". *Ann Rev Biophys Biomol Struct* **22** 283–298.

Qian, N and Sejnowski, TJ, 1988. "Predicting the secondary structure of globular proteins using neural network models". *J Molecular Biology* **202** 865–884.

Rawlings, CJ, Taylor, WR, Nyakairu, J, Fox, J and Sternberg, MJE, 1985. "Reasoning about protein topology using the logic programming language PROLOG". *Journal of Molecular Graphics* **3** 151–157.

Searls, DB and Liebowitz, S, 1990. "Logic grammars as a vehicle for syntactic pattern recognition". In: *Proceedings of Workshop on Syntactic and Structural Pattern Recognition*, pp 402–422. International Association for Pattern Recognition.

Searls, DB, 1993. "The computational linguistics of biological sequences". In: L Hunter (ed), *Artificial Intelligence in Molecular Biology*, pp 47–120. AAAI Press, CA.

Shavlik, JW, Towell, GG and Noordewier, MO, 1992. "Using neural networks to refine existing biological knowledge". *International Journal of Genome Research* **1** 81–107.

Stefik, M, 1978. "Inferring DNA structures from segmentation data". *Artificial Intelligence* **11** 85–114.

Stefik, M, 1981. "Planning with constraints [MOLGEN: Part 1]" *Artificial Intelligence* **16** 111–140.

Sternberg, MJE, King, RD, Lewis, RA and Muggleton, S, 1994. "Application of machine learning to structural molecular biology" *Phil. Trans. Roy. Soc. London (B)* (to appear).

Stormo, GD, Schneider, TD, Gold, L and Ehrenfuecht, A, 1982. "Use of the perception algorithm to distinguish translational initiation sites in E. coli". *Nucleic Acids Research* **10** 2997–3011.

Van Hentenryck, P, 1991. "Constraint logic programming". *The Knowledge Engineering Review* **6** 151–194.

Weiss, SW and Kulikowski, CA, 1991. *Computer Systems That Learn*. Morgan-Kaufman.

Weld, DS, 1984. *Switching between discrete and continuous process models to predict genetic activity*. MIT Artificial Intelligence Laboratory. Technical Report 793.

Yoshida, K, Smith, C, Kazic, T, et al., 1992. "Toward a human genome encyclopedia". In: *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp 307–319. ICOT.