Learning Theory Toward Genome l nforrnatics

Miyano, Satoru
Research Institute of Fundamental Information Science, Kyushu University

出版情報：RIFIS Technical Report. 73, 1993-07-19. Research Institute of Fundamental Information Science, Kyushu University
バージョン：
権利関係：

RIFIS-TR-CS-73

RIFIS Technical Report

Learning Theory Toward Genome Informatics

Satoru Miyano

July 19, 1993

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
E-mail: miyano@rifis.sci.kyushu-u.ac.jp Phone: 092-641-1101 ex. 4472

Learning Theory Toward Genome Informatics

Satoru Miyano

Research Institute of Fundamental Information Science
Kyushu University 33, Fukuoka 812, Japan
miyano@rifis.sci.kyushu-u.ac.jp

Abstract. This paper discusses some problems in Molecular Biology to which learning paradigms may be applicable. As a case, we present our recent study on knowledge discovery from amino acid sequences by PAC-learning paradigm.

1 Introduction

Computer Science and Molecular Biology is now creating a rapidly evolving research field called *Genome Informatics* or *Molecular Bioinformatics*. Rapid advances of biotechonolgy in the last decade and the international Human Genome Project are putting various databases of proteins and nucleic data into explosion. Everyday, a considerable amount of genomic data from laboratories are compiled into the databases via international networks.

One of the important issues in Genome Informatics is to establish technologies for discovering knowledge from amino acid sequences and nucleic sequences that may provide new directions of experimental investigations to biologists. Systematic and efficient methods are strongly desired for discovering or acquiring important biological knowledge from these genomic data.

The purpose of this paper is to show a way of getting into Genome Informatics from the side of learning theory in the above respect by (a) showing available databases of genomic data, (b) defining problems on these data, (c) giving their related works so far, and (d) presenting the trace of our approach [1, 31, 38, 39] to some of the problems.

The research on gathering, analyzing, and managing amino acid sequences of proteins and nucleic acid sequences has a considerably long tradition since the birth of Molecular Biology. On the other hand, Computer Science seems to have made relatively little effort on this topic while it has been contributing to various fields with drastic changes. Genome Informatics involves nearly every aspect of Computer Science. In particular, computational challenges of learning theory are really desired for the next advance, and the potential payoff is very high both in Computer Science and Molecular Biology. If algorithmic/computational learning theory could survive through this big trend in Genome Informatics and develop soundly, it would establish its firm identity in Science.

This paper is organized as follows: Section 2 surveys some databases which provides information about amino acid sequences of proteins and nucleic acid sequences. In Section 3, we pick up some problems to which learning technologies are applicable. Section 4 reviews the methods so far developed for these problems. In Section 5, we

present our research strategy using PAC-learning paradigm for the transmembrane domain identification problem and the signal peptide identification problem. We also sketch the machine discovery system BONSAI that is developed in this research and show its successful experimental results.

2 Databases for Genome Informatics

This section provides some useful databases which compile amino acid sequences and nucleic acid sequences with their associated information. All databases below are, for instance, accessible at Human Genome Center, Institute of Medical Science, The University of Tokyo, by ftp network service.

1. **GenBank**
   This database contains nucleic sequences from published articles and direct submissions from authors. Each record specifies the source of the data and related keywords. Translations of coding regions are also available in this database.
2. **PDB**
   Three-dimensional structures of proteins and nucleic acids are compiled in PDB. These are X-ray crystallographic and NMR data provided by direct submissions from their authors.
3. **PIR**
   PIR (Protein Identification Resources) contains amino acid sequences of proteins from published articles and their related information.
4. **EMBL**
   EMBL collects nucleic acid sequences from published articles and also by direct submission from authors.
5. **SWISS-PROT**
   Protein sequences and translations of coding regions from EMBL are compiled in this database.
6. **PROSITE**
   This database collects motifs for proteins. This is a kind of a dictionary of protein sites and patterns.

3 Problems

This section provides some problems in Genome Informatics to which learning technologies may be helpful. The final purpose is to establish techniques by employing learning paradigms for extracting biologically relevant information from amino acid sequences of proteins and nucleic acid sequences stored in the databases shown in Section 2. In practice, the problem is to determine regions or locations on a sequence which have specific functions or properties. Furthermore, it is also an important issue to discover biologically interesting knowledge from the databases by learning technologies.

In the following discussion, we may ignore biological correctness of explanations since there are lot of exceptions in biological data. Instead, we put priority on their understandability of the contents. The text books by Watson et al. [47] and Lewin [30] are very helpful to capture the general principles in Molecular Biology.

3.1 Protein Structure Prediction

The *primary structure* of a protein is described as a sequence of amino acid residues. It is generally accepted that the amino acid sequences determine their forms and functions.

The *secondary structure* of a protein describes the local structure of the protein. There are three kinds of major secondary structures; *α-helix*, *β-sheet*, and *turn*. The α-helix is a helical conformation of the polypeptide backbone that is stereochemically most satisfactory. Every turn of the helix consists of 3.6 amino acids. An α-helix comprises of $4 \sim 30$ amino acids. The β-sheet is a regular lateral association of extended polypeptide chains into a sheet. The *turn* is a structure like a hairpin.

The *secondary structure prediction problem* is to predict the locations of α-helices, β-sheets and turns on the amino acid sequence of proteins.

The *tertiary structure* of a protein is the form in the three-dimensional space. Its structure is, in many cases, complex. It is also an important problem to predict possible tertiary structures of a protein form its primary structure. This problem is called the *tertiary structure prediction problem.*

PDB collects tertiary structures of proteins. Unfortunately, the number of proteins whose tertiary structures are known and open to public is small.

The following special issues have attracted many attentions.

Membrane Spanning Segments A membrane has a structure of lipid bilayer that surrounds the cytoplasm of a cell. A membrane of a cell usually contains various kinds of protein molecules directly inserted into the lipid bilayer. Some of these proteins have parts sticking out from both sides of the membrane and pass through this bilayer several times. Regions of a protein passing right through the membrane are called the *transmembrane domains*, each of which is a sequence of length about 20 constituting an α-helix structure. It has been accepted to assume that a protein containing these transmembrane domains is a membrane protein. The length of an α-helix in a membrane protein comes from the length to span the apolar center of a membrane [13]. In a typical membrane, the apolar center is approximately 30Å, and each amino acid residue has a thickness of 1.54Å. Hence 20 residues$\times$1.54Å/residue = 31Å. If sequences corresponding to α-helices of transmembrane domains are found in an amino acid sequence of a protein, it is very likely that the protein is a membrane protein.

