BONSAI Garden: Parallel Knowledge Discovery System for Amino Acid Sequences

T. Shoudai
Department of Physics, Kyushu University 01,
Ropponmatsu, 810 Fukuoka, Japan
shoudai@rc.kyushu-u.ac.jp

M. Lappe
FB Mathematik-Informatik,
Universität-GH-Paderborn,
D-33095 Paderborn, Germany
lst@uni-paderborn.de

S. Miyano, A. Shinohara, T. Okazaki, S. Arikawa
Rsearch Institute of Fundamental Informaetion Science,
Kyushu University 33, 812 Fukuoka, Japan
{miyano,ayumi,okazaki,arikawa}@rifis.kyushu-u.ac.jp

T. Uchida
Department of Information Science,
Hiroshima City University,
731-31 Hiroshima, Japan
uchida@cs.hiroshima-cu.ac.jp

S. Shimozono, T. Shinohara
Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology,
820 Iizuka, Japan
{sin@ces,shino@ai}.kyutech.ac.jp

S. Kuhara
Graduate School of Genetic Resources Technology,
Kyushu University, 812-81 Fukuoka, Japan
kuhara@grt.kyushu-u.ac.jp

Abstract

We have developed a machine discovery system BONSAI which receives positive and negative examples as inputs and produces as a hypothesis a pair of a decision tree over regular patterns and an alphabet indexing. This system has succeeded in discovering reasonable knowledge on transmembrane domain sequences and signal peptide sequences by computer experiments. However, when several kinds of sequences are mixed in the data, it does not seem reasonable for a single BONSAI system to find a hypothesis of a reasonably small size with high accuracy. For this purpose, we have designed a system BONSAI Garden, in which several BONSAI's and a program called Gardener run over a network in parallel, to partition the data into some number of classes together with hypotheses explaining these classes accurately.

Keywords: machine learning, parallel knowledge acquisition, scientific discovery, decision tree, regular pattern, alphabet indexing, signal peptide, membrane protein

Mailing Address: S. Miyano, Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan.
Email: miyano@rifis.kyushu-u.ac.jp
fax: +81-92-611-2668

Introduction

For knowledge discovery from amino acid sequences of proteins, we have studied a learning model and related algorithmic techniques and have designed a machine discovery system BONSAI (Arikawa et al. 1993; 1992; Miyano, Shinohara, & Shinohara 1991; 1993; Shimozono 1995; Shimozono & Miyano 1995; Shimozono et al. 1994). When positive and negative examples of sequences are given as input data, BONSAI (Shimozono et al. 1994) produces a pair of a decision tree over regular patterns and an alphabet indexing (Shimozono & Miyano 1995; Shimozono et al. 1994) as a hypothesis which shall represents knowledge about the data. The name of "BONSAI" comes from the fact that the knowledge (the nature) is expressed as a small tree (a decision tree over regular patterns) in harmony with an alphabet indexing (a pot). This system has succeeded in discovering reasonable knowledge on transmembrane domain sequences and signal peptide sequences (Arikawa et al. 1993; Shimozono et al. 1994). Through these experimental results together with theoretical foundations, we have

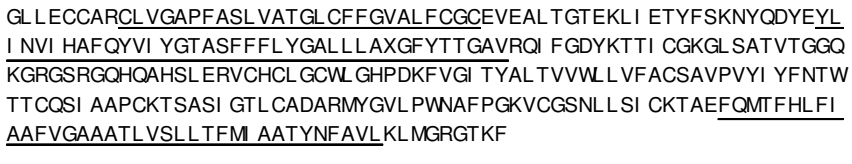

Figure 1: Myelin proteolipid protein - Human. The underlined sequences are transmembrane domains.

Positive Examples	Negative Examples
CLVGAPFASLVATGLCFFGVALFCGC	FGDYKTTICGKGLSATVTGGQKGRGSRG
YLINVIHAFQYVIYGTASFFFLYGALLLAXGFYTTGAV	PDKFVGITYALTVVWLLVFACSAVPVYIY
FQMTFHLFIAAFVGAAATLVSLLTFMIAATYNFAVL	TTCQSIAAPCKTSASIGTLCADARMYGVLPW
...	...
IALAFLATGGVLLFLAT	VFLENVIRDAVTYTEHAKRKTVTAMDVV
LDTYRIVLLLIGICSLL	NAKQDSRGKIDAARISVDTDKVSEA
EVLTAVGLMFAIVGGLA	IFTKPKAKSADVESDVDVLDTGIYS
PGYALVALAIGWMLGS	GRMVLTAEGRSVHDSSSDCYQYFCVPEY
...	...

Figure 2: Transmembrane domain sequences and non-transmembrane domain sequences

recognized that the potential ability of BONSAI is very high.

On the other hand, when several kinds of sequences are mixed in the data, i.e., a hodgepodge of sequences, it is desirable that the data should be classified into several classes and each class should be explained with a simple hypothesis with high accuracy. BONSAI Garden is designed for this purpose. BONSAI Garden consists of some number of BONSAI's and a coordinator called a **Gardener**. The **Gardener** and BONSAI's run over a network in parallel for classification of data and knowledge discovery. This paper presents the design concept developed for BONSAI Garden together with the background theory and ideas in BONSAI. Although no mathematical theorems are provided for BONSAI Garden, experimental results show an interesting nature of BONSAI Garden and we believe that it would be one of the prototypes of intelligent systems for molecular biology.

