CHAPTER

6

Integrating AI with Sequence Analysis

Richard Lathrop, Teresa Webster, Randall Smith, Patrick Winston & Temple Smith

1 Introduction

This chapter will discuss one example of how AI techniques are being integrated with, and extending, existing molecular biology sequence analysis methods. AI ideas of complex representations, pattern recognition, search, and machine learning have been applied to the task of inferring and recognizing structural patterns associated with molecular function. We wish to construct such patterns, and to recognize them in unknown molecules, based on information inferred solely from protein primary (amino acid) sequences. Besides its intrinsic interest as a difficult machine learning task of induction from complex and noisy data, this is of interest in the empirical domain for:

- suggesting targets for genetic manipulation in known molecules;
- suggesting functional identification and confirmatory tests in unknown or

newly discovered molecules; and

• increasing general scientific knowledge by suggesting essential structural elements encoding molecular function.

The work described in this chapter is part of a larger ongoing effort to associate symbolic structural patterns with functionally defined classes of proteins. The basic question that we seek to address is: How can we recognize and relate protein function and structure? In the cases of interest to us, one typically has a defining set of sequences (instances) exhibiting the structure or function of interest, plus some biological or chemical experimental data which is believed to be relevant. The task is to inductively construct the pattern(s) which detect regularities implicit in the set of defining sequences, and which discriminate them from all other sequences. In machine learning terms, this is a task of concept acquisition from positive and negative examples. Positive examples are sequences which exhibit the structure or function under study, and negative examples are sequences which do not.

A pattern-based model is an excellent starting point for a feedback loop between experimental testing of the model and model refinement. For example, the pattern-based modeling of the classical mononucleotide binding fold (MBF) in tRNA synthetases [Webster *et al.* 1987] and the simian virus 40 (SV40) and polyomavirus large tumor (large-T) antigens [Bradley *et al.* 1987] led to site-directed mutagenesis in SV40 at the site suggested by the pattern match. The experimental manipulation [Bradley *et al.* 1987] verified the MBF location there. Continued theoretical study [Figge *et al.* 1988] led to a common pattern in the SV40 large-T, E1A, and *myc* oncoproteins (cancer-related proteins), all of which co-transform cells (induce cancer-like growth) with *ras.* Experimental work [Grasser *et al.* 1988, DeCaprio *et al.* 1988, 1989] found that phosphorylation of the region matched by the pattern in SV40 large-T was associated with binding the retinoblastoma (Rb) protein (a protein that apparently suppresses cancer-like cell division, unless it is bound and inactivated). The region matched by the same pattern in E1A [Figge *et al.* 1988] was experimentally substituted for the region matched in SV40 large-T, and the resulting domain exchange was found experimentally to bind Rb [Moran 1988]. Experimental work confirmed that E1A also binds Rb [Whyte *et al.* 1988, Lillie and Green 1989]. A generalization of the pattern matched two additional proteins, E7 and CDC25, suggesting their Rb binding [Figge and Smith 1988]. Subsequent experimental work [Storey *et al.* 1988, Goldsborough *et al.* 1989] demonstrated that the degree of pattern match (its differential similarity score) in papillomavirus (the virus responsible for warts) E7 was linked to the degree of biological activity. As predicted, further experimental work [Munger *et al.* 1989, Dyson *et al.* 1989, 1990] verified that E7 binds Rb. Theoretical attempts to further generalize the pattern led to the discovery of a new pattern [Zhu *et al.* 1990] for transcriptional

activators (a large class of proteins that bind to DNA and activate genetic transcription). In experimental work continuing the Rb binding studies, Breese *et al*. [1991] constructed a number of small (14 residue) peptides containing the sequence regions matched by the pattern, demonstrated that they bound Rb, and verified the secondary structure part of the pattern with circular dichroism measurements. Further theoretical work led to the proposal of a full three-dimensional model [Figge *et al*., submitted] of the binding site in proteins that bind Rb. This prediction now awaits experimental test.

The basis for the theoretical approach is that common functions often correlate with common protein structures, domain folding types, supersecondary structures and/or a few invariant or equivalent amino acids. This is true even for proteins with very different primary sequences. Biologically, a mutation that distorts a functionally important structure tends to produce an unviable organism and so tends not to propagate. Other mutations often have little effect and so are passed to offspring. Consequently, functionally important structure tends to be conserved, and functionally irrelevant structure tends to drift. If the functionally related sequences exhibit sufficient evolutionary diversity, a conserved functional "signal" may be distinguishable above the "noise" of mutational drift.

Unfortunately, although similar protein sequences generally indicate similar folded conformations and functions, the converse does not hold [Creighton 1983]. There are proteins, e.g., the nucleotide binding proteins [Rossman *et al*. 1974; Birktoft and Banaszak 1984], in which the secondary and tertiary structure encoding a common function is conserved while primary sequence similarity is almost non-existent. Methods which detect similarities solely at the primary sequence level have difficulty addressing functional associations in such sequences. A number of features, often only implicit in the protein's primary sequence of amino acids, are important in determining structure and function.

We attempt to identify patterns which are characteristic of a structural motif assumed to carry out some particular function. Our approach involves searching for a pattern which is shared by a defining set of functionally related proteins (positive instances of the function), and which does not appear in other proteins (negative instances, comprising the control set). The features we employ can be predicted or inferred (even if only statistically) from the primary structure. An initial "complex pattern" of a structural motif potentially involved in carrying out the common function is iteratively refined in order to maximize its discrimination between the positive and negative instances. The resulting pattern is a preliminary model for the relationship between a protein's structure and function, grounded in its amino acid sequence, which includes an identification of a functional site(s) as well as structural elements characteristic of that function. In the machine learning literature, this is sometimes referred to as the "concept" associated with the

positive set. Regions of protein sequences matched by the pattern may suggest potential sites for experimental work. Because the inference is grounded in the primary sequence, the model is also suitable for hypothesizing the function in unknown proteins for which the primary sequence may be the only information available.

We have decomposed our pattern-based approach into a series of subproblems, each addressing a small part of the puzzle:

(1) ARIADNE [Lathrop *et al*. 1987] approaches our basic question by searching an annotated protein sequence for a complex pattern describing a structural motif assumed to be correlated with function. It employs hierarchical object-oriented representations of both pattern and protein, graph-based sequence and annotation representations, procedural attachment of user-defined match functions, and a complex user-extensible structural pattern language that supports pattern element weights, gaps, annotations and weights attached to the annotations, and so forth.

(2) The ability to match an unknown protein sequence against a complex pattern introduces the question of: Where does the complex pattern come from? ARIEL [Lathrop 1990, Lathrop *et al*. 1990, 1992] functions as an "Induction Assistant" for the biologist engaged in constructing such patterns. It applies massively parallel symbolic induction to the task of refining an initial seed pattern to increase the discrimination between positive and negative instances. Machine learning heuristics for the main pattern language components have been implemented on a massively parallel computer. These manipulate class membership terms, interval relationships, pattern element weights, and an overall match threshold. The time complexity of these machine learning heuristics techniques is essentially constant, and their space complexity essentially linear, in the number of instances in the training set.

(3) But now the further question arises: Where does the seed pattern come from in the first place? PIMA [Smith and Smith 1990, 1992] inductively constructs primary sequence patterns common to a functionally-related family of proteins, using a modified dynamic programming alignment method. These patterns can be more diagnostic for functional family membership than using any member of the family as a probe sequence.

The chapter begins with a background section, following which we examine each of these three areas in detail. No discussion of pattern-based sequence analysis is complete without some mention of statistical reliability, and we close the chapter with a brief discussion of this important topic.

2 Background

One of the fundamental problems in molecular biology is that of relating a protein's structure and function to its amino acid sequence. Kolata [1986] terms this difficult problem "cracking the second half of the genetic code".

Among the problems involved are: relating the protein primary sequence to higher structural levels of protein organization (including secondary, super-secondary, domain, tertiary and occasionally multi-protein quaternary structural levels); identifying the structural elements which are the determinates of a protein function (structural elements here refers broadly to all levels of protein structure); describing the organizational constraints (or patterns) that hold between such elements; and inferring the function and structure in new or unknown proteins.

Before 1959 it was generally assumed that every unique protein sequence would produce a three-dimensional structure radically different from every other protein structure. The subsequent accumulation of X-ray determined protein structures has shown that proteins tend to have a limited number of three-dimensional arrangements. Also, proteins with similar functions often have similar structure. It is this regularity of protein structure that makes it possible to investigate the relationships between sequence, encoded structure, and biological function.

The problem is, of course, difficult and complex. Approaches fall into four broad categories, two experimental and two analytical:

(1) Analysis of physical data generated from proteins, such as X-ray and nuclear magnetic resonance (NMR) data. The most rigorous way to connect primary sequence to function is through X-ray analysis of co-crystal structures, which provide a three-dimensional picture of the protein molecule bound to its substrate. Unfortunately the availability of such data is quite limited. For most proteins, crystals suitable for X-ray analysis prove difficult to obtain, and the experimental and analytical process may require years to determine a single structure (where possible at all). NMR is currently possible only for small proteins, due to resolution limitations inherent in the experimental techniques.

(2) Genetic and crosslinking experimental studies which highlight potentially important functional regions. Functional change associated with amino acid substitutions, deletions and insertions can correlate amino acid positions or regions with functional determinates. The crosslinking of substrates to nearby amino acids supports their association with binding sites. These are often the result of sophisticated laboratory techniques, and the data is usually difficult and laborious to obtain.

(3) Prediction of protein three-dimensional structure and/or function directly from the primary sequence. There are three major approaches. The first (3a) is to attempt to predict three-dimensional backbone conformation by attempting to fold the protein sequence directly from first principles. These methods generally employ empirical potential energy functions. They are very computationally intensive and currently rather unsuccessful, although an active area of research with hope for the future. The second approach (3b) is to predict structure based on similarity to a known three-di-

mensional structure. This approach is related to the first in using empirical potential energy functions, but uses them to refine an initial fit obtained from a known structure rather than to fold the chain *ab initio*. Success depends on the degree of similarity between the known and modeled structures. If the similarity is sufficiently great this method can give very good results, but many sequences fail to exhibit appreciable primary similarity to any known structure. The third (3c) comprises a wide variety of secondary and tertiary structure prediction schemes that attempt to use empirical or statistical rules of one form or another.

(4) Comparative analysis of primary sequences (or of physical values inferred from the primary sequences). There are two related approaches. The first (4a) is to compare primary sequences directly to each other. If significant similarity occurs between a protein of interest (the query) and a protein of known structure or function, one can reason by analogy that they share similar structure and function. If the similarity is great enough this inference is almost always correct. The second (4b) is to compare primary sequences to a structural or functional pattern. If a match to a known pattern occurs one can infer that the protein of interest shares the structure or function associated with the pattern. While a pattern-based approach has often been shown to be a more sensitive detector than direct primary sequence similarity to any single query sequence, the validity of the inference depends on the sensitivity and specificity of the pattern.

Others might reasonably classify some approaches differently, as many overlap or share characteristics. Some of the references below touch several of these areas and have been arbitrarily categorized. In any case our intent is not a rigorous ontological division of the field, but only a pedagogical aid to structure the presentation.

Approach (1) and (2) above rely on experimental methods that are outside the scope of this chapter, even though computational methods are often crucial in the interpretation of the experimental data. For example, the interpretation of X-ray crystal diffraction data to yield the three-dimensional placement of atoms is extremely computationally intensive. Hayes-Roth *et al*. [1986] have applied AI constraint-based techniques to the problem of inferring protein structure from NMR experimental data; Glasgow, *et al.* and Edwards, *et al.* in this volume also describe AI systems that address these approaches.

Approaches (3) and (4) rely almost completely on the development and implementation of analytical and computational methods. Most of the methods that have been developed for approach (3) apply to attempts to model the three-dimensional placement of atoms in the protein, either numerically or by assigning a qualitative structural class to the sequence or subsequences. While quantum mechanics provides a solution in principle to this problem, it is impractical for molecules such as proteins which may contain many thou-

sands of atoms. Consequently a variety of clever innovations have been used instead. Most of these are beyond the scope of this chapter and its focus on sequence analysis, although some of the methods employed for (3c) use pattern-based or machine learning techniques. Approach (3c) is described elsewhere in the present volume, e.g. in the chapters of Holbrook *et al*., and Hunter. Zhang *et al*. [1992] applied parallel processing machine learning methods to find regularities in three-dimensional protein structure. A few examples of related techniques include Bohr *et al*. [1990], Bowie *et al*. [1991], Clark *et al*. [1990], Cohen *et al*. [1989], Fasman [1989], Holbrook *et al*. [1990], Holley and Karplus [1989], King and Sternberg [1990], Maclin and Shavlik [1992], Major *et al*. [1991], McGregor *et al*. [1989], Muskal *et al*. [1990], Noordewier *et al*. [1990], Ponder and Richards [1987], Qian and Sejnowski [1988], Rawlings *et al*. [1985], Thornton *et al*. [1991], and Towell *et al*. [1990].

Comparative sequence analysis, approach (4), will be the focus of the remainder of the chapter. These methods proceed by comparing a sequence either to another sequence or to a pattern. Although this chapter is primarily directed towards protein sequences, many of the computational techniques are equally applicable to both protein sequences (strings from an alphabet of twenty letters) and DNA sequences (strings from an alphabet of four letters).

2.1 Comparing Primary Sequences to Each Other

By far the most common approach to relating a protein's amino acid sequence to its structure and function is by comparing its sequence to one or more known protein sequences [Wilbur and Lipman 1983; Pearson and Lipman 1988; Altschul *et al*. 1990]. Many important advances have been made by these methods, for example, when the sequence of an oncogene (cancer related gene) was found to be similar to sequences of human growth hormones [Doolittle *et al*., 1983]. The highly similar sequences clearly related cancerous growth to defective normal cell growth.

The basic idea of most sequence comparison algorithms is to obtain a measure of the similarity (or distance) between two sequences. This usually reflects the minimum number of changes ("edit distance") required to convert one sequence into the other [Sellers 1974, Smith and Waterman 1981a,b]. For biological sequences, there are basically three types of mutation events commonly counted: point mutations, deletions, and insertions. These are "nature's typos:" NATORE, NTURE, and NATEURE. An alignment of such sequences is defined as an ordered sequence of n-tuples, each n-tuple containing one element, or null, from each sequence. For example:

```
N A T - U R E
N A T - O R E
N - T - U R E
N A T E U R E
```

is one alignment of these sequences. Alignments are usually constructed so as to maximize the measure of similarity (or minimize distance) between the sequences. For a few examples of related techniques, see Bacon and Anderson [1986], Barton [1990], Barton and Sternberg [1987a,b], Brutlag *et al*. [1990], Corpet [1988], Felsenstein [1988], Feng and Doolittle [1987], Fischel-Ghodsian *et al*. [1990], Hein [1990], Henneke [1989], Hunter *et al*. [1992], Sankoff and Kruskal [1983], Schuler *et al*. [1991] Taylor [1988a], Vingron and Argos [1989], Waterman [1984, 1986]. Parallel processing versions have been implemented by Collins *et al*. [1988] and Lander *et al*. [1988].

2.2 Comparing Primary Sequences to Patterns

Inspecting the four aligned sequences above, one might notice that their observed variability could be concisely represented by "NgTgvRE", if we assume that "g" (gap) matches zero or one characters of any type, "v" matches any one vowel, and each upper-case letter matches exactly itself. While direct sequence comparison often yields important information, in some cases it may be more desirable to derive a pattern representing the structure or function under study and then compare sequences to that pattern. This is because a pattern is often a more sensitive detector of the regularity under study than any single sequence, due to the "noise" in the rest of the sequence. Further, elements of the pattern often highlight biologically important aspects of the protein.

What is required to compare a pattern to a protein? We must: (1) represent the protein and the pattern to the computer; (2) have an algorithm which performs the comparison; and (3) somehow obtain a pattern to compare. These are all closely related, of course, but we shall adopt this division as an organizing theme.