The *transmembrane domain identification problem* is, given an amino acid sequence of a protein, to decide if it contains transmembrane domains and to show the regions of transmembrane domains.

PIR provides amino acid sequences of proteins with FEATURE field where transmembrane domains are indicated as shown in Fig. 1.

Signal Peptides There are proteins that shall be conveyed across the membrane to the outside of the cell. Most of such proteins are synthesized in the form of longer precursors that consist of additional amino acid sequences of length $15 \sim 30$ at their N-termini. A *signal peptide* is a sequence locating at this N-terminal region. This sequence usually begins with a Methionine (`M`).

```
ENTRY          MPHUPL    #Type Protein
TITLE          Myelin proteolipid protein - Human
ALTERNATE-NAME  PLP lipophilin
DATE           30-Sep-1987 #Sequence 30-Sep-1987 #Text 30-Sep-1991
PLACEMENT       1081.0   1.0   1.0   1.0   1.0
SOURCE         Homo sapiens #Common-name man
ACCESSION      A24657
REFERENCE
  #Authors     Stoffel W., Giersiefen H., Hillen H., Schroeder W.,
               Tunggal B.
  #Journal     Biol. Chem. Hoppe-Seyler (1985) 366:627-635
  #Reference-number A24657
  #Accession   A24657
  #Molecule-type protein
  #Residues    1-276 [STO]
COMMENT        Five hydrophobic domains are present; three are
               transmembrane domains and two are intramembranous
               domains.
COMMENT        Most, and perhaps all, of the Cys residues are
               involved in disulfide bonds.
COMMENT        This is the major protein of myelin from the central
               nervous system.
SUPERFAMILY    #Name myelin proteolipid protein
KEYWORDS       membrane protein myelin oligodendrocyte
               structural protein
FEATURE
  9-34              #Domain transmembrane I [TM1]
  59-96             #Domain transmembrane II [TM2]
  151-190           #Domain hydrophobic intramembranous I
                    [HC1]
  205-216           #Domain hydrophobic intramembranous II
                    [HC2]
  232-267           #Domain transmembrane III [TM3]
  198               #Binding-site fatty acid
SUMMARY        #Molecular-weight 29890  #Length 276  #Checksum   559
SEQUENCE
          5       10       15       20       25       30
    1 G L L E C C A R C L V G A P F A S L V A T G L C F F G V A L
   31 F C G C E V E A L T G T E K L I E T Y F S K N Y Q D Y E Y L
   61 I N V I H A F Q Y V I Y G T A S F F F L Y G A L L L A X G F
   91 Y T T G A V R Q I F G D Y K T T I C G K G L S A T V T G G Q
  121 K G R G S R G Q H Q A H S L E R V C H C L G C W L G H P D K
  151 F V G I T Y A L T V V W L L V F A C S A V P V Y I Y F N T W
  181 T T C Q S I A A P C K T S A S I G T L C A D A R M Y G V L P
  211 W N A F P G K V C G S N L L S I C K T A E F Q M T F H L F I
  241 A A F V G A A A T L V S L L T F M I A A T Y N F A V L K L M
  271 G R G T K F
```

Fig. 1. An amino acid sequence in PIR containing four transmembrane domains.

GenBank provides nucleic sequences together with information about signal peptides. An example from GenBank is shown in Fig. 2 that is a DNA sequence containing a signal peptide.

The *signal peptide identification problem* is, given a DNA sequence, to decide if it contains a sequence representing a signal peptide at the N-terminal region and to identify the location of the signal peptide.

```
LOCUS       AFABCP       783 bp ds-DNA            BCT      15-MAR-1989
DEFINITION  A.faecalis blue copper protein gene, complete cds.
ACCESSION   M18267
KEYWORDS    anaerobic nitrate respiration system regulator;
            blue copper protein; electron carrier; pseudoazurin.
SOURCE      A.faecalis S-6 DNA, clone pAB101.
  ORGANISM  Alcaligenes faecalis
            Prokaryota; Bacteria; Gracilicutes; Scotobacteria;
            Aerobic rods and cocci; Alcaligenaceae.
REFERENCE   1  (bases 1 to 783)
  AUTHORS   Yamamoto,K., Uozumi,T. and Beppu,T.
  TITLE     The blue copper protein gene of Alcaligenes faecalis S-6 directs
            secretion of blue copper protein from Escherichia coli cells
  JOURNAL   J. Bacteriol. 169, 5648-5652 (1987)
  STANDARD  simple staff review
FEATURES             Location/Qualifiers
     CDS             249..689
                     /note="blue copper protein precursor"
                     /codon_start=1
                     /translation="MRNIAIKFAAAGILAMLAAPALAENIEVHMLNKGAEGAMVFEPA
                   YIKANPGDTVTFIPVDKGHNVESIKDMIPEGAEKFKSKINENYVLTVTQPGAYLVKCT
                     PHYAMGMIALIAVGDSPANLDQIVSAKKPKIVQERLEKVIASAK"
     sig_peptide     249..317
                     /note="blue copper protein signal peptide"
                     /codon_start=1
     mat_peptide     318..686
                     /note="blue copper protein"
                     /codon_start=1
BASE COUNT       207 a    204 c    216 g    156 t
ORIGIN      5 bp upstream of SphI site.
        1 gcatgcaggc ttgtcatgtc gcgcctaggc tggccggagg ctgcggcaaa agggctggcg
       61 ggcatatcga atggtgcgat ggtggaatgc agaaaatcaa acagttttat tctgagcgtc
      121 attatcatga ataacttcac ctgtttgatc cagatcaaag aggttggcag gcgacaggtc
      181 taaaccccgt tacgtggcgt gttgaggccg agacggcagg tgcgccaatc gttggagatc
      241 aggacaaaat gcgtaacatc gcgatcaaat ttgctgccgc aggcatcctc gccatgctgg
      301 ctgcccccgc tcttgccgaa aatatcgaag ttcatatgct caacaagggc gccgagggcg
      361 ccatggtttt cgagcctgcc tatatcaagg ccaatcccgg cgacacggtc acctttattc
      421 cggtggacaa aggacataat gtcgaatcca tcaaggacat gatccctgaa ggcgccgaaa
      481 agttcaaaag caagatcaac gagaactatg tgctgacggt tacccagccc ggcgcatatc
      541 tggtaaagtg cacaccgcat tatgccatgg gtatgatcgc gctcatcgct gtcggtgaca
      601 gcccggccaa tctcgaccag atcgtttcgg ccaagaagcc gaagattgtt caggagcggc
      661 tggaaaaggt catcgccagc gccaaataag agcgccaaat aagattgacc gaaaactctc
      721 gatgagccga acttgaaccg gcttcatgac gaggacatca tgaccagaca gccaggcctg
      781 cag
```

Fig. 2. A DNA sequence containing a signal peptide.

Epitopes This problem involves an aspect related to the tertiary structures of proteins. An *epitope* is also called an *antigenic determinant*. Each epitope consists of $5 \sim 8$ amino acids and locates on the protein surface. The expected number of antigenic determinants is at least $10,000$ and probably much more although the number of known epitopes is not large. PIR and GenBank provide information about eptiope sequences.

3.2 Coding Region Identification

For prokaryotic genes such as bacterial genes, there is the colinearity, i.e., a direct correspondence between the sequence of a gene, its mRNA and its protein product.

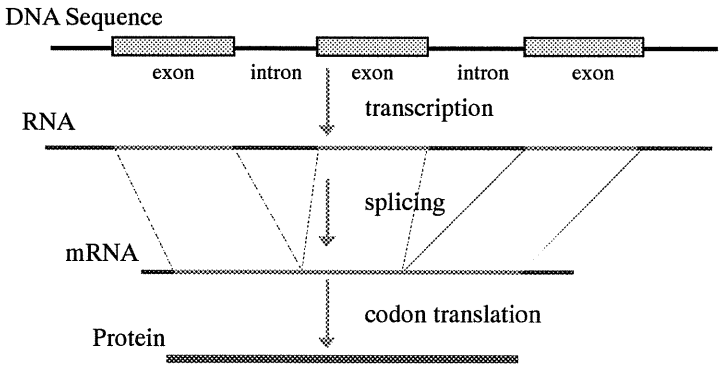

Fig. 3. The gene is transcribed into a very large precursor RNA molecule, from which the introns are removed and the exons are concatenated to form a mRNA.

On the other hand, the nucleotides sequences in eukaryotes that specify amino acid sequences of proteins are usually interrupted by blocks of nucleotides that have no coding information. The segments that encode amino acids for proteins are called *exons* and the blocks that are not translated into amino acids and removed by the process called *RNA splicing* are called *introns* (Fig. 3). There is no specific range of the lengths of exons and introns. Moreover, the number of exons varies gene to gene. The *splicing site identification problem* is to find locations of exon/intron boundaries in a given DNA sequence. Information about these data are provided in GenBank.

The *coding region* is a region of a DNA sequence which is finally translated by ribosomes into a polypeptide. In prokaryotes, the coding region is a single contiguous sequence (*open reading frame* (ORF)), while the coding region of a eukaryotic gene comprises of several disjoint ORFs since the sequence is interrupted by introns. The problem of determining the coding region of a DNA sequence is one of the most challenging problems in Genome Informatics.

Translation of mRNA encoding a protein starts at some location and stops at some location. These locations are also encoded in a DNA sequence. Three codons (`UAA`, `UAG`, and `UGA`) are stop signals where translation stops. These codons do not correspond to any amino acids. The start signal is usually `AUG` that encodes Methionine (`M`). However, the converse is not true. Namely, the region starting with `AUG` is not necesarry the starting location of translation. The coding region is delimited by these start and stop signals.

In the upstream of the start signal, there is a region called a *promoter* that is recognized by RNA polymerase to locate the start site of RNA transcription on DNA template. The problem of searching for promoters in DNA sequences is still difficult. Promoter sites of E. coli RNA polymerase are compiled in [27]. Promoters in eukaryotes are collected in the Eukaryote Promoter Database (EPD) [8]. An important problem about promoters is to characterize promoter sequences and sites. GenBank is also useful to collect promoters.

4 Methods

This section surveys some techniques for solving problems given in Section 3 with bibliographic data. Most of the techniques are not directly related to learning theory, but may give useful suggestions to the research on these problems by learning paradigms. The text book [18] may be helpful to understand the overview and current state of research.

4.1 Protein Structure Prediction

For the secondary structure prediction problem, there have been proposed several methods to predict the locations of *α-helices*, *β-sheets* and *turns* from amino acid sequences of proteins. The prediction accuracy remains in 55–65%. The evaluation of accuracy strongly depends on the choice of data. Therefore, exact comparison is rather difficult by the data presented in published articles.

The most common strategy is the use of statistical information about amino acid residues appearing in α-helices, β-sheets and turns. Chou and Fasman [9] and Garnier et al. [15] employed this strategy. By analyzing the sequence patterns of the various types of secondary structure segments, Cohen et al. [10] proposed a turn prediction method. The Neural network approach is currently the most successful one. There are a bunch of neural network approaches and their experiments: Qian and Sejnowski [35], Holley and Karplus [21], Kneller at al. [26], etc. We do not enumerate those works here. The strategy of hidden Markov model also achieved good accuracy of prediction: [3, 19]. There are some interesting approaches to protein secondary structure prediction [1, 11, 39, 43]. Our approach by PAC-learning paradigm [1, 39] also attained the former accuracy but did not exceed drastically.

Transmembrane Domain Identification The most common method for prediction transmembrane domains is the hydropathy plot of the amino acid residues of the sequence. The hydropathy index due to Kyte and Doolittel [28] works quite well. Yanagihara [48] developed a very accurate method for this problem. Section 5 presents our approach by PAC-learning paradigm.

Signal Peptides Signal peptides have the hydrophobic nature [45, 46]. Ladunga et al. [29] used neural networks to predict signal peptide regions. There have been developed some software packages for this problem [14, 34]. Our approach by PAC-learning paradigm is shown in Section 5.

Epitopes Antigenic regions locate on the surfaces of proteins and are formed by two or more segments. Hopp and Woods [22] used the hydrophilicity for amino acids and developed a method to predict antigenic regions by plotting the hydrophilicity of the sequence. Jameson and Wolf [24] defined the antigenicity index by a linear combination of hydrophilicity, predicted side chain flexibility [25], surface probability [12], turn prediction by Chou-Fasman [9] and Garnier et al. [15].

4.2 Exon-Intron Boundary Prediction

In coding region identification problem, the most important problem is to predict the *splice junctions* which are boundaries of exons and introns. Recently, neural network approached have been developed [6, 7]. Brunak et al. [7] used neural networks to predict splice site location in human pre-mRNA. Most of the approaches are based on statistical analysis of patterns [16, 20, 23, 32, 37]. Senapathy et al. [37] have analyzed the base frequency distributions for the donor and acceptor sites for the the general categories in GenBank (for these categories, see Section 5.5.2). There are a lot of works on this problem including software packages [41, 42].