For pattern matching purposes, the simplest protein representation is a linear sequence denoting its amino acids. This basic amino acid sequence is sometimes annotated with additional information, representing additional features (known or inferred) of the sequence. The degree of annotation possible is a function of the level of our knowledge.

At a minimum, most useful patterns must be able to represent protein positions in which any of several alternate amino acids are acceptable (amino acid physico-chemical classes), as well as regions in which a variable number of amino acids may occur (variable-length gaps). Beyond this, the ability to tolerate a certain amount of mismatch to a pattern lends robustness in the face of mutational diversity. Weights or frequencies are often used to specify greater tolerance in some positions than in others. There is a great deal of effort in the field aimed at extending the power, flexibility, and expressive power of patterns beyond these simple desiderata. Protein sequences fold up to form complex dynamic mechanisms, in which mutations, interactions and dependencies abound. Representations which capture in a manageable way the complexity

inherent in Nature may expose some of her regularities more clearly.

The simplest pattern match algorithm possible is an exact match to a literal string. This fails to handle most of the naturally occurring variability in biological sequences. The necessary robustness for inexact matches can often be supplied by the pattern match algorithm instead of the pattern itself. For example, regular expression-based patterns (which specify an exact match in the usual finite state machine construction) can be made more robust by a match algorithm which allows some mis-matches before discarding a potential match.

In some cases the pattern to compare may come directly from biochemical investigation, known three-dimensional structures, analysis of genetic or mutational data, knowledge of similar sequences or patterns, and other sources. Such information is not often available in sufficient quantity and quality to form the sole basis for pattern construction, although it may be adequate to provide initial guesses or seed patterns. Consequently, inductive construction of the pattern is often necessary. The simplest pattern discovery method is to align the sequences maximizing the number of matching amino acids, then construct a consensus sequence from the conserved regions by assigning each consecutive pattern position to consecutive aligned sequence positions. The pattern above, "NgTgvRE", was constructed in this way. This simple method may fail to find patterns in defining sets having widely diverse primary sequences, and consequently more sophisticated approaches are often desirable. Pattern induction techniques fall broadly into two classes, depending on whether a sequence alignment is performed to bring sequence positions into explicit correspondence with each other before pattern discovery is attempted, or not. There is an intermediate set of techniques for which sequence alignment and pattern discovery proceed in alternating cycles. A number of the approaches are "semi-automatic" in actual domain practice, the domain expert applying domain knowledge by direct manual intervention where deemed appropriate or desirable. Any existing experimental data may be used, either as a source of additional clues in the pattern construction process, or to substantiate the pattern once discovered (for example, by verifying that the pattern elements and the positions of matches within the sequences reflect experimentally known associations between sequence position and function).

Hierarchical pattern-matching was pioneered by Abarbanel [1985] and Cohen *et al*. [1983, 1986], and these researchers originated the term "complex pattern". Cohen *et al*. [1991a,b] added a meta-level of control knowledge. Taylor and Thornton [1983] originated the use of secondary structure predictions and hydropathy (hydrophobicity) in super-secondary structure patterns. An explicit machine learning approach to pattern discovery in protein sequences was developed by Gascuel and Danchin [1986]. Automatic evaluation of functional patterns was described by Guigo *et al*. [1991].

Hodgman [1986] describes a pattern library. The chapter by Searls in the present volume discusses complex grammars for biosequences. A few related examples include Abarbanel *et al*. [1984] Barton [1990], Barton and Sternberg [1990], Blundell *et al*. [1987], Bork and Grunwald [1990], Boswell [1988], Cockwell and Giles [1989], Gribskov *et al*. [1987, 1988], Hertz *et al*. [1990], Lawrence and Reilly [1990], Myers and Miller [1989], Owens *et al*. [1988], Patthy [1987, 1988], Sibbald and Argos [1990a], Smith *et al*. [1990], Smith and Smith [1989], Staden [1989], Stormo [1990], Stormo and Hartzell [1989], Taylor [1986, 1988b], Thornton and Gardner [1989], Waterman and Jones [1990], and Webster *et al*. [1989].

Comparative sequence analysis has been an active and fruitful area for the application of computation to biological problems, and a number of very clever techniques have been devised. The discussion and references above provide only an initial window. Next we turn our attention to examining our approach to integrating AI techniques with existing domain methods for sequence analysis. The approach uses three systems: ARIADNE, which matches a complex pattern to an annotated protein sequence; ARIEL, which inductively constructs these complex patterns by refining one or more "seed" patterns; and PIMA, which constructs seed patterns given a family of proteins.

3 ARIADNE[1]

ARIADNE was developed to explore representation and match algorithm issues. Our motivation was to allow a more complex representation of protein sequences, in order make richer information sources explicitly available; and correspondingly, to provide a more complex pattern language in which to express similarities among proteins at a higher level than primary sequence. Because the "best" indicators of protein structure are surely not yet known, both protein and pattern representations had to be easily extensible. In turn, the matching algorithm had to be flexible enough to support unknown future extensions to the representations; extensible itself in order to easily support novel match behavior; and also efficient enough to quickly match complex patterns to large sets of protein sequences. The resulting system facilitates direct expression and manipulation of higher-order structures. It identifies the optimal match between a given complex pattern and protein sequences annotated with various inferred features, by abstracting intermediate levels of structural organization. Inference is grounded solely in knowledge derivable from the primary sequence.

A biologist first hypothesizes a possible protein structure, based on biochemical knowledge (for example, Figure 1a). This is used to form a pattern describing the hypothesized common features, as a sequence of primary sequence elements and their annotations (for example, Figure 1b). It is often convenient to be able to describe the pattern in terms of hierarchical group-

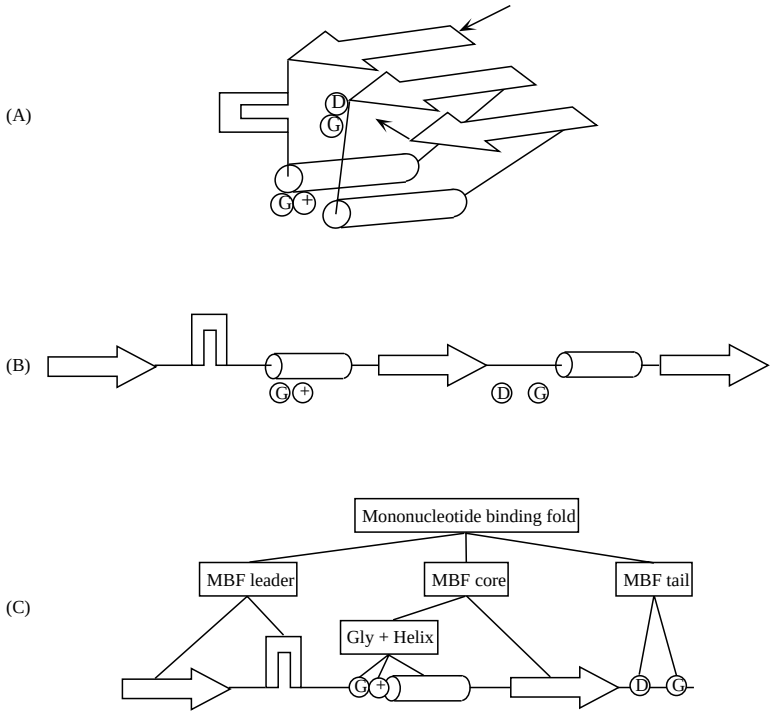


Figure 1 (a) Schematic of the Mononucleotide Binding Fold-like Structure. Beta-sheet strands are represented by arrows, alpha-helices by cylinders, and beta-turns by angular bends. (b) The Mononucleotide Binding Fold Unfolded into a Linear Sequence. The first beta-strand/beta-turn/alpha-helix/beta-strand sequence will form the basis of the structural descriptor below. Key amino acids have been labeled.. (c) The Unfolded Mononucleotide Binding Fold as Hierarchical Groupings. It is often convenient to be able to describe a structure in terms of intermediate levels. This figure appeared as figure 3 of Lathrop et al. (1987).

ings of sub-patterns (for example, Figure 1c). ARIADNE receives as input these pattern(s), and also one or more annotated protein primary sequences.

ARIADNE's biological structure knowledge is encoded in a number of pattern/action inference rules: an antecedent (pattern) that describes a relationship between structural elements, and a consequent (action) that executes in a context with variables bound to reflect the current state of the match (the consequent usually, but not always, hypothesizes the presence of a higher-order structure). Patterns are represented as a hierarchy of sub-patterns, each level an inference based on sub-patterns at lower levels. The target protein is searched for regions which are plausibly similar to the rule antecedent. A dif-

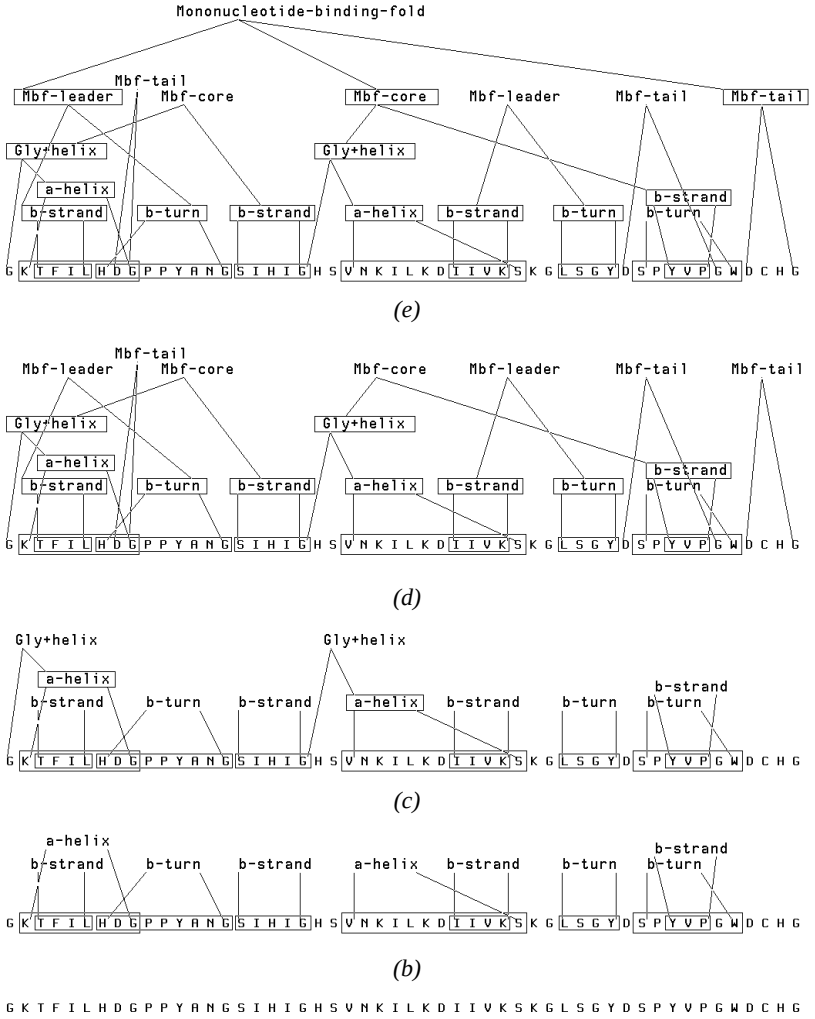


Figure 2 (a) E. coli *Isoleucyl-tRNA synthetase (residues 48-99 of 939 residues). Primary sequence input to ARIADNE. (b)* E. coli *Isoleucyl-tRNA synthetase (residues 48-99 of 939 residues). Secondary structure predictions (Chou and Fasman, 1978; Ralph et al. 1987) input to ARIADNE. (c)* E. coli *Isoleucyl-tRNA synthetase (residues 48-99 of 939 residues). Intermediate predictions constructed by ARIADNE. (d)* E. coli *Isoleucyl-tRNA synthetase (residues 48-99 of 939 residues). Intermediate predictions constructed by ARIADNE. (e)* E. coli *Isoleucyl-tRNA synthetase (residues 48-99 of 939 residues). Final prediction constructed by ARIADNE. No other occurrences of Mononucleotide-Binding-Fold are predicted in this sequence. This figure appeared as figure 4 of Lathrop et al. (1987).*

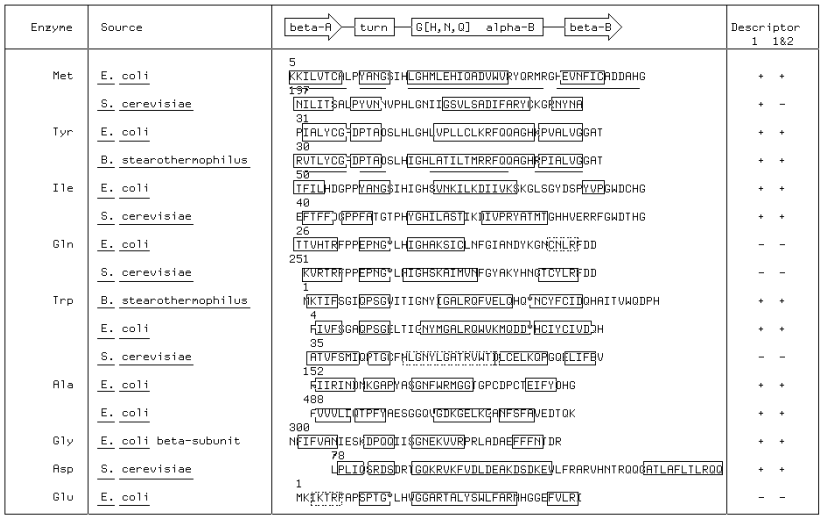


Figure 3. Proposed alignment of aminoacyl-tRNA synthetase sequences with the first beta-alpha-beta fold of a mononucleotide bindinglike structure. The regions predicted to fold into beta-A, the turn, alpha-B, and beta-B are enclosed in the first, second, third and fourth solid box*, respectively, of each sequence. Sequences which match the composite descriptor and/or descriptor 1 are indicated with a "+" to the right of the figure. Secondary structure assignments for the* B. stearothermophilus *Tyr-tRNA synthetase and* E. coli *Met-tRNA synthetase X-ray structures are indicated with an* underline*.* Dashed boxes *indicate weakly predicted secondary structure elements. This figure is adapted from figure 5 of Webster et al. (1987), which includes references for the sequences.*

ferential similarity score between the pattern and the target protein elements is computed, and the rule fires where this equals or exceeds some user-specified threshold. When the rule fires, its consequent typically creates a new entry in the overlay of inferred structures. The new entry can enable the firing of subsequent rules. For example, Figure 2a-e shows the hierarchical pattern matching of the mononucleotide binding fold pattern (Figure 1) to an annotated protein sequence.

The output describes, for each input protein, which (sub)pattern(s) matched; what their support in the annotated protein sequence was; and the computed similarity score between the (sub)pattern(s) and the protein elements in the support. From the patterns and their match support, it is possible to construct a local alignment of the matched members of the family, by aligning elements which support the same pattern. For example, Figure 1a-c shows a pattern for a mononucleotide binding foldlike structure, and Figure 3

shows the local alignment induced by the matches of the pattern to the positive sequences. If the the input sequences are divided into the positive (defining) and the negative (control) sets, it is also possible to obtain the sensitivity/specificity spectrum of the pattern as the threshold similarity score varies. These take the form of 2x2 contingency tables (see the section on Significance, Validity, and Pattern Quality, below) for each possible setting of the threshold score, as well as a graph summarizing the spectrum of sensitivity and specificity attainable as the threshold varies.

3.1 ARIADNE Protein Representation

We enriched the protein sequences by explicitly annotating the sequences with information which normally is only implicit in the sequence. For example, secondary structure predictions [Chou and Fasman, 1978; Garnier *et al*. 1978], hydropathy and amphipathy weighted average profiles [Eisenberg *et al*. 1982, 1984], and various charge templates and clusters [Karlin *et al*. 1989, Zhu *et al*. 1990], all contain clues to higher levels of organization. All are implicit in, but can be inferred from, the amino acid sequence. By explicitly annotating the protein sequence with these implicit information sources (often computed externally by existing domain programs), we can make the additional structural clues they contain directly available to the pattern match process. For example, Figure 2b shows a protein sequence annotated with secondary structure predictions [Ralph *et al*. 1987]. Any information source which usefully distinguishes certain subsequences of the protein may be employed.