5 Machine Discovery by PAC-Learning Paradigm

In this section, we present our approach by PAC-learning paradigm and some algorithmic strategies [1, 39] to the signal peptide identification problem and the transmembrane domain identification problem.

5.1 Polynomial-Time PAC-Learnability

Let us begin with basic terminology about PAC-learning. We call a subset of Σ^* a *concept*. A *concept class* $\mathcal{C}$ is a nonempty collection of concepts. For a concept $c \in \mathcal{C}$, a pair $\langle w, c(w)\rangle$ is called an example of c for $w \in \Sigma^*$, where $c(w) = 1$ ($c(w) = 0$) if w is in c (is not in c). For an alphabet Σ and an integer $n \geq 0$, $\Sigma^{\leq n}$ denotes the set $\{w \in \Sigma^* \mid |w| \leq n\}$.

We assume a representation system R for concepts in $\mathcal{C}$. We use a finite alphabet Λ for representing concepts. For a concept class $\mathcal{C}$, a *representation* is a function $R : \mathcal{C} \to 2^{\Lambda^*}$ such that $R(c)$ is a nonempty subset of Λ^* for c in $\mathcal{C}$ and $R(c_1) \cap R(c_2) = \emptyset$ for any distinct concepts c_1 and c_2 in $\mathcal{C}$. For each $c \in \mathcal{C}$, $R(c)$ is the set of *names* for c.

A concept class $\mathcal{C}$ is said to be *polynomial-time learnable* [5, 33, 44] if there is an algorithm $\mathcal{A}$ which satisfies (1) and (2).

(1) $\mathcal{A}$ takes a sequence of examples as an input and runs in polynomial-time with respect to the length of input.
(2) There exists a polynomial $p(\cdot,\cdot,\cdot)$ such that for any integer $n \geq 0$, any concept $c \in \mathcal{C}$, any real numbers ε, δ $(0 < \varepsilon, \delta < 1)$, and any probability distribution P on $\Sigma^{\leq n}$, if $\mathcal{A}$ takes $p(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$ examples which are generated randomly according to P, then $\mathcal{A}$ outputs, with probability at least $1 - \delta$, a name of a hypothesis h with $P(c \oplus h) < \varepsilon$.

5.2 Learning Decision Trees over Regular Patterns

Let us explain our approach [1, 39] to, for example, the signal peptide identification problem. Let $\mathcal{R} = \{\mathtt{A}, ..., \mathtt{Z}\}$ be the alphabet consisting of 20 symbols of amino acid residues and additional symbols for representing ambiguity. From GenBank, we have collected 1032 sequences of amino acids for primate signal peptides. Namely, these sequences are strings over $\mathcal{R}$. These data have been provided by experiments by

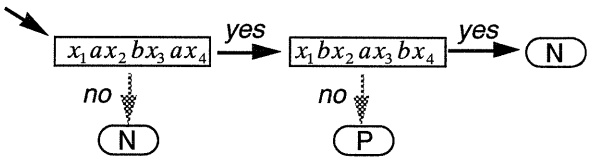

Fig. 4. Decision tree over regular patterns defining the language $\{a^m b^n a^l \mid m, n, l \geq 1\}$ over $\Sigma = \{a, b\}$.

biologists who have their own interests. Hence we may regard that the data are coming out by following some unknown probability distribution.

The objective is to find a "short explanation" of these sequences which may have a sense in Molecular Biology. Then by using the knowledge described in the "short explanation", we may predict signal peptides in the sequences. We also need to prepare negative examples for signal peptides. This is discussed in the sequel.

For applying PAC-learning paradigm to this problem, the first thing we have to do is to define a target concept class $\mathcal{C}$ in which there may be a concept representing the primate signal peptides. Discussions with molecular biologists would be helpful to define a target concept class. Then for this concept class $\mathcal{C}$, we have to find a learning algorithm running in feasible time. The following result is useful to show the polynomial-learnability of $\mathcal{C}$:

Theorem 1. [33] A concept class $\mathcal{C}$ is polynomial-time learnable if $\mathcal{C}$ is of polynomial dimension and there is a polynomial-time fitting for $\mathcal{C}$.

In analyzing of amino acid sequences and nucleic sequences, "motifs" are often searched that are common to the given sequences as complied in PROSITE [4]. For example, the expression `C-x(2,4)-C-x(12)-H-x(3,5)-H` is called the zinc finger motif. These "motifs" can be regarded as a kind of simple regular expressions. By this observation, we use *regular patterns* [40] as the view by which we analyze the sequences.

Formally, a regular pattern is an expression of the form $\pi = \alpha_0 x_1 \alpha x_1 \cdots x_n \alpha_n$, where $\alpha_0, ..., \alpha_n$ are strings over an alphabet Σ and $x_1, ..., x_n$ are mutually distinct variables to which arbitrary strings in Σ^* are substituted. Thus it defines a regular language over Σ. We denote its defining language by $L(\pi)$ (we allow ε-substitutions). A regular pattern containing at most k variables is called a *k-variable regular pattern.*

Now we have established the basic view on the sequences. The next step is to combine these regular patterns in order to make a concept. The idea is to use a decision tree whose nodes have regular patterns as their labels for classification. We call such a tree a *decision tree over regular patterns*, which is a binary tree T such that each leaf is labeled with class name N (negative) or P (positive) and each internal node is labeled with a regular pattern (see Fig. 4). Let $L(T)$ be the set of strings that are classified as P. Obviously, $L(T)$ is also a regular language.

For integers $k, d \geq 0$, we denote by $DTRP(d, k)$ the class of languages defined by decision trees over k-variable regular patterns with depth at most d.

We have shown in [1] the following theorem:

Theorem 2. [1] $DTRP(d, k)$ is polynomial-time learnable for all $d, k \geq 0$.

We have discussed in [31] that the polynomial-time learnability requires the constant bound k on the number of variables.

5.3 Indexing of Amino Acid Residues

Kyte and Doolittle [28] have given the hydropathy index for amino acid residues in their study of membrane spanning segments. Each amino acid residue is assigned a real number between $-4.5 \sim 4.5$. According to the hydropathy index, we have classified 20 kinds of amino acid residues into three categories $\{*, +, -\}$ as shown in Table 1 and found that the transformation of amino acid residues by Table 1 makes learning drastically efficient in the experiments [2, 1]. Suprizingly, the transmembrane domain sequences transformed by Table 1 have only a few overlaps with nontransmembrane domain sequences. Hence, the information about positiveness and negativeness is not lost by this transformation.

Amino Acids	Hydropathy Index	New Symbol
A M C F L V I	$1.8 \sim 4.5$ $\rightarrow$	*
P Y W S T G	$-1.6 \sim -0.4$ $\rightarrow$	+
R K D E N Q H	$-4.5 \sim -3.2$ $\rightarrow$	-

Table 1. Categorization of amino acid residues by the hydropathy index

This observation led us to the following notion:

Definition 3. Let Σ be a finite alphabet and P and N be two disjoint subsets of Σ^*. Let Γ be a finite alphabet, called an *indexing alphabet*, satisfying $|\Sigma| > |\Gamma|$. An *indexing* ψ of Σ by Γ with respect to P and N is a mapping $\psi : \Sigma \rightarrow \Gamma$ such that $\tilde{\psi}(P) \cap \tilde{\psi}(N) = \emptyset$, where $\tilde{\psi} : \Sigma^* \rightarrow \Gamma^*$ is the homomorphism defined by $\tilde{\psi}(a_1 \cdots a_n) = \psi(a_1) \cdots \psi(a_n)$ for $a_1, \ldots, a_n \in \Sigma$.

In the above definition, the indexing ψ with respect to P and N must satisfy the condition $\tilde{\psi}(P) \cap \tilde{\psi}(N) = \emptyset$. However, this condition is too strong for practical use because we have the following theorem:

Theorem 4. [38] The problem of finding an indexing is NP-complete.

It should be also stated that the data from the PIR and GenBank databases contains some amount of errors. Therefore it is reasonable to relax the condition of indexing so that $\tilde{\psi}(P)$ and $\tilde{\psi}(N)$ have a few overlaps.

For the representation of a concept, we use a pair (T, ψ) of a mapping $\psi : \mathcal{R} \rightarrow \Gamma$ and a decision tree T over regular patterns, where constant strings appearing in the patterns are taken from Γ^*. Strings in $\mathcal{R}^*$ are first transformed by ψ to strings in Γ and then classified by T. The target concept class $\mathcal{C}$ consists of the concepts over $\mathcal{R}^*$ defined by such pairs (T, ψ).

5.4 Algorithms for Decision Trees and Indexing

Although the learning algorithm showing Theorem 2 runs in polynomial time, the running time is enormous and the algorithm does not have any sense in practical use. This is the reason why we developed in [1] a heuristic algorithm for learning a decision tree over regular patterns.

Based on these ideas, we have developed a machine learning system BONSAI that shall produce, from sets of positive and negative examples, a pair of an indexing and a decision tree over regular patterns.

This section gives algorithms for finding approximate indexings and constructing decision trees over regular patterns that are implemented in BONSAI.

Let POS and NEG be two disjoint subsets of $\mathcal{R}^*$, where POS is,in practice, the set of all collected positive examples and NOG is the set of suitably selected negative examples. From POS and NEG, two small subsets pos and neg are randomly chosen for training sets. Let Γ be a finite alphabet for indexing.

By following [1], we briefly review how to construct decision trees over regular patterns from pos and neg of positive and negative training examples. We employed the idea of ID3 algorithm [36] for constructing a decision tree. The ID3 algorithm assumes examples specified with explicit attributes in advance. On the other hand, our approach assumes a space of regular patterns which are simply generated by given pos and neg. The algorithm tries to find appropriate regular patterns from this space of attributes dynamically during the construction of a decision tree. Thus the view to the data determines the space of attributes. This is a point which is very suited for our empirical research.

Let P and N be the sets of strings obtained by transforming pos and neg of the training examples by some indexing ψ of $\mathcal{R}$ by Γ. Notice that P and N may have some intersection. Using P and N, we deal with regular patterns of the form $\alpha_0 x_1 \alpha_1 x_2 \cdots x_k \alpha_k$ such that $\alpha_0, ..., \alpha_k$ are substrings of some strings in $P \cup N$. Let $\Pi(P, N)$ be some family of such regular patterns made from P and N. The family $\Pi(P, N)$ is appropriately given and used as a space of attributes.

For a regular pattern $\pi \in \Pi(P, N)$, the cost $E(\pi, P, N)$ is the one defined in [36] by

$$E(\pi, P, N) = \frac{p_1 + n_1}{|P| + |N|} I(p_1, n_1) + \frac{p_0 + n_0}{|P| + |N|} I(p_0, n_0),$$

where p_1 (resp. n_1) is the number of positive examples in P (resp. negative examples in N) that match π, i.e., $p_1 = |P \cap L(\pi)|$, $n_1 = |N \cap L(\pi)|$, and p_0 (resp. n_0) is the number of positive examples in P (resp. negative examples in N) that do not match π, i.e., $p_0 = |P \cap \overline{L(\pi)}|$, $n_0 = |N \cap \overline{L(\pi)}|$, $\overline{L(\pi)} = \Sigma^* - L(\pi)$, and

$$I(x, y) = \begin{cases} 0 & (\text{if } x = 0 \text{ or } y = 0) \\ -\frac{x}{x+y} \log \frac{x}{x+y} - \frac{y}{x+y} \log \frac{y}{x+y} & (\text{otherwise}). \end{cases}$$

Algorithm *MakeTree*(P, N) sketches the decision tree algorithm for $\Pi(P, N)$, where *Create*(π, T_0, T_1) returns a new tree with a root labeled with π whose left and right subtrees are T_0 and T_1, respectively.

The above algorithm is sufficiently fast and experiments show that small enough trees are very often produced. When we consider the class $DTRP(k, d)$, we have to specify k of variables in a pattern and d of the depth of a decision tree. However,

```
function MakeTree( P, N : sets of strings ): node;
  begin
    if N = ∅ then return( Create("P", null, null) )
    else if P = ∅ then return( Create("N", null, null) )
    else begin
      Find a shortest pattern π in Π(P, N) that minimizes E(π, P, N);
      P1 ← P ∩ L(π);  P0 ← P − P1;
      N1 ← N ∩ L(π); N0 ← N − N1;
      if (P0 = P and N0 = N) or (P1 = P and N1 = N) then
                        /* No more division is possible */
          return( Create("P", null, null) )
      else return( Create(π, MakeTree(P0, N0), MakeTree(P1, N1) ) )
    end
  end
```

Algorithm 1: *MakeTree*

we can not know in advance which k and d are appropriate for the signal peptide problem. Therefore, as long as small hypotheses are produced, we may be satisfied with the results.