Although the amino acid sequences are inherently strings in a twenty-letter alphabet, we represent the instance as an object-oriented directed graph. Nodes and arcs are the typed and weighted data objects. Typed nodes in our graph represent the twenty amino acid types found in protein sequences. They participate in a user-extensible class generalization hierarchy depending on amino acid physico-chemical properties (volume, charge, hydropathy, and so forth). Directed arcs annotate the sequence with extended higher-level features. The arc type reflects the feature type, and the arc connects the node that begins the feature to the node that ends it. Arcs represent the annotations to the protein sequence supplied in the input data, as well as the higher-level structural organization inferred as a result of pattern matching. For example, nodes represent the protein sequence shown in Figure 2a, while arcs represent both the input secondary structure predictions shown in Figure 2b and the subsequent pattern matches shown in Figure 2c-e.

This approach allows a uniform treatment of information derived from both external and internal sources. Both are treated as annotations to the protein sequence, which make explicit some information previously implicit in the sequence but inferrable from it. This is similar in some respects to a "blackboard" architecture [Erman and Lesser 1975]. The hierarchical organi-

zation of protein structure in the domain is readily mirrored, and hierarchical patterns are easily supported. Because each extended feature represents its own beginning and end, overlapping features of the same type are distinct. This facilitates representation of the annotated structural correlates and inferred structures. The major alternative approach (in a purely string-based sequence representation) is to mark the individual string characters with tokens indicating which feature types they occur within; but then overlapping features of the same type lose their boundaries and hence their individual identities, inseparably blurring together.

3.2 ARIADNE Pattern Language.

ARIADNE's pattern language is divided into primitive and composite pattern elements. Primitive patterns usually appear only as components in composite patterns. Their match behavior is completely determined by three attached procedures that directly inspect and manipulate the target protein data structures:

1. GENERATE-EXPANSIONS returns a list of possible pairings of the pattern element to target protein elements.
2. SIMILARITY-SCORE returns the similarity score that the pattern element actually attains on any given pairing to target protein elements.
3. MAX-POSSIBLE-SCORE returns (an upper bound on) the maximum similarity score the pattern element could ever achieve.

A number of useful primitive pattern elements are provided by default. These include the twenty primitive amino acids; their various physico-chemical classes (e.g., positively charged, hydrophobic, H-bond donors, etc.); several spacer (gap), overlap, positioning, and containment operators; features in various numeric transforms of the amino acid sequence (e.g., amphipathy peaks are often associated with surface helices); primitives for recognizing sequence annotations (including user-defined annotations); and so forth.

However, ARIADNE is predicated on the assumption that we do not yet know the best structural features with which to analyze all proteins. The intent is to provide a development framework wherein pattern primitives with complex behavior can easily be created in response to new needs, as domain knowledge expands through exploration and experimentation. ARIADNE therefore provides a framework of procedural attachment for the user to define new primitive pattern elements, with new match behavior. This is done by allowing the user (or more likely, a programmer working with the user) to write new procedures for the three determinates of primitive match behavior above. Also, any annotation type used to annotate the input protein sequence automatically induces a corresponding primitive element that recognizes occurrences of the annotation. Thus the pattern language is extensi-

ble either by defining new annotations, or by directly coding the match behavior of new primitives.

A composite pattern is defined by giving a list of the lower-level subpatterns and relationships required as support. Its match behavior is completely determined by the match behavior of its components. Hierarchical pattern construction and matching is supported because recognition of a hierarchical organization from low-level detail proceeds most naturally by hierarchical construction of the intervening patterns. This is implemented internally by annotating the protein to reflect the newly inferred structure, in exactly the same way as we annotated the protein on input to reflect structure inferred by domain programs. Each instance of a composite pattern, when recognized in a low-level description, becomes available as a feature element for higher-order composite patterns. In this way a pyramid of inferences may connect the low-level features to the more abstract.

3.3 ARIADNE Pattern Matching Algorithm.

The power of pattern-directed inference (e.g., rule-based expert systems) is well known, as is its applicability to molecular biology [Friedland and Iwaskai 1985]. One of the first such systems ever constructed, DENDRAL [Lindsay *et al*. 1980], also performed the task of chemical structure recognition. However, we allow flexible rule invocation based on a controllable degree of partial pattern similarity. This is implemented by an A* search [Winston 1984] through the space of target protein subsequences.

The search for a differential similarity to a composite pattern consists of attempting to pair each component subpattern to an admissible subset of target objects. A partial pairing, constructed at some intermediate stage, might pair only some of the pattern components. For a given composite pattern, ARIADNE's search space is the set of all possible partial pairings. The single start state in this search space is the empty partial pairing, and goal states are complete pairings of all pattern components. An operator which carries one partial pairing into its successors, is to expand the next unpaired pattern component by hypothesizing pairings to every admissible set of target objects. By applying this operator first to the start state and then iteratively to resulting partial pairings, all complete pairings may be found. Typically, a single new target object is created for each pair showing a positive similarity (Figure 2c-e). For example, in Figure 2c the pattern "Gly+helix" is shown matching its components "G" and "a-helix". Each time a complete pairing is found, a new "Gly+helix" object is created.

Viewed from top-to-bottom, the added target objects impose a hierarchical organization. Viewed from left-to-right they impose a lattice structure because of the partial ordering, "followed-by", inherited from the underlying linear chain. Pattern recognition consists of exploring alternate pathways through the lattice structure. For example, in Figure 2b the target object rep-

resenting the first lysine (the first "K" in "G K T F ...") may be followed either by a threonine object ("T") or by an object representing a beta-strand prediction. The beta-strand object, in turn, may be followed either by a histidine object ("H") or by a beta-turn object. This permits structural elements (at any level) to be manipulated and searched as a unit, independent of their actual length or composition.

Complete pairings are ordered by a similarity score and only the higher-scoring ones are of interest, so an efficient search strategy is desirable. The well known A* search [Winston 1984] efficiently accommodates differentially inexact similarities to a pattern and tends to focus search effort on the most promising candidates. A* is a best-first branch-and-bound search with dynamic elimination of redundant pairings and an optimistic estimate of the contribution of the remaining unpaired pattern components. (The elimination of redundant pairings may optionally be suppressed.) Optimality and convergence are both guaranteed.

The key to A* search is in the selection of which partial pairing to expand. Each partial pairing has a "best possible score", which is the highest score that the most favorable possible pairing of yet-unpaired pattern components could ever yield. At each step the partial pairing with the highest BEST-POSSIBLE-SCORE is selected. If its BEST-POSSIBLE-SCORE is below the threshold the search can fail immediately, as no partial pairing could possibly exceed the threshold. Similarly, if it is a complete pairing then no other partial pairing can ever complete to a higher score. Otherwise, its next unpaired pattern is expanded and the algorithm iterates. It is possible to enumerate all complete pairings in decreasing order of similarity score, pausing and continuing the search at will.

3.4 Discussion of ARIADNE

The principle sources of power in ARIADNE are:

- The ability to utilize multiple, unreliable, inconsistent knowledge sources. Since no prediction scheme produces accurate predictions, any inference procedure which vitally depended on the consistency of its database (e.g., some forms of theorem-proving) would be ineffective.
- The use of a pattern-similarity measure to guide flexible invocation of inference rules. This conveys a degree of robustness in the face of pattern fluctuations such as mutations.
- Implementation of the rule-invocation similarity measure as an A* search. This provides an efficient enumeration of match candidates, in order of decreasing similarity.
- A flexible framework for pattern language development and extension. This is important because all the appropriate pattern elements are surely

not yet known.

- Explicit identification and representation of the intermediate hierarchy, which helps in several ways:
  - Many of the higher-order (super-secondary) structures of interest are most effectively expressed in terms of lower and intermediate levels of hierarchy (secondary structure groupings), and not directly at the lowest level of description.
  - Handling patterns in small pieces encourages selective pattern refinement.
  - Expressing patterns consisting of key residues embedded in secondary structures involves the interaction of different hierarchical levels.
  - Breaking a large pattern into pieces increases search efficiency by reducing the potentially exponential time dependency on pattern size.

The approach presented here is limited to detecting similarities in sequence patterns of known and/or predicted structural elements. To the extent that hypotheses of interest can be expressed in the form of a structural pattern, ARIADNE provides a powerful and efficient vehicle for finding supporting regions in the target proteins. However, no use is currently made of primary sequence similarities (or homologies), which would provide additional evidence for favoring some matches over others (particularly similarity of sequence elements in between paired pattern elements across the defining set). No direct use is made of three-dimensional spatial constraints. The secondary structure predictions remain inherently inaccurate, even though trade-offs can be made between reliability and coverage. Some three-dimensional structural motifs are composed from elements widely dispersed in the primary sequence but folded to be contiguous in space, and these are unlikely to be seen by any method which draws its power from exploiting constraints which are local in the sequence. No attempt has been made to encode or exploit "expert rule-of-thumb" knowledge of general biochemical heuristics.

Construction of abstract structural hypotheses implies that low-level features meet the additional constraints imposed by higher-order patterns and relationships. These constraints take two forms: requiring a specified relationship with an element unambiguously present in the primary input (e.g., key amino acids); and requiring a specified relationship with other predicted or inferred features. Importantly, in a hierarchical pattern recognizer the structure imposed by higher-order patterns implies strong constraints on the admissibility and interpretation of low-level features, because those not fitting into a higher-level pattern will be dropped. A pattern acts to prune the (uncertain, heuristic, empirical) low-level features by selective attention, based on the strong constraint of fitting into higher-order organization (see

Figure 2a-e). Low-level features will be interpreted in terms of the expectations encoded in the patterns being searched for.

This has both good and bad aspects. When an intelligent agent (e.g., a biologist) hypothesizes and searches for the existence of a particular pattern based on supporting biochemical or circumstantial evidence, selective feature attention extends that evidential support down to low-level feature selection, and features supporting the pattern will be propagated upward. When a large number of patterns are sought randomly in a large number of targets (as in a database search), then each pattern will impose its own selective bias and additional confirming experimental evidence should be sought. In either case, an important estimate of the false positive (resp. false negative) rate may be had by testing a control set known not to (resp. known to) actually satisfy the pattern.

4 ARIEL^2

Although ARIADNE provides a powerful and flexible means to match a complex pattern against a group of protein sequences, it raises the question, "Where do the patterns come from?" One methodology for pattern construction by a domain expert was described in [Webster *et al*. 1988]. An initial pattern is refined in an iterative loop consisting of:

a. matching the pattern against the positive and negative instances;

b. evaluating the performance (sensitivity and specificity) of the pattern; and

c. modifying the pattern and repeating.

Even in cases where a clear idea of actual structure provides an initial seed pattern, transforming that into the final pattern that best separates the positive and negative sequences involves a potentially large search through pattern space. ARIEL [Lathrop 1990, Lathrop *et al*. 1990, 1992] functions as an "Induction Assistant" to the domain expert engaged in such a process. It automates and parallelizes parts (a), (b), and the low-level aspects of part (c), of the methodology above, while the domain expert provides high-level control and direction (see Figures 4 and 5). Symbolic induction heuristics ("operators") are provided for the major syntactic components of the pattern language. Various amino acid classes may be explored at primary sequence positions; inter-element intervals (gaps) may be varied; weights associated with the several pattern elements may be increased or decreased; the overall match threshold may be changed; and pattern elements may be added or deleted.

ARIEL runs on a CM-2 Connection Machine, and implements efficient massively parallel machine learning algorithms for several symbolic induction heuristics (of the sort familiar to the symbolic machine learning commu-

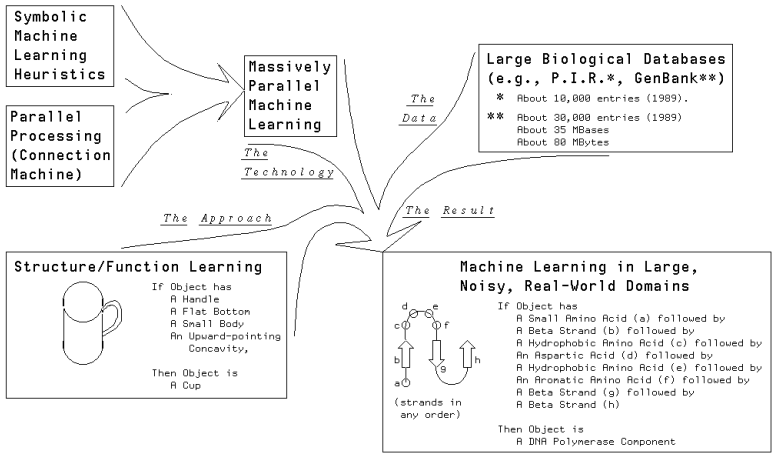


Figure 4 . Overview of ARIEL. Structure/function relationships, massively parallel processing, symbolic machine learning, and on-line biological databases converge in learning structure/function patterns as discussed in this chapter. This figure appeared as figure 1 of Lathrop et al. (1990).

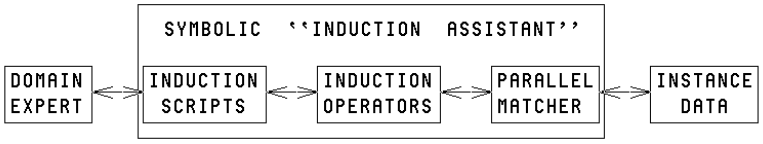


Figure 5. Logical system architecture. he domain expert invokes induction scripts on a ``parent'' pattern (terms are explained in the text). Induction scripts consist of a sequence of induction operators, separated by filters. Induction operators construct a set of syntactically variant patterns (``children'') from the parent. A subset of the children (the ``induction basis set'') is matched against the instance data using the parallel matcher. The results are returned to the induction operator, which composes them to compute the performance of the remaining children. All the evaluated children are returned to the script, which uses filters to prune unpromising possibilities. The surviving results of the script are returned to the domain expert. Scripts and filters execute in the serial front-end host, while induction operators and matching execute mostly in the parallel hardware. This figure appeared as figure 2 of Lathrop et al. (1990).