Now we describe the algorithm that finds indexings. It searches for a locally optimal indexing using a score function $Score(\psi)$ that calls *MakeTree* as a subroutine. Let Ψ be the set of all indexings of $\mathcal{R}$ by Γ. Let POS be the set of positive examples and NEG be the set of negative examples from $\mathcal{R}^*$. Let pos and neg be the sets of positive and negative training examples. To evaluate the "goodness" of an indexing $\psi \in \Psi$, it constructs a tree T by running the procedure $MakeTree(\tilde{\psi}(pos), \tilde{\psi}(neg))$ and then evaluates the success rate that T explains $\tilde{\psi}(POS)$ and $\tilde{\psi}(NEG)$ correctly. $Score(\psi)$ is determined by $Verify(MakeTree(\tilde{\psi}(pos),\tilde{\psi}(neg)),\tilde{\psi}(POS),\tilde{\psi}(NEG))$, where the function *Verify*(T, Pos, Neg) returns the value $\sqrt{\frac{|L(T)\cap Pos|}{|Pos|} \cdot \frac{|\overline{L(T)}\cap Neg|}{|Neg|}}$, which represents the geometric mean of the success rates of the decision tree T for positive examples Pos and negative examples Neg.

For indexings ψ and ϕ in Ψ, the distance between ψ and ϕ is defined by $d(\psi, \phi) = |\{\sigma \in \Sigma \mid \psi(\sigma) \neq \phi(\sigma)\}|$. The neighbors of ψ are the indexings whose distance from ψ is one. Algorithm *FindGoodIndex* begins with an indexing ψ which is selected randomly from Ψ. Then it searches an indexing ϕ from its neighbors such that its score is the best among the neighbors of ψ. This process continues until no better indexing is found from its neighbors. The strategy of the algorithm is sketched in Algorithm *FindGoodIndex*.

These two algorithms are combined and related as shown in Fig. 5. The following two subsections describe how to apply BONSAI to acquire knowledge about transmembrane domains and signal peptides.

5.5 Experiments

We have to specify two parameters in BONSAI. The first is the size of the small sets pos and neg which shall be chosen from POS and NEG at random. The second

```
function FindGoodIndex(POS,NEG: sets of strings) indexing;
  begin
    Select small subsets pos of POS and neg of NEG randomly;
    Select randomly ψ0 from Ψ;
    repeat
      φ ← ψ;
      for each φ' ∈ Ψ with d(ψ, φ') = 1 do
        if Score(φ') > Score(φ) then φ ← φ';
    until ψ = φ;
    return ψ;
  end
```

Algorithm 2: *FindGoodIndex*

is the size of the indexing alphabet. With these parameters, the system with will produce a hypothesis (T, ψ) consisting of a decision tree T over regular patterns and an indexing ψ. The accuracy of (T, ψ) for *POS* and *NEG* is represented by a pair $(p\%, n\%)$, where $p\%$ (resp. $n\%$) of *POS* (resp. *NEG*) are recognized as positive (resp. negative).

In the following experiments, we set the size of *pos* and *neg* to be $|pos| = |neg| = 10$ and the indexing alphabet Γ has 2∼3 symbols.

Moreover, in order to avoid combinatorial explosions, we also assume that the regular patterns attached to the nodes of decision trees are of the form $x\alpha y$, where x and y are variables and α is a substring taken from *pos* and *neg*. There is no other special reason why we used only these regular patterns except that they are the simplest.

Transmembrane Domains The sequences corresponding to the transmembrane domains are positive examples. Negative examples are sequences of length around 30 taken from the parts other than transmembrane domains. For example, in the sequence shown in Fig. 1, there are three transmembrane domains as is indicated in its FEATURE field as below. These are taken as positive examples.

```
FEATURE
    9-34                    #Domain transmembrane I
    59-96                   #Domain transmembrane II
    232-267                 #Domain transmembrane III
```

As negative examples, we use the amino acid sequences located in other parts than transmembrane domains. For instance, the sequence $w[100..130]$ is a possible negative example, while $w[46..75]$ is not, because the terminal segment of $w[46..75]$ is located in a transmembrane domain $w[59..96]$.

We collect all the positive examples from the PIR database. The number of positive examples is 689. We use 19256 negative examples randomly chosen. These sequences form the sets *POS* and *NEG* for the transmembrane domain identification problem.

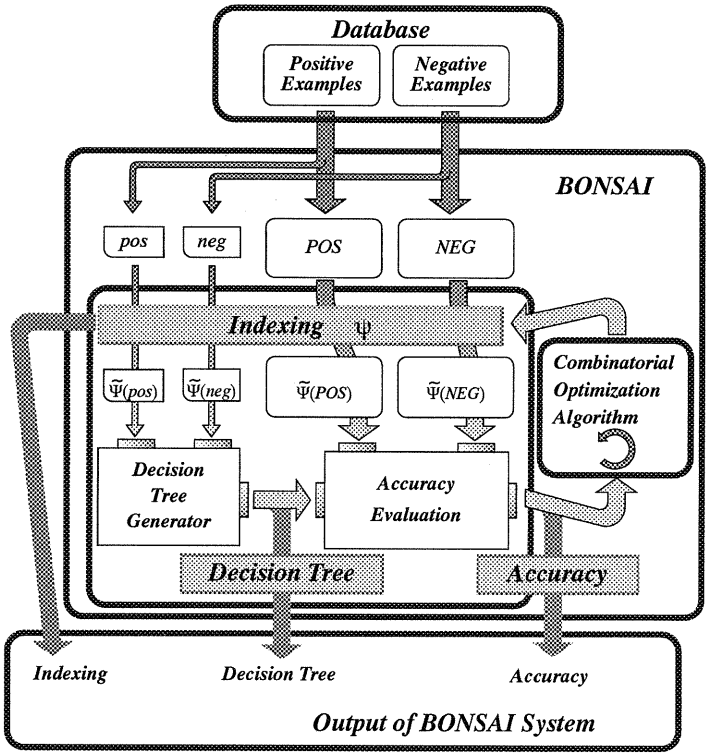

Fig. 5. BONSAI

We have reported in [39] that BONSAI has discovered a decision tree over regular pattern with just three internal nodes with accuracy more than 92%. More interestingly, the indexing associated with the decision tree exactly corresponds to the hydropathy index of amino acids due to Kyte and Doolittle [28] except Asparagine (N) as shown in Table 2. Biologists gave a comment that the exception of N is reasonable.