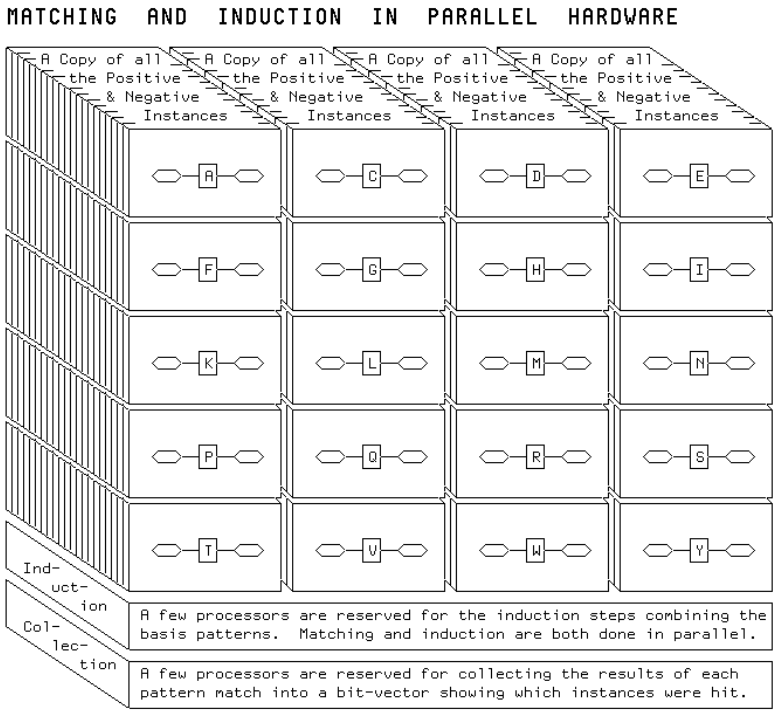
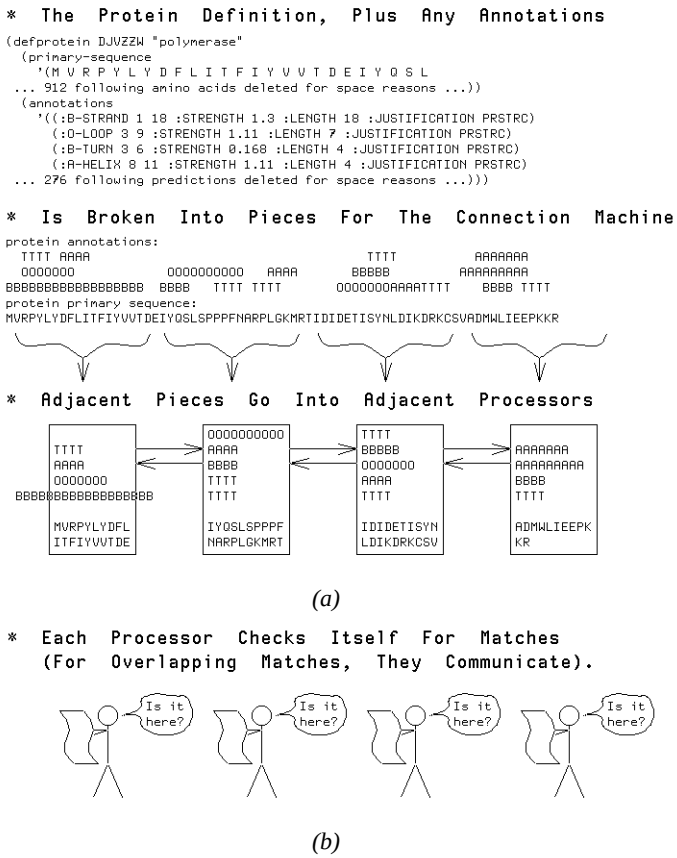


Figure 6. Schematic organization of parallel hardware usage. An induction operator can be decomposed into a ``basis set'' of multiple patterns (shown as boxed letters representing different pattern terms, flanked by lozenges representing preceding and following pattern components). These may be applied to multiple copies of the full positive and negative instance sets in parallel (each copy of the sets is shown fronted by the pattern matched against it, with vertical ``slices'' representing the individual instances). All patterns may be matched against all instances in parallel. The results may be recombined, also in parallel, within the few induction processors. This figure appeared as figure 3 of Lathrop et al. (1990).

nity as Climb-Tree, Drop-Link, Expand-Range, and so forth; for example, see Winston [1984]). We have parallelized both the match and the induction steps (see Figure 6). First an efficient, noise-tolerant, similarity-based parallel matching algorithm was developed. (See Figure 7.) The parallel matcher was used as infrastructure to construct efficient parallel implementations of several symbolic machine learning induction operators. (See Figure 8 for an example of an induction operator.) Finally, the induction operators were sandwiched together, separated by filters (both syntactic and empirical), to compose a crude form of induction scripts. These are invoked by the domain expert. (See Figure 9 for an example of an induction script.) Scripts and



(a)

(b)

Figure 7. Embedding and matching instances in the machine. Instances occur in very different lengths, but a constant-sized segment is placed in each processor. During the match phase, each processor checks only the constant-sized segment it contains. A communication protocol handles matches which extend over several processors. This figure appeared as figure 1.6 of Lathrop (1990).

filters execute in the serial front-end host, while induction operators and matching execute mostly in the parallel hardware.

Initial input to ARIEL consists of one or more seed patterns, and also two sets of protein primary sequences (one positive, one negative) annotated as described above. The proteins are loaded into static data structures in the parallel hardware at start-up. Thereafter the domain expert and ARIEL iterate through the pattern construction loop (a, b, c) described above. At each iteration, the domain expert invokes an induction script on a “parent” pattern. A single script may chain together several induction operators and filters. In-

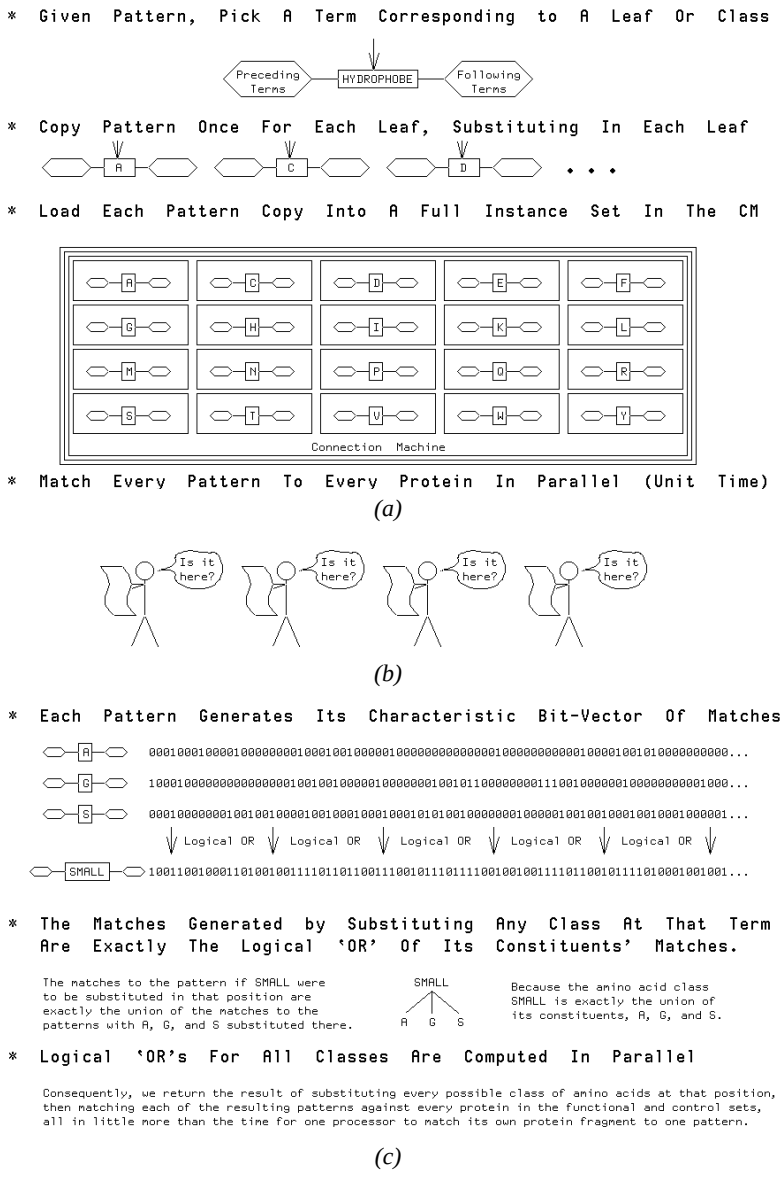
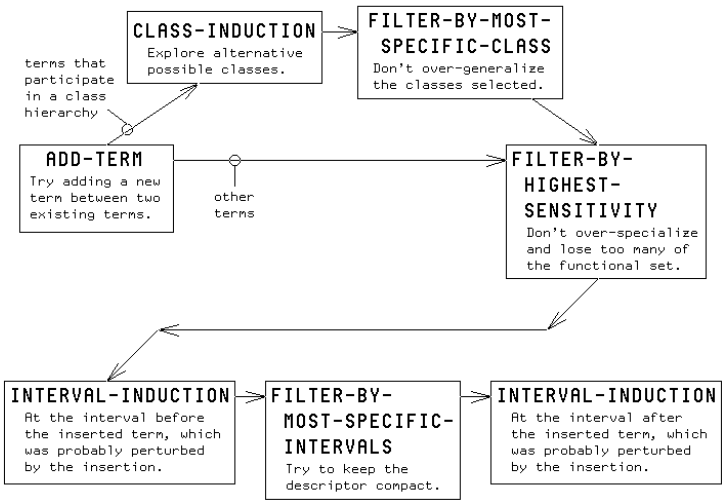
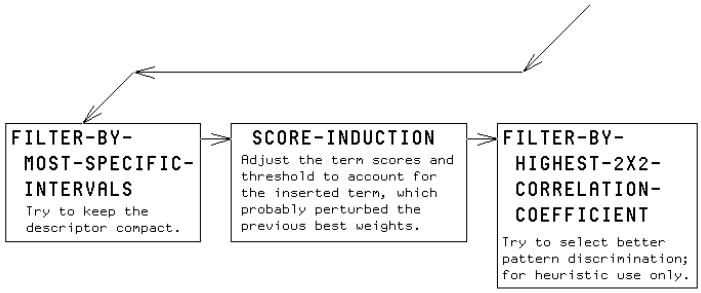


Figure 8. Induction operator example of CLASS-INDUCTION. This explores optimizations attainable by replacing a specified term in the parent pattern (the term HYDROPHOBE as illustrated here) by every other term from the generalization hierarchy (its replacement by SMALL is shown here for concreteness, but its replacement by every other term in the generalization hierarchy is also computed in parallel). This figure appeared as figure 4 of Lathrop et al. (1990).

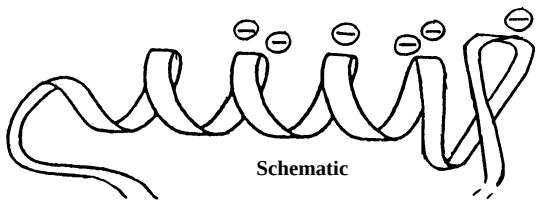


(a)



(b)

Figure 9. Induction script example of ADD-TERM-AND-REFINE. A script consists of a series of induction operators, separated by filters to prune unpromising candidates. These chain together common sequences of induction operators. Each induction operator generates and evaluates a large number of candidate patterns. Each filter step removes patterns based on some criteria. In the script shown here, the operator ADD-TERM is first applied to the parent pattern. If the term participates in a class generalization hierarchy, CLASS-INDUCTION is then applied to explore appropriate class terms. INTERVAL-INDUCTION is next applied, to adjust the intervals before and after the added term. Finally, SCORE-INDUCTION adjusts the term scores and threshold. This figure appeared as figure 5 of Lathrop et al. (1990).



Elements						
Local Hydrophobic profile minimum	{-1, 7}	alpha-helix with acid-faced segment	{-1, 8}	Local Hydrophobic profile maximum	[**V L I**] {4, 8}	ß-turn {-5,0} **D**

	Weights					
If absent	-0.5	-	-	-0.5	-	-1.5
If present	+0.5	function of net template charge	0.0	+0.5	0.0	+1.5
Space skipped if absent	+1	+16	+1	+1	+4	+1

Figure 10. The pattern found to be most diagnostic of a class of transcriptional activators containing the steroid hormone receptors. The schematic diagram was drawn by a domain expert. The remainder of the figure describes the pattern found by ARIEL. The pattern elements must be found in sequence. Elements in curly braces {n, m} *are allowed spaces between elements, where* n *is the minimum distance and* m *is the maximum. Note that a negative value for* n *allows the beginning of the following element to overlap the end of the preceding element. The weights given correspond to the pattern elements that they are below. All weights are relative to a pattern match threshold of +3.5 (see text). The spacer elements do not have weights; if the match is within the allowed space limits, the weight is zero, and if not, the weight is negative infinity.*

duction operators construct a set of syntactically variant patterns ("children") from the parent pattern. Only a subset of the children (the "induction basis set") is actually matched against the instance data, and from these the performance of the remaining children is computed. Filters prune unpromising possibilities. Finally, surviving children are returned to the domain expert, and the loop iterates. At the end of the session, output consists of a number of patterns evaluated for their sensitivity and specificity against the data set (see the section on Significance, Validity, and Pattern Quality, below), as well as the spectrum of sensitivity and specificity explored across pattern space.

Figure 10 shows a schematic representation (drawn by the domain expert) of the pattern found to be most diagnostic of a class of transcriptional activa-

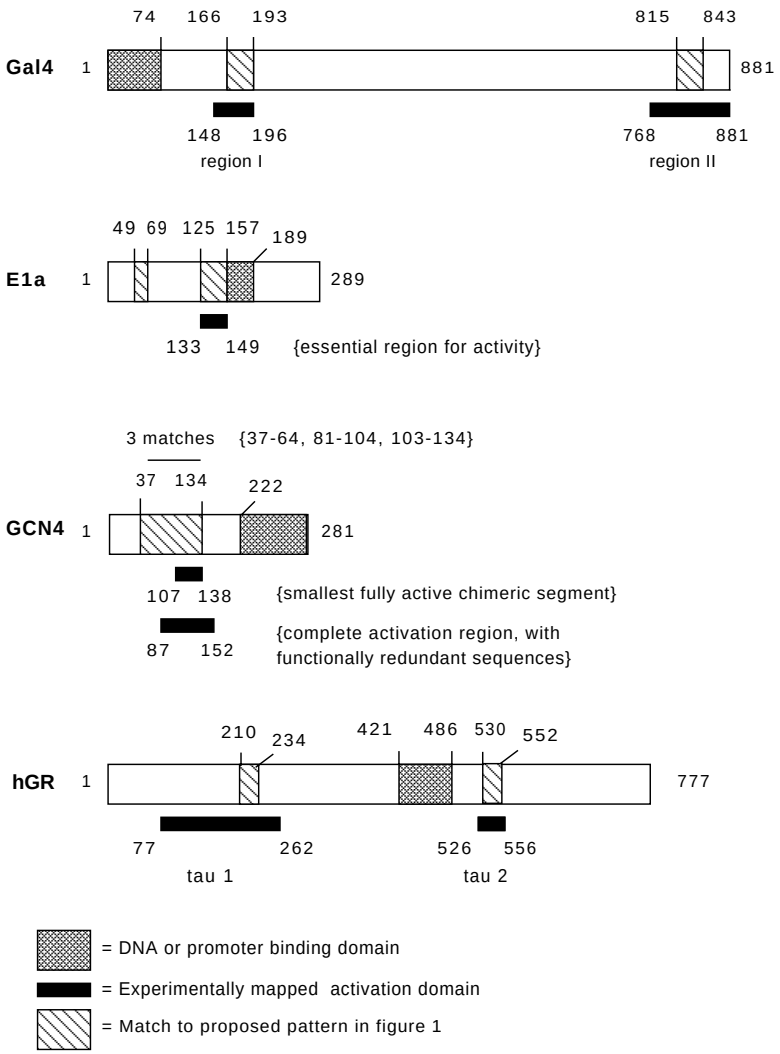


Figure 11 . Schematic of Four Transactivating Proteins. A schematic of four transactivating proteins showing the relative positions of the pattern matches, DNA binding domains, and experimentally mapped regions essential to that activity (hGR (Hollenberg and Evans 1988); E1a (Lillie and Green 1989); Gal4 (Ma and Ptashne 1987); and GCN4, (Hope et al. 1988)) These regions were used in the pattern construction, and so cannot be considered experimental verification of the pattern. Nonetheless, the close correspondence between pattern matches and functionally essential regions illustrates the potential utility of the pattern match for guiding future experimental work in unknown transactivating proteins. This figure appeared as figure 2 of Zhu et al. (1990).

tors containing the steroid hormone receptors, as well as the corresponding pattern as refined by ARIEL. Figure 11 shows in schematic four transcriptional activating proteins with the relative positions of the pattern matches and the experimentally mapped activation domains known to be essential to the function. These regions were used in the pattern construction, and so cannot be considered experimental verification of the pattern. Nonetheless, the close correspondence between pattern matches and functionally essential regions illustrates the potential utility of the pattern match for guiding future experimental work in unknown transactivating proteins.