Table 2. Indexing discovered by BONSAI and its correspondence with the hydropahty index.

Amino Acid	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
New Symbol	0	0	1	1	0	0	1	0	1	0	0	0	1	1	1	0	0	0	0	0
Hydropathy	1.8	2.5	-3.5	-3.5	2.8	-0.4	-3.2	4.5	-3.9	3.8	1.9	-3.5	-1.6	-3.5	-4.5	-0.8	-0.7	4.2	-0.9	-1.3

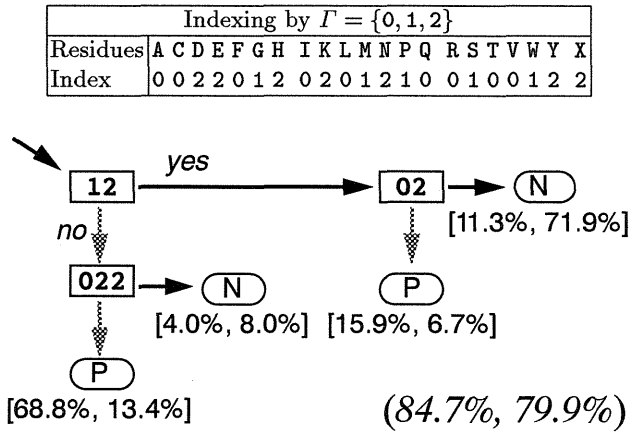

Fig. 6. The indexing and the decision tree for the primate signal peptides. The pair $[p\%, n\%]$ attached to a leaf means that $p\%$ of 1032 positive examples (resp. $n\%$ of 3162 negative examples) reached the leaf.

Signal Peptides We use the GenBank database. Signal peptides are indicated in the FEATURE field as shown in Fig. 2. We collected as positive examples the signal peptide sequences beginning with a Methionine (M). The negative examples are N-terminal regions of length 30 obtained from complete sequences that have no signal peptide and begin with M. The following table shows the numbers of positive and negative examples taken from the files of viral, bacterial, invertebrate, primate, rodent, other mammalian, other vertebrate, and plant in GenBank.

Sequences	Positive	Negative
Viral	120	4882
Bacterial	495	7330
Invertebrate	263	1927
Primate	1032	3162
Rodent	1018	3158
Other Mammalian	235	588
Other Vertebrate	207	1056
Plant	370	3074

For each file, we made an experiment for finding a good decision tree and an indexing. We have reported the results in [39] that achieved good accuracy. The most unsuccessful is the case for the primate signal peptides shown in Fig. 6 (accuracy is $(84.7\%, 79.9\%)$), while the most successful is the case for the bacterial signal peptides whose accuracy is $(86.3\%, 90.6\%)$ for positive and negative examples.

6 Conclusion

We have given an overview of problems for which learning paradigms are required. But the methods so far developed are just on the first stage of research. The next

development is strongly expected for Genome Informatics. In the framework of PAC-learning, we sketched our research on knowledge acquisition from protein data. Based on the research, we have designed and implemented the system BONSAI, which produces a decision tree over regular patterns and an indexing of amino acid residues. We also reported some experiments on transmembrane domains and signal peptides by BONSAI that showed a large potential for discovering knowledge from biological data.

7 Acknowledgments

The author would like to thank S. Kuhara at Graduate School of Genetic Resources Technology, Kyushu University for guiding him into this new field. Special thanks are due to A. Shinohara and S. Shimozono. Without their contributions, the project could not achieve this high level of success. The author is also indebted to many people, especially, S. Arikawa, T. Shinohara and T. Uchida. This research is partly supported by Grant-in-Aid for Scientific Research on Priority Areas "Genome Informatics" from the Ministry of Education, Science and Culture, Japan.

References

1. Arikawa, S., Kuhara, S., Miyano, S., Mukouchi, Y., Shinohara, A., and Shinohara, T. [1993], Machine discovery of a negative motif from amino acid sequences by decision trees over regular patterns, *New Generation Computing* **11**, 361–375.
2. Arikawa, S., Kuhara, S., Miyano, S., Shinohara, A., and Shinohara, T. [1992], A learning algorithm for elementary formal systems and its experiments on identification of transmembrane domains, *Proc. 25th Hawaii International Conference on System Sciences*, 675–684.
3. Asai, K., Hayamizu, S., and Onizuka, K. [1993], HMM with protein structure grammar, *Proc. 26th Hawaii International Conference on System Sciences*, 783–791.
4. Bairoch, A. [1991], PROSITE: a dictionary of sites and patterns in proteins, *Nucleic Acids Res.* **19**, 2241–2245.
5. Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M.K. [1989], Learnability and the Vapnik-Chervonenkis dimension, *JACM*, **36**, 929–965.
6. Brunak, S., Engelbrecht, J., and Knudsen, S. [1990], Neural network detects erros in the assignment of mRNA splice sites, *Nucleic Acids Res.* **18**, 4797–4801.
7. Brunak, S., Engelbrecht, J., and Knudsen, S. [1991], Prediction of human mRNA donor and acceptor sites from the DNA sequence, *J. Mol. Biol.* **220**, 49–65.
8. Bucher, P. [1988], The eukaryote promoter database of the Weizmann Institute of Science, *EMBL Nucleiotite Sequence Data Library Release* **17**, Heidelberg, Germany.
9. Chou, P.Y. and Fasman, G.D. [1978], Prediction of the secondary structure of proteins from their amino acid sequence, *Advances in Enzymology* **47**, 45–147.
10. Cohen, R.E., Abarbanel, R.A., Kuntz, I.D., and Fletterick, R.J. [1986], Turn prediction in proteins using a pattern matching approach, *Biochemistry* **25**, 266–275.
11. Dowe, D.L., Oliver, J., Dix, T.I., Allison, L., and Wallace, C.S. [1993], A decision graph explanation of protein secondary structure prediction, *Proc. 26th Hawaii International Conference on System Sciences*, 669–678.
12. Emini, E.A., Hughes, J.V., Perlow, D.S., and Boger, J. [1985], Induction of hepatitis A virus-neutralizing antibody by a virus-specific peptide, *J. Virol.* **55**, 836–839.