4.1 ARIEL Protein and Pattern Representations.

ARIEL accepts as input the same annotated protein sequences as ARIADNE. Internally, ARIEL employs a graph-based pattern representation similar to ARIADNE, but it is not fully object-oriented due to data storage considerations arising from the massively parallel Connection Machine implementation. The major differences are that (1) the protein data structure is static, and so neither patterns nor target objects can be hierarchically constructed because annotations cannot be dynamically added; and (2) the annotations are implemented as typed weighted pointers, rather than as full-fledged objects in an object-oriented style, and so lack the notion of "containing" the amino acids which compose them.

ARIEL currently implements a restricted subset of the pattern language generality provided by ARIADNE. Only the most commonly used language terms (pattern elements) from ARIADNE are currently included. The set of ARIEL language primitives was fixed by choosing the ARIADNE language constructs that had been found to be most useful in the domain task. ARIADNE was designed to be a research vehicle into useful language constructs for the domain; consequently its pattern primitives are expressed in arbitrary LISP code making the language nearly arbitrarily user-extensible. ARIEL was designed to be a research vehicle into parallel symbolic induction; consequently it has a fixed set of pattern primitives whose semantics are directly encoded in the primitive hardware. ARIEL's pattern language permits description of nodes and their generalization classes, arcs, weights, threshold, and interval relationships. A pattern consists of a series of TERMS separated by INTERVALS. A THRESHOLD governs the overall match quality required to qualify as a successful match.

Each term specifies either a node class or an arc type. The semantics of a term specifying a node class are to match to a node belonging to that class. The semantics of a term specifying an arc type are to match to and traverse an arc of that type. Terms carry an associated SCORE-IF-ABSENT and SCORE-IF-PRESENT that govern how the pattern term matched against instance features contributes to the overall match score. These may indicate either a numeric score, minus infinity, or the weight attached to the instance

feature. Arc type terms indicate a default length, usually set to the average length of arcs of that type. Intervals separate adjacent pattern terms. The interval between two terms governs where to look for the next term after having found the previous one. An interval consists of a fixed part (representing a mandatory offset from the preceding term) and a variable part (indicating a window length within which to look for the next term), and thus is equivalent to a variable-length gap in specifying a minimum and maximum number of amino acids to skip.

4.2 Pattern Matching in Parallel Hardware.

The induction mechanisms are built upon, and call as a subroutine, a parallel matching algorithm. Virtually all processors in the parallel hardware (over 98%) are of type MATCH, and receive a pattern to test and an instance segment to match against it. A few processors are of type COLLECTION, and collect the global results of the match. A few processors are of type INDUCTION, and are used to combine the results of the induction operators (see Figure 6).

Both the pattern, and the instance data against which it is matched, are stored in the private data of a processor (see Figures 6 and 7). Once loaded, the instance data is permanently resident in the processor. New patterns are broadcast to the processors at the beginning of each match cycle. Bit-vectors are transmitted back to the host when the cycle ends. Thus the communication channel between front-end host and parallel hardware is of low bandwidth.

Large proteins are broken into several segments and stored in several adjacent processors. A communication protocol handles cases in which a successful match spans more than one processor. Each instance is assigned a unique bit position in a bit vector, which is stored with the instance in the processor. When each match cycle is complete, each processor compares the highest score achieved by any match terminating in its segment to the threshold from the pattern. If the threshold is met or exceeded, that processor sends (with logical inclusive OR) the instance bit representing its protein to the collection processor for that pattern. There is one collection processor dedicated to each pattern concurrently matched. Bits sent by processors meeting or exceeding the threshold set the corresponding bit in the collection processor, and thereby specify that the corresponding instance matched that pattern. Each collection processor winds up with a global bit-vector corresponding exactly to the successfully matched instances of its associated pattern.

Because we usually use no more than one or two hundred instances in the positive and negative sets, but have thousands of processors, it is possible to hold many duplicate copies of the full sets of positive and negative instances in the parallel hardware. Each full copy of the positive and negative instances is called a full instance set, and the processors holding them a full

processor set. Matching one pattern against one instance is independent of matching another pattern against a different instance, and so can be done in parallel (see Figures 6, 7 and 8). We put different patterns in each full processor set, but each processor in the same full set receives the same pattern. By matching against all of these concurrently, we obtain the result of matching many different patterns exhaustively against all the positive and negative instances, in essentially the real time taken to match one pattern against one segment of one instance.

For the domain studied here, this yields the following attractive match properties:

- Nearly Constant Time in Number of Instances; because the match against each instance is independent and concurrent. The only time dependency on number of instances occurs in transmitting the bit-vectors, an operation that consumes a tiny fraction of the time of the match.
- Nearly Linear Space in Total Instance Pool Size; because we need but one processor per segment. The space dependency on number of instances involves storing the instance data together with the instance-ID.
- Nearly Constant Time in Instance Size; because an instance is segmented and split over several processors. Any one processor looks only at a constant-length segment regardless of the total length of the instance. Communication costs for following arcs from one processor to another are nearly constant in instance size because the domain has high locality (i.e., arcs are mostly short).
- Most Communication Local for Domains With High Locality; because if the domain is sufficiently local, most communication except the global bit-vector OR used to compute the characteristic set can use local communication with adjacent processors.
- Nearly All Non-Local Communication Evenly Distributed in Both Source and Destination (No Bottlenecks or Collisions); because except for the global bit-vector OR, all non-local communication has unique source and unique destination.
- Space Constant Adjustable to Available Hardware, for Nearly 100% Processor Utilization; because we can vary the length of the segment we put in each processor. Decreasing the length uses more processors, increasing it uses fewer. We can vary it to fill the machine.
- Works on real-world problems using realistic hardware; for example the transcriptional activator pattern from Zhu *et al*. [1990] was run on an 8K CM-2 Connection Machine.

These bounds fail to be exactly constant or linear because each match pro-

cessor must also store and transmit its own instance-ID (specifying the bit position corresponding to its instance). A binary encoding of the instance-ID must consume at least log N bits, forcing the formal time complexity to O(log N) and the formal space complexity to O(N log N). However, bounding the size of the binary instance-ID by as few as 64 bits (tiny compared to the many tens of thousands of bits consumed by the typical instance itself, and less than 1/10 of 1 percent of each CM-2 processor's 64K bits of private memory) would suffice for well over ten sextillion instances, certainly far beyond the foreseeable future of parallel hardware. The time consumed in processing the instance-ID is currently trivial compared to seconds or even minutes for the match process itself, and the space consumed so small relative to other uses that the implementation actually encodes the instance-ID as a single bit set in a large bit-field for simplicity. Once the match and induction steps are complete, transmitting the bit-vector back to the front-end host incurs a small communication cost of one bit per instance per pattern.

4.3 ARIEL Pattern Induction Mechanisms.

Induction operators have been provided for the main pattern language components: class membership terms, interval relationships, weights and a threshold, and pattern elements. Each operator seeks to explore a single characteristic class of perturbations to the original pattern, related to a specific pattern language component. An induction operator transforms one initial pattern (parent) into a set of related patterns (children), and evaluates their performance on the positive and negative instances.

One of the main points of this research is to demonstrate parallel induction algorithms rendering it unnecessary to actually match each and every child pattern against the instances in order to evaluate its match performance. Matching each child pattern would be an undesirable waste of our computing resources, because the match step consumes most of the time and space of the system. Instead, we find a subset of pattern space (an "induction basis set") from which, once the performance of that subset is known, we can readily infer the exact performance of the other child patterns implied by the induction operator. Thus we match only a few child patterns against the instance data, and use those results to compute what the match performance of other children would be.

For example, CLASS-INDUCTION operates on a language term that participates in a class generalization network (often termed an A-Kind-Of, or AKO, hierarchy). In the domain, this corresponds to the physico-chemical classes of amino acids. The effect of applying CLASS-INDUCTION to a term in a pattern is to explore all new pattern variants which may be constructed by changing its class to any other class in the generalization hierarchy. As shown in Figure 8, for this operator the "induction basis set" can be just the leaves (or some suitable subset of the class hierarchy). In CLASS-

INDUCTION these are the children which substitute individual amino acids (leaves of the generalization hierarchy) at the position of interest. In the example of Figure 8, the class SMALL is composed of the leaves A, G, and S. The instances matched by a child substituting SMALL at a given term are exactly the union of the instances matched by the three children substituting A, G, or S. In general, each class is just the union of its leaves, so the instances matched by a child substituting any given class at the term are exactly the union of the instances matched bychildren substituting one of the leaves making up the class. Match performance on instance data is encoded in a bit-vector indicating which instances matched a pattern (by whether the bit corresponding to a given instance is on or off). Pattern results for the leaves can be merged by ORing together their bit-vectors of hit instances according to the union comprising the class. Figure 8 shows how the match bit-vector of the child substituting SMALL is computed by ORing the bit-vectors obtained from A, G, and S. The SMALL child is never actually matched to the instances.

The time required to explore all the children is nearly independent of either the number of instances or the number of class terms in the generalization hierarchy because both the match and the induction steps occur in parallel hardware. Provided sufficient parallel hardware is available, in little more than the time taken to match one pattern within one processor, we have effectively evaluated every pattern which can be formed from the parent by substituting any generalization class for the specified term, against every instance in the positive and negative sets. (Throughout this section, the discussion is phrased as if sufficient hardware were available. The hardware requirements are linear in the instance data size, and we have been able to solve real problems using an 8K Connection Machine. Thus, this is not an unreasonable simplification. The actual implementation adjusts to accommodate the hardware available, automatically iterating when an instance basis set will not entirely fit in the available processors.)

The other induction operators function similarly. The key is the decomposition of the syntactic operation on a pattern in such a way that we actually match against the instance data only a subset of the children (the “induction basis set”), but can then compute the results for all the other children without actually matching against them.

One important induction operator explores variations in a range of values, such as the interval separating two adjacent terms. The children are patterns that vary from the parent by having a different interval (a mandatory offset plus an active window within which the following term may appear, specifying a variable-length gap) separating the two terms. The corresponding induction basis set would be the intervals having offsets of different lengths and an active window of width one. From the appropriate union of these primitive (“basis”) intervals we can construct every other interval in the set

of child patterns, i.e., intervals with an arbitrary offset and window. Consequently, from the OR of the bit-vectors associated with these "basis" patterns we can compute the bit-vectors to asssociate with every other child pattern. Thus we evaluate every pattern which can be formed by substituting a different interval between the two adjacent terms, again in little more than the time taken to match one pattern within one processor.

Another induction operator explores variations in the overall match threshold. Although there are potentially an infinite number of children, we are really interested only in the critical set of threshold values at which some instance switches between being matched and not matched. Rather than explicitly constructing each possible child, this induction operator finds these critical values using the global maximum instruction implemented by the parallel hardware. First the parent pattern is matched as described in the previous section. The maximum score achieved in any processor is retrieved and treated as the threshold. As above, the bit-vector OR of instances matching at this threshold is retrieved, and the corresponding child is constructed. Successively each distinct next lower score is retrieved and treated as the threshold, and the corresponding bit-vector and child constructed. In this way we rapidly evaluate the match performance of all children, while performing the match step only once.

Computing the induction operator for term weights is more subtle. It explores simultaneous changes to the threshold and the SCORE-IF-ABSENT and SCORE-IF-PRESENT parameters of any single term. (In practice we compute this operator for all terms in parallel, but the exposition is simpler when only a single term is considered.) For a given term, the basis set consists of two child patterns: one with SCORE-IF-ABSENT zero and SCORE-IF-PRESENT minus-infinity, the other with the values reversed. These two patterns are matched against the instance data, and their match performances for each different setting of the match threshold evaluated as just described. The instances matched by either one of these patterns at any setting of the threshold is identical to those that same pattern would match if its zeroed score were set to the negative of the match threshold and the threshold were set to zero. Also, the match support for this term in the two patterns is necessarily disjoint. Consequently, we can compute a child term that matches the union of the instance matches of both patterns at any of the match thresholds individually, by setting the child's SCORE-IF-ABSENT to the negative of the threshold chosen for the first pattern, its SCORE-IF-PRESENT to the negative of the threshold chosen for the second pattern, and its threshold to zero. Thus we evaluate every pattern which can be formed by substituting a different SCORE-IF-ABSENT and SCORE-IF-PRESENT at a single given term and simultaneously changing the match threshold to any value, again in little more than the time taken to match one pattern within one processor.

The set of induction operators as a whole is suitable for use in a heuristic

search through pattern space. Their major strength is that a whole class of characteristic perturbations may be explored rapidly. This greatly speeds the search for a pattern which satisfactorily discriminates positive from negative instances; eliminates the whole lowest level of search planning complexity that previously attended to exploring each class of perturbations efficiently; and allows a more systematic search through pattern space. Their major weakness is that, because they explore only one major characteristic at a time, they are poor at detecting interaction effects. This is only partially compensated for by the use of induction scripts, which automatically chain together some of the more common interactions.

Induction scripts are constructed by sequencing induction operators together in a pipeline, separated by filters to prune unpromising candidates. For example, the script for adding a new term (see Figure 9) involves first exploring the class generalization hierarchy (to find plausible terms to add), then exploring different intervals before and after the new term (because the insertion will disturb the previous spacing), and finally exploring different weights for the new term, and a new match threshold. Overall, the induction scripts implement a variant of symbolic hill-climbing beam search through symbolic pattern space. The search is analogous to hill-climbing because each induction operator modifies only one aspect of a pattern at a time, just as hill-climbing search steps in one direction at a time, and because we seek to go "uphill" as determined by increasing sensitivity and specificity (this is not a single "direction"). It is analogous to beam search because filters retain a number of promising candidates at each stage.

4.4 Discussion of ARIEL

The resulting induction scripts are a fairly crude search heuristic. The pruning of intermediate patterns may be too aggressive at some step and so discard a possible variant. The scripts are also limited in that they contain no conditionalization, no parameterization, and no variables. There is currently no automated effort to systematically explore all alternative avenues from step to step within an induction script, nor from script to script. The scripts and operators use a "generate and test" strategy for finding candidate patterns, and so are less direct than methods which construct candidate patterns by inspecting and manipulating the instances (as does PIMA, discussed in the next section).

In spite of their current limitations, in practice the scripts prove to be reasonably effective at exploring different possibilities and focusing computational effort into useful areas. The scripts raise the general level of abstraction at which the domain expert plans the search through pattern space. By removing the necessity to expend planning effort on low-level induction steps the domain expert is freed to formulate higher-level plans at a higher level of abstraction, covering more possibilities with each planning step. Fu-

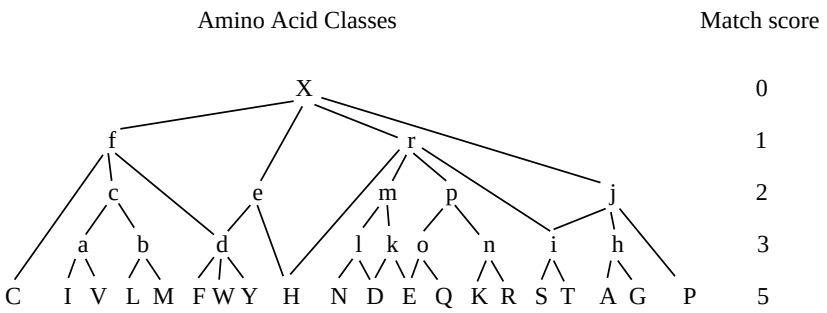


Figure 12. The amino acid class hierarchy used to construct AACC patterns during multi-alignment. Uppercase characters, one-letter amino acid codes; lowercase characters, designated amino acid classes; X, wild-card character representing one amino acid of any type. The match score between any two aligned elements equals the score assigned to the minimally inclusive class in the hierarchy that includes both elements. [From Smith and Smith, 1992].

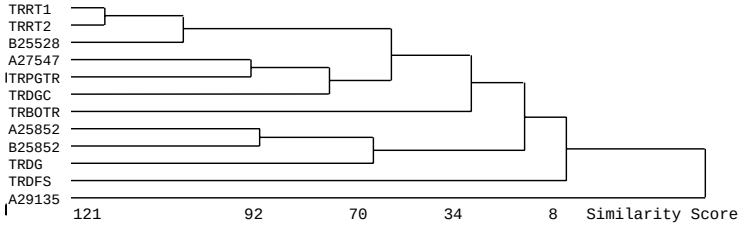
ture improvements to ARIEL would add a planning component on top of (and interacting with) the induction scripts, rather than attempting to expand the scope and coverage of the script-based search approach itself (see Hunter's chapter in this volume on the relationship of planning to learning).

5 Covering Pattern Construction (PIMA) and Search (PLSEARCH) Tools

As ARIEL provides an automated method to construct complex patterns from seed patterns, we now ask: "Where does the seed pattern come from in the first place?" A completely automated method for seed pattern construction is provided by PIMA (our Pattern-Induced Multiple Alignment program), which uses a modified dynamic programming algorithm to inductively construct primary sequence patterns common to a family of functionally-related proteins [Smith and Smith, 1990, 1992].

PIMA employs a pre-defined set of amino acid classes (based on a physico-chemical generalization hierarchy, Figure 12) to construct a primary sequence pattern from a dynamic programming generated alignment. Given an alignment between two sequences generated using the Smith and Waterman [1981b] local optimal alignment algorithm, an "Amino Acid Class Covering" (AACC) pattern can be constructed from the alignment by identifying the smallest amino acid class which includes (covers) each pair of aligned amino acids. If two identical amino acids are aligned at a position, then the symbol

A. Trypsinogen/Venom serine proteases




```
TRRT1                Trypsinogen  I precursor         - Rat    #EC-number 3.4.21.4
TRRT2                Trypsinogen  II precursor        - Rat    #EC-number 3.4.21.4
B25528               Trypsinogen  precursor           - Mouse  #EC-number 3.4.21.4
A27547               Trypsinogen, cationic, precursor - Rat    #EC-number 3.4.21.4
TRPGTR               Trypsinogen                     - Pig    #EC-number 3.4.21.4
TRDGC                Trypsinogen  cationic, precursor - Dog    #EC-number 3.4.21.4
TRBOTR               Trypsinogen                     - Bovine #EC-number 3.4.21.4
A25852               Trypsinogen  I precursor         - Human  #EC-number 3.4.21.4
B25852               Trypsinogen  II precursor        - Rat    #EC-number 3.4.21.4
TRDG                 Trypsinogen  anionic, precursor  - Dog    #EC-number 3.4.21.4
TRDFS                Trypsinogen                     - Spiny Dogfish  #EC-number 3.4.21.4
A29135               Venom serine protease           - Barba amarilla #EC-number 3.4.21.29
```

B. Covering pattern constructed from the above cluster.

```
1                20 21                40 41                60 61                80
lllhXaaGGXXCXXlXXPbX XXcXXXXigFCGXkLIXXXW VakApHCXlXXclaiLGlXX XXXXlXXEXcXXXXXXcXXP 80
lXlXXXcllgDIcLIiLXlX XXXlXlaXXaXLPlXXXXXG lXXXaXGWGXXXlgXXXXXl XXlCXlXXacXlXXClXXYX 16(
GgaXXlXcCXGcclGGXDkC XGDSGGPaaXlGXcQGaaSW  200 GXXgCAXXXXPpcXXiVclb aXWIllXaA      /22
```

C. Pattern constructed from 4 protein tyrosine kinase sequences

```
1              20 21              40 41              60 61              80
lXaXXaXXXaXXcXXXXlXX XXXXXlXgggXlXlXlllXl ghlAgXlcXXcXgggXXXXl laXXXXXXXXXXXXXXlXXX 80
XXXggXXXXXcXXXXNXXaX XgggggggXXXXXXlXXlXXX lXXXXXXXXXiXXaXXXGXX XXlXXiXlXXXXXXXggXlXl 16(
XXaXXcXXXXXXXXXXXXllc XXXXlXXXXXXXXclXgglX XlXXlXXgggggggglXXlgg lXXXXXXXTXXXXXXlXXXX 24(
XlXXXXXXXXXXXXXXXaXX XXXXXXXXXXcaXXXXXcXX XXXXXXXXXXXXXXXXgggg XXXXXcXbXlXXXXggXlXXX 32(
XXXXXSXXXXXaXXlXXXgg gggggggglcXRllaXLXXl LGlGXFGXVcXplXXXaXXX XXXXXVAVKXallgAlXlXX 40(
XXcXlExlaclXcXgXXlaa lccGXcklXXPXccaXEcXX XGlLXXbLilgggggggggg gggggggggggggggggggg 48(
gggXXXXXXXXXGXgggggg gggggggggggggggggggg gggggggggggggggggggg gggggggggXXXXcXlcaX 56(
cXXlaXXGMXbLXXXlcVHR DLAXRNccaXlXXXcKIXDF GcXRDaclllXYXXXXlXXa PaiWMXXESaXXXXbTTXSD 64(
cWSFGaaLWEIXkcpXlPbX XaXXllXcglXaXlGXXcXl PXXXXXlcXXaMXXCWlXlX lXggggXXXXlXcXXXXggi 72(
XXXXXcXXXXXlXXXXXXlX XXgggggXXXXlXXXcP                                          /75
```

Figure 13. (A) A binary dendrogram constructed by clustering the pair-wise scores for a set of serine protease-related sequences using a maximal linkage rule. (B) The AACC pattern generated from this cluster. Upper case characters: IUPAC one-letter amino acid codes; lower case characters: amino acid classes: a=[ILV], b=[FWY], c=[ILVFWYCM], h=[DE], i=[HKR], j=[NQ], k=[ST], l=[DEHKRNQSTBZ], p=[AG], X=wild-card (1 amino acid of any type), g= gap character (0 or 1 amino acid of any type). The 8 cysteine residues forming the 4 disulfide bonds that are common to this set of proteases are all present in the pattern and are shown highlighted in bold/underlined type. The highlighted regions at positions 178-188 and 199-200 are at the substrate binding site; the former region contains the "GDSG" motif conserved in known serine proteases. Highlighted positions 46 and 91, along with position 184 constitute the active site charge relay system. (C) AACC pattern constructed from 4 protein serine/threonine kinase sequences: human insulin receptor (NBRF/PIR locus A05274) and feline trk (locus A25184), mouse PDGF receptor (locus A25742) and human ret (locus TVHURE). Six conserved regions previously identified within the catalytic domains of protein serine/threonine kinases are shown highlighted. [Adapted from Fig. 4 of Smith and Smith, 1990].

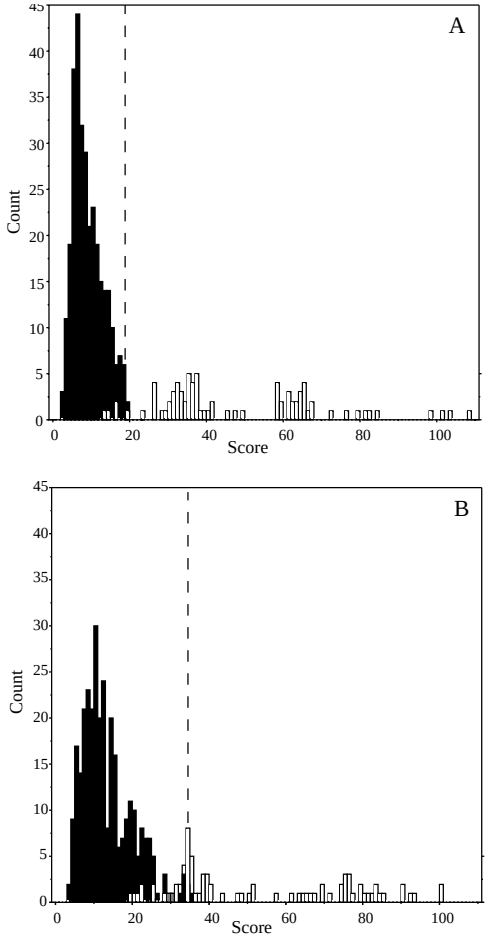


Figure 14. Match score distributions comparing AACC pattern vs. sequence matches (A) to sequence vs. sequence matches (B). (A) An AACC pattern was constructed from 4 protein tyrosine kinase sequences (Fig. 14C) and matched to 2 control sets: a positive control set composed of 81 protein kinase (PK) related sequences (white bars) and a negative control set 313 non-PK-related sequences (black bars; mean±s.d=10.88±4.60); where the two sets of matches overlap, the smaller bar at each position is shown in the foreground; at those positions where the bars are the same height, a single stippled bar is displayed. The dashed line indicates the 99th percentile of the negative control scores. (B) Score distributions of a single PK sequence (human ret, locus TVHURE) matched against the same positive (white bars) and negative (black bars; mean±s.d=13.59±6.59) control sets described above. As in (A), the dashed line indicates the 99th percentile of the negative control set. [From Smith and Smith, 1990]

for that amino acid is placed at the corresponding position in the pattern. In the other case, two non-identical residues are aligned, and the smallest covering class (represented by a lower-case characters in Figure 12) is placed at that position. Positions in an alignment where an amino acid was is paired to the null element are converted into gap characters (`g') in patterns. The dynamic programming algorithm has been extended such that each gap character can function as 0 or 1 amino acid of any type during subsequent pattern alignment (described below). This is analogous to the way variable spacing can be specified in a regular expression pattern.

Given an unaligned set of protein sequences from a homologous family (such as all alpha globin sequences), a single primary sequence pattern, representing the conserved sequence elements common to all members of the family, can be constructed. The method involves using patterns constructed from pairwise alignments as input for subsequent rounds of alignment and pattern construction. First, all pairwise alignments between sequences in a set are performed. The resulting pairwise scores are then clustered using a maximal linkage rule [Sneath and Sokal, 1973] to generate a binary dendrogram (i.e., tree; Figure 13). The two most similar sequences in the clustered set are then aligned and a covering pattern constructed as described above. Patterns are similarly constructed for each node in the tree by sequentially moving down the tree, aligning at each step the patterns or sequences connected by the next most similar node, until a single "root" pattern is constructed for the entire set. The root pattern represents those conserved primary sequence elements common to all members of the set.

In a manner analogous to a regular expression pattern, covering patterns so constructed will match with equal score all of the sequences from which the pattern was derived. Indeed, these covering patterns can be directly translated into standard regular expression patterns. This pattern construction algorithm can be thought of as a method to construct a single regular expression pattern for a set of homologous sequences, and our modified dynamic programming algorithm is, in essence, a method to perform "regular expression matching with mismatching" (the latter problem has been approached differently by [Myers and Miller, 1989]).

Covering patterns can be more diagnostic for family membership than any of the individual sequences used to construct the pattern. This can be shown by comparing the diagnostic capability of a pattern with that of the sequences used to construct the pattern. An example of this is shown in Figure 14, where a pattern constructed from four sequences taken from the protein kinase (PK) family is matched against (1) a positive control set of 80 PK sequences, and (2) a negative control set of 313 different sequences not related to the PK family. For the PK pattern, only four PK sequences in the positive set had match scores less than the 99th percentile of the nega-

tive control set (Figure 14a). Using the same 99th percentile criterion for determining false negative (FN) matches, the four individual PK sequences generated 46, 29, 23, and 20 FN matches (the example with 20 FN matches is shown in Figure 14b).

There are two apparent reasons why covering patterns can be more diagnostic than the sequences used to construct them. First, in sequence vs. sequence alignments, mismatch and gap penalties generated at non-conserved positions can easily outweigh the match scores contributed at the limited number of conserved sites. This is especially true in families such as the PKs which encompass a diverse set of sequences with low overall sequence similarity. In the patterns, non-conserved positions are converted into "wild-card" (X) and gapped (g) characters that do not contribute to mis-matching. Second, chance similarities between non-conserved regions of any single positive instance sequence and non-related sequences can broaden the score distribution of the negative control set. Non-conserved site/regions are not represented in the patterns and thus such chance similarities are eliminated.

5.1 An Application of PIMA.

Using the covering pattern construction methodology described above, we have constructed a database of covering patterns for all sequence families in the SWISS-PROT Protein Sequence Database. Families of related protein sequences were identified by performing all possible pairwise comparisons between all sequences in the SWISS-PROT database (i.e., "running the database against itself") using BLAST, an ultra-fast k-tuple search program [Altschul *et al*., 1990]. The resulting set of pairwise scores was then clustered into families using a maximal-linkage clustering algorithm. The covering pattern construction program was then used to generate a single primary sequence pattern for each family. The current pattern library (release 4, based on SWISS-PROT release 13) contains 2026 patterns derived from all families of 2 or more members (encompassing 10664 of the 13837 sequences in the database). In collaboration with the Human Retrovirus and AIDS Sequence Database, we have also constructed a pattern library for all gene families of HIV (AIDS) -related viruses. Biologists with new sequences of unknown function can search these pattern databases with PLSEARCH (Pattern Library SEARCH), a pattern search tool that utilizes PIMA's pattern alignment algorithm. As described above, pattern searches can be more sensitive than conventional sequence vs. sequence database search programs since covering patterns can be more diagnostic for family membership than any of the individual sequences used to construct a pattern. We are also using these primary sequence patterns as "seed" inputs to construct complex hierarchical patterns using ARIEL's pattern induction system.

6 Significance, Validity, and Pattern Quality

> "Validity, in a metric sense, is how well the test actually measures what its name indicates that it measures the degree to which the test reports what it purports to report."
>
> —R. G. Lathrop [1969]

Questions of significance and validity are familiar fare to the statistics and machine learning communities. Here we briefly touch on these issues with respect to protein sequence patterns (see also [Felsenstein 1988; Karlin and Macken 1991; Karlin *et al*. 1991]). The issue is of more than theoretical concern to the domain practitioners: on more than one occassion, a pattern for some defining class has been published which, while indeed matching the defining (positive) set, subsequently was found to match almost every other sequence in the sequence databases as well! Such a pattern is of little value.

At a minimum, patterns must be tested against both a positive and a negative set of sequences. The result yields a 2x2 contingency table:

		Actual class: Positive	Actual class: Negative
	Matched pattern	TP (number of true positives)	FP (number of false positives)
	Did not match pattern	FN (number of false negatives)	TN (number of true negatives)

While this provides a complete description of the pattern behavior on both positive and negative sequences, other derivative statistics are also useful:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Positive\ \text{Pr}edictive\ Value = \frac{TP}{TP + FP}$$

$$2x2\ Correlation\ Coefficient = \frac{(TP \cdot TN - FP \cdot FN)}{\sqrt{(TP + FP) \cdot (FP + TN) \cdot (TN + FN) \cdot (FN + TP)}}$$

In particular, sensitivity and specificity should always be reported for any published pattern. A pattern which achieves 1.0 for both sensitivity and specificity exhibits perfect discrimination and is sometimes refered to as "diagnostic". Patterns may be quite useful even if their sensitivity and specificity are not both 1.0, but these values must be known for its full utility

to be realized. For example, a pattern for alpha helices that achieved 1.0 sensitivity but only 0.75 specificity would be quite useful—- it would guarantee to find every alpha helix, although it would over-predict somewhat. Similarly, a pattern for alpha helices that achieved only 0.75 sensitivity but 1.0 specificity also would be quite useful—- it would not find every alpha helix, but it would guarantee complete confidence in the matches it did find. Nonetheless, matches to these two patterns would be interpreted very differently, and this is not possible unless their sensitivity and specificity are both known.

> "The word `significantly' has a somewhat different meaning in statistics than it has in common usage. In everyday usage, a significant difference is one which is of practical import. A significant difference in statistical terms implies a difference that is unlikely to have occurred by chance alone."
>
> —R. G. Lathrop [1969]

The statistical significance of induced patterns is always of concern. This is especially so for those interested in applying AI methods. In our case, we have greatly enriched the expressive representational power of both the instance and the pattern languages relative to existing better-understood sequence-based languages current in the domain. Is it possible that their expressive power is now so rich that we could detect a discriminating "regularity" in any randomly drawn set, even when no such regularity existed, or existed solely due to chance? The nature of the domain hampers the usual methods for establishing significance, especially due to sequence similarity, sample bias, and the small size of typical defining sets. Additionally, any method in which the domain expert intervenes in any way cannot use the cross-validation (leave-one-out) verification methodology, because the domain expert cannot be "reset" between trials. Any method which refines patterns in an iterative loop cannot use methods such as a means test, analysis of variance, or chi square, all of which assume a single independent trial.