13. Endgelman, D.M., Steiz, T.A., and Goldman, A. [1986], Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins, *Ann. Rev. Biophys. Chem.* **15**, 321–354.
14. Folz, R.J. and Gordon, J.I. [1987], Computer-assisted predictions of signal peptibase processing sites, *Biochem. Biophys. Res. Comm.* **146**, 870–877.
15. Garnier, J., Osguthorpe, D.J., and Robon, B. [1978], Analysis of the accuracy and implication of simple methods for predicting the secondary structure of globular proteins, *J. Mol. Biol.* **120**, 97–120.
16. Gelfand, M.S. [1989], Statistical analysis of mammalian pre-mRNA splicing sites, *Nucleic Acids Res.* **17**, 6369–6382.
17. GenBank, Genetic Sequence Data Bank, National Institute of General Medical Science, NIH by contract to Intelligenetics, Inc., and Los Alamos Laboratory.
18. Gribskov, M. and Devereux, J. [1991], *Sequence Analysis Primer*, UWBC Biotechnical Resource Series, Macmillan Publishers Inc.
19. Haussler, D., Krogh, A., Mian, I.S., and Sjölander, K. [1993], Protein modeling using hidden Markov models: analysis of globins, *Proc. 26th Hawaii International Conference on System Sciences*, 792–802.
20. Harris, N.L. and Senapathy, P. [1990], Distribution and consensus of branch point signals in eukaryotic genes: a computerized statistical analysis, *Nucleic Acids Res.* **18**, 3015–3019.
21. Holley, L.H. and Karplus, M. [1989], Protein secondary structure prediction with a neural network, *Proc. Nal. Acad. Sci. USA* **86**, 152–156.
22. Hopp, T.P. and Woods, K.R. [1981], Prediction of protein antigenic determinants from amino acid sequences, *Proc. Natl. Acad. Sci. USA* **78**, 3824–3828.
23. Iida, Y. and Sasaki, F. [1983], Recognition patterns for exon-intron junctions in higher organism as revealed by a computer search, *J. Biochem.* **94**, 1731–1738.
24. Jameson, B.A. and Wolf, H. [1988], The antigenic index: a novel algorithm for predicting antigenic determinants, *Comput. Appl. Biosci.* **4**, 181–186.
25. Karplus, P.A. and Schulz, G.E. [1985], Prediction of chain flexibility in proteins, *Naturwissenschaften* **72**, 212–213.
26. Kneller, D.G., Choen, F.E., and Langridge, R. [1990], Improvements in protein secondary structure prediction by an enhanced neural network, *J. Mol. Biol.* **214**, 171–182.
27. Kroeger, M., Wahl, R., and Rice, P. [1990], Compilation of DNA sequences of Escherichia coli (update 1990), *Nucleic Acids Res.* **18**, 2549–2587.
28. Kyte, J. and Doolittle, R.F. [1982], A simple method for displaying the hydropathic character of protein, *J. Mol. Biol.*, **157**, 105–132.
29. Ladunga, I., Czako, F., Csabai, I., and Geszti, T. [1991], Improving signal peptide prediction accuracy by simulated neural network, *Comput. Appl. Biosci.* **7**, 485–487.
30. Lewin, B. [1987], *Genes: Third Edition*, John Wiley & Sons, Inc.
31. Miyano, S., Shinohara, A., and Shinohara, T. [1993], Learning elementary formal systems and an application to discovering motifs in proteins, Technical Report RIFIS-TR-CS-37, Research Institute of Fundamental Information Science, Kyushu University, revised in April, 1993 (former version: Proc. 2nd Algorithmic Learning Theory, 139–150, 1991).
32. Nakata, K., Kanehisa, M., DeLisi, C. [1985], Prediction of splice junctions in mRNA sequences, *Nucleic Acids Res.* **13**, 5327–5340.
33. Natarajan, B.K. [1989], On learning sets and functions, *Machine Learning*, **4**, 67–97.
34. Pascarella, S. and Bossa, F. [1989], CLEAVAGE: a microcomputer program for predicting signal sequence cleavage sites, *Comput. Appl. Biosci.* **5**, 53–54.

35. , Qian, N. and Sejnowski, T.J. [1988], Predicting the secondary structure of globular proteins using neural network models, *J. Mol. Biol.* **202**, 865–884.
36. Quinlan, J.R. [1986], Induction of decision trees, *Machine Learning*, **1**, 81–106.
37. Senapathy, P., Shapiro, M.B., and Harris, N.L. [1990], Splice junctions, branch point sites, and exons: sequence statistics, identification, and applications to the genome project, *Meth. Enzym.* **183**, 252–278.
38. Shimozono, S. and Miyano, S. [1992], Complexity of finding alphabet indexing, Technical Report RIFIS-TR-CS-61, Research Institute of Fundamental Information Science, Kyushu University, August, 1992.
39. Shimozono, S., Shinohara, A., Shinohara, T., Miyano, S., Kuhara, S., and Arikawa, S. [1993], Finding alphabet indexing for decision trees over regular patterns: an approach to bioinformatical knowledge acquisition, *Proc. 26th Hawaii International Conference on System Sciences*, 763–772.
40. Shinohara, T. [1983], Polynomial time inference of extended regular pattern languages, *Proc. RIMS Symp. Software Science and Engineering* (Lecture Notes in Computer Science, **147**, 115–127.
41. Staden, R. [1990], An improved sequence handling package that runs on the Apple Macintosh, *Comput. Applic. Biosciences* **6**, 387–393.
42. Staden, R. [1990], Finding protein coding regions in genomic sequences, *Meth. Enzym.* **183**, 163–180.
43. Unger, R. and Moult, J. [1993], On the applicability of genetic algorithms to protein folding, *Proc. 26th Hawaii International Conference on System Sciences*, 715–725.
44. Valiant, L. [1984], A theory of the learnable, *Commun. ACM*, **27**, 1134–1142.
45. von Heijne, G. [1981], On the hydrophobic nature of signal sequences, *Eur. J. Biochem.* **116**, 419–422.
46. von Heijne, G. [1986], A new method for predicting signal sequences cleavage sites, *Nucleic Acids Res.* **14**, 4683–4690.
47. Watson,J.D., Hopkins, N.H., Robets, J.W., Steitz, J.A., and Weiner, A.M. [1987],*Molecular Biology of The Gene: Fourth Edition*, The Benjamin/Cummings Publishing Company, Inc.
48. Yanagihara, N., Suwa, M., and Mitaku, S. [1989], A theoretical method for distinguishing between soluble and membrane proteins, *Biophysical Chemistry*, **34**, No. 1, 69–77.