Nonetheless, the languages we use appear to more closely reflect the underlying domain ontology as understood by domain experts; the patterns inferred reflect plausible generalizations that are directly subject to experimental tests by domain experts; tests of the pattern against a control (negative) set not used in pattern construction can provide an unbiased estimate of the false positive rate; and pattern-based discrimination should always be related to the domain's existing scientific literature. In the end, any computational technique can do no more than supply a hypothesis. Verification occurs only in an experimental setting, where a prediction is compared to Nature. Clearly, more rigor is desirable, and statistical significance and validity is an area of inquiry where formal consideration by theoreticians and their methods would be welcome. The remainder of this section will briefly survey some of

the domain-based issues that arise, making an attempt to discuss issues that are intrinsic to the general problem of inferring functional patterns in the domain rather than artifacts of our "Induction Assistant" approach.

Instances often share evolutionary history (known as "homology" in the domain), and hence fail to be independent of each other. These can be thought of as near-identical twins in an evolutionary sense. This is especially true of functionally related proteins, because related functions often spring from a common ancestral gene. Typically, the more recent the evolutionary divergence, the greater the degree of similarity (for example, functionally equivalent proteins from man and chimpanzee may differ by a few percent or less, while corresponding proteins from man and yeast may differ by more). This complicates the analysis of our positive (defining) set. The best current approach is to run a primary sequence similarity analysis beforehand. One may then count homologous families (clusters) as a single sequence for reporting purposes, or discard sequences until each family has but one remaining representative. Unfortunately, information is lost by these approaches, while dependencies undetectable by primary sequence similarity may still remain. Clustering methods proposed by Felsenstein [1985], Altschul *et al*. [1989], and Sibbald and Argos [1990b] promise to eventually provide a better approach. Here one can differentially weight each sequence relative to its similarity to other sequences, thereby attempting to retain the extra information present in their diversity while compensating for the partial lack of independence. However, the effect of cluster-based differential weights on any of the standard statistical tests of significance is unclear.

The available databases do not represent anything close to a uniform sampling of the space of all proteins. Rather, the sampling represents a highly skewed bias reflecting both the research interests of individual domain researchers (for example, hemoglobins and myoglobins are vastly over-represented in the protein sequence databases), and current technological limitations on what we can study (for example, membrane-bound proteins are vastly under-represented in the three-dimensional (folded shape) database because they are virtually impossible to crystallize). Thus it is currently impossible to construct a control set of negative instances whose distribution accurately reflects the space of all negative instances. One current approach is to construct a stratified negative set that attempts to sample the major protein groups systematically. Another common approach is to randomly sample the current databases, controlling for homology as discussed above.

Another problem, also familiar to the statistics and machine learning communities, arises from the small size of typical defining sets (in many cases, only a few dozen positive instances, or less, are known to science, especially if highly similar sequences are removed). This raises the possibility of overfitting the data, also known as memorizing the training set. While cross-validation or jack-knife techniques may offer assistance, holding out a test set

may leave an impoverished training set. Additionally, the domain issue of similar sequences is especially bothersome for these methods, and domain opinion is divided. The problem arises when a sequence in the test set is highly similar to another sequence in the training set. One school of thought asserts that, because the test sequence is essentially duplicated in the training set, the situation is as if the test sequence had not been held out at all and one is effectively testing on the training data, invalidating the trial. The contrary school of thought holds that, because similar sequences occur so frequently in Nature, the next unknown sequence that arises may be highly similar to a known sequence and one is effectively modeling an intrinsic domain fact, yielding a better estimate of expected performance in practice. The former position is probably preferable because more conservative, but opinions differ.

We close by pointing out several other minor protein-based violations of common underlying statistical assumptions. First, the positive and negative sets often differ in measurable ways from each other in ways unrelated to the phenomenon under test (for example, the average positive instance length is often significantly longer or shorter than the average negative instance). Second, the underlying distributions are not normal. However, most statistical tests are relatively insensitive to minor deviations from normality. Third, while the negative set is usually stratified or otherwise corrected for the sample independence problem, the positive set typically employs all the positive instances known to science (while compensating for homology as discussed above). Thus, the selection criteria for inclusion in the sets differs. Fourth, if homology is controlled for in the control set differently than in the defining set, then the degree of internal independence varies between the two sets.

Notes

1. Ariadne was the Cretan princess who gave Theseus a ball of thread, by which he found his way out of the Labyrinth after slaying the Minotaur.
2. ARIEL is a combination of ARI-adne and parall-EL, and also a pleasing character from Shakespeare's *The Tempest*

References

Abarbanel, R. M. (1985) *Protein Structural* Proteins *Knowledge Engineering*. Ph.D. diss., Univ. of California, San Francisco.

Abarbanel, R. M., Wieneke, P. R., Mansfield, E., Jaffe, D. A., Brutlag, D. L. (1984) "Rapid Searches for Complex Patterns in Biological Molecules." *Nucleic Acids Res.* 12:263-280.

Altschul S. F., Carroll R. J., Lipman D. J. (1989) "Weights for Data Related by a Tree." *Journal of Molecular Biology* 207:647-653.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J. (1990) "Basic Local Alignment Search Tool." *Journal of Molecular Biology* 215(3):403-410.

Bacon, D. J., Anderson, W. F. (1986) "Multiple Sequence Alignment." *Journal of Molecular Biology* 191:153-161.

Barton, G. J. (1990) "Protein Multiple Sequence Alignment and Flexible Pattern Matching." in *Methods in Enzymology*, ed. R. F. Doolittle, 183:403-429. Academic Press.

Barton, G. J., Sternberg, M. J. E. (1987a) "Evaluation and Improvements in the Automatic Alignment of Protein Sequences." *Protein Engineering* 1:89-94.

Barton, G. J., Sternberg, M. J. E. (1987b) "A Strategy for the Rapid Multiple Alignment of Protein Sequences." *Journal of Molecular Biology* 198:327-337.

Barton, G. J., Sternberg, M. J. E. (1990) "Flexible Protein Sequence Patterns: A Sensitive Method to Detect Weak Structural Similarities." *Journal of Molecular Biology* 212:389-402.

Bashford, D., Chothia, C., Lesk, A. M. (1987) "Determinants of a Protein Field." *Journal of Molecular Biology* 196:199-216.

Birktoft, J.J., Banaszak, L.J. (1984) "Structure-function Relationships Among Nicotinamide-adenine Dinucleotide Dependent Oxidoreductases." In *Peptide and Protein Reviews* ed. Hearn, M. T. W., 4:1-47, Marcel Dekker, New York.

Blundell, T. L., Sibanda, B. L., Sternberg, M. J. E., Thornton, J. M. (1987) "Knowledge-based Prediction of Protein Structures and the Design of Novel Molecules." *Nature* 326:347-352.

Bode, W., Schwager, P. (1975) "The Refined Crystal Structure of Bovine Beta-trypsin at 1.8 A Resolution." *Journal of Molecular Biology* 98(4):693-717.

Bohr, H., Brunak, S., Cotterill, R., Fredholm, H., Lautrup, B., Petersen, S. (1990) "A Novel Approach to Prediction of the 3-dimensional Structures of Protein Backbones by Neural Networks." *FEBS Letters* 261:43.

Bork, P., Grunwald, C. (1990) "Recognition of Different Nucleotide-binding Sites in Primary Structures Using a Property-pattern Approach." *Eur. J. Biochem.* 191:347-358.

Boswell, D.R. (1988) "A Program for Template Matching of Protein Sequences." *CABIOS* 4(3):345-350.

Bowie, J.U., Luthy, R., Eisenberg, D. (1991) "A Method to Identify Protein Sequences that Fold into a Known Three-dimensional Structure." *Science* 253:164-170.

Bradley, M., Smith, T.F., Lathrop, R.H., Livingston, D., Webster., T.A. (1987) "Consensus Topography in the ATP Binding Site of the Simian Virus 40 and Polyomavirus Large Tumor Antigens." *Proc. of the Natl. Acad. of Sciences USA*, 84:4026-4030.

Breese, K., Friedrich, T., Andersen, T.T., Smith, T.F., Figge, J. (1991) "Structural Characterization of a 14-residue Peptide Ligand of the Retinoblastoma Protein: Comparison with a Non-Binding Analog." *Peptide Res.* 4(4):220-226.

Brutlag, D.L., Dautricourt, J.-P., Maulik, S., Relph, J. (1990) "Improved Sensitivity of Biological Sequence Databases." *CABIOS* 6(3):237-245.

Chou, P.Y., Fasman, G.D. (1978) "Empirical Predictions of Protein Conformation." *Ann. Rev. Biochem* 47:251-276.

Clark D.A., Barton G.J., Rawlings C.J. (1990) "A Knowledge-based Architecture for Protein Sequence Analysis and Structure Prediction." *J. Mol. Graph.* 8:94-107.

Cockwell, K.Y., Giles, I.G. (1989) "Software Tools for Motif and Pattern Scanning." *CABIOS* 5(3):227-232.

Cohen, B. I., Presnell, S.R., Cohen, F.E. (1991a) "Pattern-based Approaches to Protein Structure Prediction." *Methods of Enzymology* 202:252-268.

Cohen, B. I., Presnell, S. R., Morris, M., Langridge, R., Cohen, F.E. (1991b) "Pattern Recognition and Pprotein Structure Prediction." *Proc. 24th Hawaii Intl. Conf. on System Sciences*, pp. 574-584, IEEE Computer Soc. Press, Los Alamitos, CA, USA.

Cohen, F. E., Abarbanel, R. M., Kuntz, I. D., Fletterick, R. J. (1983) "Secondary Structure Assignment for Alpha/beta Proteins by a Combinatorial Approach." *Biochemistry* 22:4894-4904.

Cohen, F.E., Abarbanel, R.M., Kuntz, I.D., Fletterick, R.J. (1986) "Turn Prediction in Proteins Using a Complex Pattern Matching Approach." *Biochemistry* 25:266-275.

Cohen, F.E., Gregoret, L., Presnell, S.R., Kuntz, I.D. (1989) "Protein Structure Predictions: New Theoretical Approaches." *Prog. Clin. Biol. Res.* 289:75-85.

Collins, J.F., Coulson, A.F.W., Lyall, A. (1988) The Significance of Protein Sequence Similarities. *CABIOS* 4(1):67-71.

Corpet, F. (1988) "Multiple Sequence Alignment with Hierarchical Clustering." *Nucleic Acids Res.* 16(22):10881-10890.

Creighton, T.E. (1983) *Proteins: Structure and Molecular Properties*. W.H. Freeman and Company, New York.

DeCaprio, J.A., Ludlow, J.W., Figge, J., Shew, J.-Y., Huang, C.-M., Lee, W.-H., Marsilio, E., Paucha, E., Livingston, D.M. (1988) "SV40 Large Tumor Antigen Forms a Specific Complex with the Product of the Retinoblastoma Susceptibility Gene." *Cell* 54:275-283.

DeCaprio, J.A., Ludlow, J.W., Lynch, D., Furukawa, Y., Griffin, J., Piwnica-Worms, H., Huang, C.-M., Livingston, D.M. (1989) "The Product of the Retinoblastoma Susceptibility GEne has Properties of a Cell Cycle Regulatory Component." *Cell* 58:1085-1095.

Doolittle, R.F., Hunkapillar, M.W., Hood, L.E., Davare, S.G., Robbins, K.C., Aaronson, S.A., Antoniades, H.N. (1983) "Simian Sarcoma Virus Onc Gene, v-sis, Is Derived from the Gene (or Genes) Encoding a Platelet-derived Growth Factor." *Science* 221:275-277.

Dyson, N., Howley, P.M., Munger, K., Harlow, E. (1989) "The Human Papilloma Virus-16 E7 Oncoprotein Is Able to Bind to the Retinoblastoma Gene Product." *Science* 243:934-937.

Dyson, N., Bernards, R., Friend, S.H., Gooding, L.R., Hassell, J.A., Major, E.O., Pipas, J.M., Vandyke, T., Harlow, E. (1990) "Large T Antigens of Many Polyomaviruses Are Able to Form Complexes with the Retinoblastoma Protein." *J. Virology* 64:1353-1356.

Eisenberg, D., Weiss, R.M., Terwilliger, T.C. (1982) "The Helical Hydrophobic Moment: a Measure of the Amphilicity of a Helix." *Nature* 299:371-374.

Eisenberg, D., Weiss, R.M., Terewilliger, T.C. (1984) "The Hydrophobic Moment Detects Periodicity in Protein Hydrophobicity." *Proc. Natl. Acad. Sci. USA.* 81:140-144.

Emi, M., Nakamura, Y., Ogawa, M., Yamamoto, T., Nishide, T., Mori, T., Matsubara, K. (1986) "Cloning, Vharacterization and Nucleotide Dequences of Ywo cDNAs Rncoding Human Pancreatic Trypsinogens." *Gene* 41:305-310.

Erman, L.D., Lesser, V.R. (1975) "A Multi-level Organization for Problem Solving Using Many Diverse, Cooperating Sources of Knowledge." *Proc. Intl. Joint Conf. Artif. Intell.* (IJCAI-4), pp. 483-490.

Fasman, G.D. (1989) *Prediction of Protein Structure and the Principles of Protein Conformation*, Plenum Press, New York, pp. 193-316.

Felsenstein, J. (1985) "Phylogenies and the Comparative Method." *Amer. Naturalist* 125(1):1-15.

Felsenstein, J. (1988) "Phylogenies from Molecular Sequences: Inference and Reliability." *Annual Rev. Genetics* 22:521-65.

Feng, D.-F., Doolittle, R.F. (1987) "Progressive Sequence Alignment As a Prerequisite to Correct Phylogenetic Trees." *J. Mol. Evol.* 25:351-360.

Figge, J., Webster, T., Smith, T.F., Paucha, E. (1988) "Prediction of Similar Transforming Region in Simian Virus 40 Large T, Adenovirus E1A, and Cyc Oncoproteins." *J. Virology,* 62(5):1814-1818.

Figge, J., Smith, T.F. (1988) "Cell-Division Sequence Motif." *Nature* 334:109.

Fischel-Ghodsian, F., Mathiowitz, G., Smith, T.F. (1990) "Alignment of Protein Sequences Using Secondary Structure: a Modified Dynamic Programming Method." *Protein Engineering* 3(7):577-581.

Friedland, P., Iwaskai, Y. (1985) "The Concept and Implementation of Skeletal Plans." *J. Autom. Reasoning* 1(2):161-208.

Garnier, J., Osguthorpe, D.J., Robson, B. (1978) "Analysis of the Accuracy and Implications of Simple Methods for Predicting the Secondary Structure of Globular Proteins." *Journal of Molecular Biology* 120(1):97-120.

Gascuel, O., Danchin, A. (1986) "Protein Export in Prokaryotes and Eukaryotes: Indications of a Difference in the Mechanism of Exportation." *J. Mol. Evol.* 24:130-142.

Goldsborough, M.D., DiSilvestre, D., Temple, G.F., Lorincz, A.T. (1989) "Nucleotide sequence of human papillomavirus type 31: a cervical neoplasia-associated virus." Virology 171(1):306-11.

Grasser, F.A., Scheidtmann, K.H., Tuazon, P.T., Traugh, J.A., Walter, G. (1988) "In Vitro Phosphorylation of SV40 Large T Antigen." *Virology* 165(1):13-22.

Gribskov, M., McLachlan, A.D., Eisenberg, D. (1987) "Profile Analysis: Detection of Distantly Related Proteins." *Proc. Natl. Acad. Sci. USA* 84:4355-4358.

Gribskov, M., Homyak, M., Edenfield, J., Eisenberg, D. (1988) "Profile Scanning for Three-Dimensional Structural Patterns in Protein Sequences." *CABIOS* 4(1):61-66.

Guigo, R., Johansson, A., Smith, T.F. (1991) "Automatic Evaluation of Protein Sequence Functional Patterns." *CABIOS* 7(3):309-315.

Hanks, S.K., Quinn, A.M., Hunter, T. (1988) "The Protein Kinase Family: Conserved Features and Deduced Phylogeny of the Catalytic Domains." *Science* 241:42-52.

Hayes-Roth, B., Buchanan, B., Lichtarge, O., Hewette, M., Altman, R., Brinkley, J., Cornelius, C., Duncan, B., Jardetzky, O. (1986) "PROTEAN: Deriving Protein Structure from Constraints." *Proc. Fifth Natl. Conf. on Artificial Intelligence,* pp. 904-909, Morgan Kaufman, Los Altos, Calif.

Hein, J. (1990) "Unified Approach to Alignment and Phylogenies." in *Methods in Enzymology,* ed. R.F. Doolittle, 183:626-645. Academic Press.

Henneke, C.M. (1989) "A Multiple Sequence Alignment Algorithm for Homologous Proteins Using Secondary Structure Information and Optionally Keying Alignments to Functionally Important Sites." *CABIOS* 5:141-150.

Hertz, G.Z., Hartzell, G.W., Stormo, G.D. (1990) "Identification of Consensus Patterns in Unaligned DNA Sequences Known to be Functionally Related." *CABIOS* 6(2):81-92.

Hodgman, T.C. (1986) "The Elucidation of Protein Function from Its Amino Acid Sequence." *CABIOS* 2:181-187.

Holbrook, S., Muskal, S., Kim, S., (1990) "Predicting Surface Exposure of Amino Acids

from Protein Sequence." *Protein Engineering* 3(8):659-665.

Hollenberg, S.M., Evans, R.M. (1988) "Multiple and Cooperative Trans-activation Domains of the Human Glucocorticoid Receptor." *Cell* 55:899-906.

Holley, L.H., Karplus, M. (1989) "Protein Structure Prediction with a Neural Network." *Proc. Natl. Acad. Sci. USA* 86:152-156.

Hope, I.A., Mahadevan, S., Struhl, K. (1988) "Structural and Functional Characterization of the Short Acidic Transcriptional Activation Region of Yeast GCN4 Protein." *Nature* 333:635-640.

Hunter, L., Harris, N., States, D. (1992) "Efficient Classification of Massive, Unsegmented Datastreams." *Proc. 10th International Machine Learning Workshop,* Morgan Kaufmann, Los Altos, CA, USA (forthcoming).

Itoh, N., Tanaka, N., Mihashi, S., Yamashina, I. (1987) "Molecular Cloning and Sequence Analysis of cDNA for Batroxobin, a Thrombin-like Snake Venom Enzyme." *J. Biol. Chem.* 262(7):3132-3135.

Karlin, S., Blaisdell B.E., Mocarski E.S., Brendel V. (1989) "A Method to Identify Distinctive Charge Configurations in Protein Sequences, with Applications to Human Herpesvirus Polypeptides." *Journal of Molecular Biology* 205:165-177.

Karlin, S., Bucher, P., Brendel, V., Altschul, S.F. (1991) "Statistical Methods and Insights for Protein and DNA Sequences." *Annu. Rev. Biophys. Biophys. Chem.* 20:175-203.

Karlin, S., Macken, C. (1991) "Some Statistical Problems in the Assessment of Inhomogeneities of DNA Sequence Data." *J. American Statistical Assoc.* 86:27-35.

King, R.D., Sternberg, M.J.E. (1990) "Machine Learning Approach for the Prediction of Protein Secondary Structure." *Journal of Molecular Biology* 216:441-457.

Kolata, G. (1986) "Trying to Crack the Second Half of the Genetic Code." *Science* 233:1037-1040.

Lander, E., Mesirov, J., Taylor, W. (1988) "Study of Protein Sequence Comparison Metrics on the Connection Machine CM-2." *Proc. Supercomputing*-88.

Lathrop, R.G. (1969) *Introduction to Psychological Research: Logic, Design, Analysis.* Harper and Row, New York.

Lathrop, R.H. (1990) Efficient Methods for Massively Parallel Symbolic Induction: Algorithms and Implementation. PhD. diss., Massachusetts Inst. of Technology, Cambridge, MA, USA.

Lathrop, R.H., Webster, T.A., Smith T.F. (1987) "ARIADNE: Pattern-directed Inference and Hierarchical Abstraction in Protein sSructure Recognition." *Communications of the ACM* 30(11):909-921.

Lathrop, R.H., Webster, T.A., Smith T.F., Winston, P.H. (1990) "ARIEL: A Massively Parallel Symbolic Learning Assistant for Protein Structure/Function." in *Artificial Intelligence at MIT: Expanding Frontiers*, ed. Winston, P.H., with Shellard, S., MIT Press, Cambridge, MA, USA.

Lathrop, R.H., Webster, T.A., Smith T.F., Winston, P.H. (1992) "Massively Parallel Symbolic Induction of Protein Structure/Function Relationships." in *Machine Learning, From Theory to Applications*, (ed.) Hanson, S., Remmele, W., Rivest, R.L., Springer-Verlag (forthcoming); reprinted from *Proc. 24th Hawaii Intl. Conf. on System Sciences* (1991), pp. 585-594, IEEE Computer Soc. Press, Los Alamitos, CA, USA.

Lawrence, C.E., Reilly, A.A. (1990) "An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences." *Proteins* 7:41-51.

Leytus, S.P., Loeb, K.R., Hagen, F.S., Kurachi, K., Davie, E.W. (1988) "A Novel Trypsin-like Serine Protease (Hepsin) with a Putative Transmembrane Domain Expressed by Human Liver and Hepatoma cells." *Biochemistry* 27:1067-1074.

Lillie, J.W., Green, M.R. (1989) "Transcription activation by the adenovirus E1a protein." *Nature* 338:39-44.

Lindsay, R., Buchanan, B., Feigenbaum, E., Lederberg, J. (1980) *The DENDRAL Project* McGraw-Hill, New York.

Ma, J., Ptashne, M. (1987) "Deletion Analysis of GAL4 Defines Two Transcriptional Activating Segments." *Cell* 48:847-853.

Maclin, R., Shavlik, J.W., (1992) "Refining Algorithms with Knowledge-Based Neural Networks: Improving the Chou-Fasman Algorithm for Protein Folding" *Machine Learning* (forthcoming).

Major, F., Turcotte, M., Gautheret, D., Lapalme, G., Fillion, E., Cedergren, R. (1991) "The Combination of Symbolic and Numerical Computation for Three-dimensional Modeling of RNA." *Science* 253:1255-60.

McGregor, M., Flores, T., Sternberg, M. (1989) "Prediction of Beta-turns in Proteins Using Neural Networks." *Protein Engineering* 2(7):521-526.

Mikes, O., Holeysovsky, V., Tomasek, V., Sorm, F. (1966) "Covalent Structure of Bovine Trypsinogen. The Position of the Remaining Amides." *Biochem. Biophys. Res. Commun.* 24(3):346-352.

Moran, E. (1988) "A Region of SV40 Large T Antigen can Substitute for a Transforming Domain of the Adenovirus E1A Product." *Nature* 334:168-170.

Munger, K., Werness, B.A., Dyson, N., Phelps, W.C., Harlow, E., Howley, P.M. (1989) "Complex Formation of Human Papillomavirus E7 Proteins with the Retinoblastoma Tumor Suppressor Gene Product." *EMBO J.* 8:4099-4105.

Muskal, S., Holbrook, S., Kim, S. (1990) "Prediction of the Disulfide-bonding State of Cysteine in Proteins." *Protein Engineering* 3(8):667-672.

Myers, E.W., Miller, W. (1989) "Approximate Matching of Regular Expressions." *Bull. Math. Biol.* 51:5-37.

Noordewier, N.O., Towell, G.G., Shavlik, J.W. (1990) "Training Knowledge-based Neural Networks to Recognize Genes in DNA Sequences." *Proc. 1990 Neural Info. Processing Conf.*

Owens, J., Chatterjee, D., Nussinov, R., Konopka, A., Maizel, J.V.J. (1988) "A Fixed Point Alignment Technique for Detection of Recurrent and Common Sequence Motifs Associated with Biological Features." *CABIOS* 4:73-77.

Patthy, L. (1987) "Detecting Homology of Distantly Related Proteins with Consensus Sequences." *Journal of Molecular Biology* 198:567-577.

Patthy, L. (1988) "Detecting Distant Homologies of Mosaic Proteins." *Journal of Molecular Biology* 202:689-696.

Pearson, W.R., Lipman, D.J. (1988) "Improved Tools for Biological Sequence Comparison." *Proc. Natl. Acad. Sci. USA* 85:2444-2448.

Ponder, J.W., Richards, F.M. (1987) "Tertiary Templates for Proteins: Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes." *Journal of Molecular Biology* 193:775-791.

Qian, N., Sejnowski, T. (1988) "Predicting the Secondary Structure of Globular Proteins Using Neural Network Models." *Journal of Molecular Biology* 202:865-884.

Ralph, W.W., Webster, T.A. Smith, T.F. (1987) “A modified Chou and Fasman protein structure algorithm.” CABIOS 3:211-216.

Rawlings, C.J., Taylor, W.R., Nyakairu, J., Fox, J., Sternberg, M.J.E. (1985) “Reasoning about protein topology using the logic programming language PROLOG.” J. Mol. Graph. 3:151-157.

Rossman, M.G., Moras, D., Olsen, K.W. (1974) “Chemical and Biological Evolution of a Nucleotide-binding Protein.” *Nature* 250:194-199.

Sankoff, D., Kruskal, J.B. (eds.) (1983) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Weslye, Reading, MA, USA.

Schuler, G.D., Altschul, S.F., Lipman, D.J. (1991) “A Workbench for Multiple Alignment Construction and Analysis.” *Proteins* 9(3):180-90.

Sellers, P.H. (1974) “On the Theory and Computation of Evolutionary Distances.” *Siam J. Appl. Math.* 26:787-793.

Sibbald, P.R., Argos, P. (1990a) “Scrutineer: a Computer Program that Flexibly Seeks and Describes Motifs and Profiles in Protein Sequence Databases.” CABIOS 6(3):279-88.

Sibbald, P.R., Argos, P. (1990b) “Weighting Aligned Protein or Nucleic Acid Sequences to Correct for Unequal Representation.” *Journal of Molecular Biology* 216(4):813-818.

Smith, H.O., Annau, T.M., Chandrasegaran, S. (1990) “Finding Sequence Motifs in Groups of Functionally related Proteins.” *Proc. Natl. Acad. Sci. USA* 87:826-830.

Smith, R.F., Smith, T.F. (1989) “Identification of New Protein Kinase-related Genes in Three Herpes Viruses, Herpes Simplex Virus, Varicella-zoster Virus, and Epstein-Barr Virus.” *J. Virology* 63:450-455.

Smith, R.F., Smith, T.F. (1990) “Automatic Generation of Primary Sequence Patterns from Sets of Related Protein Sequences.” *Proc. Natl. Acad. Sci. USA* 87:118-122.

Smith, R.F., Smith, T.F. (1992) “Pattern-induced Multi-sequence Alignment (PIMA) Algorithm Employing Structure-dependent Gap Penalties for Use in Comparitive Protein Modelling.” *Protein Eng.* 5:35-41.

Smith, T.F., Waterman, M.S. (1981a) “Comparison of Biosequences.” *Adv. Appl. Math.* 2:482-489.

Smith, T.F., Waterman, M.S. (1981b) “Identification of Common Molecular Subsequences.” *Journal of Molecular Biology* 147:195-197.

Sneath, P.H., Sokal, R.R. (1973) *Numerical Taxonomy* Freeman, San Francisco.

Staden, R. (1989) “Methods to define and locate patterns of motifs in sequences.” CABIOS 5(2):89-96.

Storey, A., Pim, D., Murray, A., Osborn, K., Banks, L., Crawford, L. (1988) “Comparison of the In Vitro Transforming Activities of Human Papillomavirus types.” *Embo. J.* 7(6):1815-1820.

Stormo, G.D. (1990) “Consensus Patterns in DNA” in *Methods in Enzymology,* ed. R.F. Doolittle, 183:211-221, Academic Press.

Stormo, G.D., Hartzell, G.W. (1989) “Identifying Protein-binding Sites from Unaligned DNA Fragments.” *Proc. Natl. Acad. Sci. USA* 86(4):1183-1187.

Taylor, W.R. (1986) “Identification of Protein Sequence Homology by Consensus Template Alignment.” *Journal of Molecular Biology* 188:233-258.

Taylor, W.R. (1988a) “A Flexible Method to Align Large Numbers of Biological Sequences.” *J. Mol. Evol.* 28:161-169.

Taylor, W.R. (1988b) “Pattern Matching Methods in Protein Sequence Comparison and Structure Prediction.” *Protein Eng.* 2:77-86.

Taylor, W.R., Thornton, J.M. (1983) "Prediction of Super-secondary Structure in Proteins." *Nature* 301:540-542.

Thornton, J.M., Flores, T.P., Jones, D.T., Swindells, M.B. (1991) "Protein Structure. Prediction of Progress at Last [News]" *Nature* 354:105-106.

Thornton, J.M., Gardner, S.P. (1989) "Protein Motifs and Data-base Searching." TIBS 14:300-304.

Towell, G.G., Shavlik, J.W., Noordewier, M.O. (1990) "Refinement of Approximate Domain Theories by Knowledge-Based Artificial Neural Networks" *Proc. Natl. Conf. on Artificial Intelligence (AAAI-90)*, pp. 861-866.

Vingron, M., Argos, P. (1989) "A Fast and Sensitive Multiple Sequence Alignment Algorithm." *CABIOS* 5:115-121.

Waterman, M.S. (1984) "General Methods of Sequence Comparison." *Bull. Math. Biol.* 46:473-500.

Waterman, M.S. (1986) "Multiple Sequence Alignment by Consensus." *Nucleic Acids Res.* 14:9095-9102.

Waterman, M.S., Jones, R. (1990) "Consensus Methods for DNA and Protein Sequence Alignment." in *Methods in Enzymology*, ed. R.F. Doolittle, 183:221-237, Academic Press.

Webster, T.A., Lathrop, R.H., Smith, T.F. (1987) "Evidence for a Common Structural Domain in Aminoacyl-tRNA Synthetases Through Use of a New Pattern-directed Inference System." *Biochemistry* 26:6950-6957.

Webster, T.A., Lathrop, R.H., Smith, T.F. (1988) "Pattern Descriptors and the Unidentified Reading Frame 6 Human mtDNA Dinucleotide-Binding Site." *Proteins* 3(2):97-101.

Webster, T.A., Lathrop, R.H., Smith, T.F. (1989) "Potential Structural Motifs in Reverse Transcriptases" *Mol. Biol. Evol.*, 6(3):317-320.

Whyte, P., Ruley, H.E., Harlow, E. (1988) "Two Regions of the Adenovirus Early Region 1A Proteins Are Required for Transformation." *Nature* 334:124-129.

Wilbur, W.J., Lipman, D.J. (1983) "Rapid Similarity Searches of Nucleic Acid and Protein Data Banks." *Proc. Natl. Acad. Sci. USA* 80(3):726-730.

Winston, P.H. (1984) *Artificial Intelligence, 2nd ed.* Addison-Wesley, Reading, MA.

Zhang, X., Mesirov, J., Waltz, D., (1992) "A Hybrid System for Protein Secondary Structure Prediction" *Journal of Molecular Biology* (to appear).

Zhu, Q., Smith, T.F., Lathrop, R.H., Figge, J. (1990) "Acid Helix-Turn Activator Motif." *Proteins* 8:156-163.