In the following section, we give our framework of machine discovery by PAC-learning paradigm (Blumer **et al.** 1989; Valiant 1984) and related concepts with which BONSAI is developed. Then, we discuss the system BONSAI from the viewpoint of practice and sketches the system briefly. The idea of BONSAI Garden is given in detail and we report some experimental results on BONSAI Garden.

Machine Discovery by Learning Paradigm

Framework

The sequence in Fig. 1 is an amino acid sequence of a membrane protein where three transmembrane domains are underlined on the sequence. When we are given a collection of such sequences, the task of knowledge discovery is to find "explanations" or "concepts" about, in this case, transmembrane domains.

In order to acquire knowledge about such sequences, it may be the first step to collect the sequences for transmembrane domains (**positive examples**) and to compare them with the sequences other than transmembrane domains (**negative examples**) as shown in Fig. 2.

Based on the principle of Occam's razor, our idea of knowledge discovery is to find a "short" explanation or concept which distinguishes positive examples from negative examples. To cope with such a situation, a general framework of machine discovery by learning paradigm which consists of the following items is proposed in (Miyano 1993): (a) View design for the data. (b) Concept design with the views. (c) Learning algorithm design for the concept class. (d) Experiments by the learning algorithms for discovering knowledge.

A **view** is a collection of "words" with which we make a sentence of "explanation" about the given data. A **hypothesis space** consists of sentences for "explanations" and the hypothesis space design is to give a formal definition of expressions for the sentences. In the following section, we shall discuss the items (a)–(d) in detail by following our work (Arikawa **et al.** 1993; 1992; Shimozono 1995; Shimozono & Miyano 1995; Shimozono **et al.** 1994).

View and Concept Designs in BONSAI

As data, we deal with amino acid sequences of proteins. In BONSAI (Shimozono **et al.** 1994), two kinds of views on sequences are employed. An amino acid sequence of a protein is a string over the alphabet of twenty symbols representing the amino acid residues. The following views assume sequences of this kind as data.

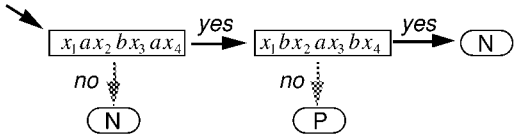

Figure 3: Decision tree over regular patterns with $L(T) = \{a^m b^n a^l \mid m, n, l \geq 1\}$.

A regular pattern over an alphabet Γ (Angluin 1980; Shinohara 1983) is an expression of the form $\pi = \alpha_0 x_1 \alpha_1 x_2 \cdots x_n \alpha_n$, where $\alpha_0, ..., \alpha_n$ are strings over the alphabet Γ and $x_1, ..., x_n$ are mutually distinct variables to which arbitrary strings in Γ^* (or Γ^+) are substituted. It defines a regular language which is denoted by $L(\pi)$. A regular pattern containing at most k variables is called a k-variable regular pattern. A string w is classified according to its membership $w \in L(\pi)$ or $w \notin L(\pi)$. These regular patterns constitute the first view on sequences.

The second view is one on the alphabet itself. Shimozono and Miyano (Shimozono & Miyano 1995) defined a notion of an alphabet indexing in the following way: Let Σ be a finite alphabet and P and N be two disjoint subsets of Σ^* whose strings are of the same length. Let Γ be a finite alphabet with $|\Sigma| > |\Gamma|$, called an indexing alphabet. An alphabet indexing ψ of Σ by Γ with respect to P and N is a mapping $\psi : \Sigma \to \Gamma$ such that $\tilde{\psi}(P) \cap \tilde{\psi}(Q) = \emptyset$, where $\tilde{\psi} : \Sigma^* \to \Gamma^*$ is the homomorphism defined by $\tilde{\psi}(a_1 \cdots a_n) = \psi(a_1) \cdots \psi(a_n)$ for $a_1, \ldots, a_n \in \Sigma$.

With these two views on sequences, we defined in (Arikawa et al. 1993; Shimozono et al. 1994) a concept class by introducing decision tree over regular patterns. A decision tree over regular patterns (Arikawa et al. 1993; Shimozono et al. 1994) is a binary decision 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 for classification (see Fig. 3). For a pair (T, ψ) of a decision tree T over regular patterns over Γ and an alphabet indexing $\psi : \Sigma \to \Gamma$, we define $L(T, \psi) = \{x \in \Sigma^* \mid \tilde{\psi}(x)$ is classified as P by $T\}$. Obviously, $L(T, \psi)$ is a regular language over Σ. The pairs (T, ψ) are used as the representation of concepts. Thus the hypothesis space consists of such pairs (T, ψ) . For finite sets $POS, NEG \subseteq \Sigma^*$, the accuracy of a hypothesis (T, ψ) for POS and NEG is defined by

$$Score(T, \psi) = \frac{|L(T,\psi) \cap \tilde{\psi}(POS)|}{|\tilde{\psi}(POS)|} \cdot \frac{|\overline{L(T,\psi)} \cap \tilde{\psi}(NEG)|}{|\tilde{\psi}(NEG)|}.$$

Background Theory and Complexity

This section presents a framework of knowledge discovery by PAC-learning paradigm (Valiant 1984) developed in our work (Arikawa et al. 1993; 1992; Shimozono et al. 1994) and discusses some related complexity issues.

We review some notions from concept learning. A subset of Σ^* is called a concept and a concept class $\mathcal{C}$ is a nonempty collection of concepts. For a concept $c \in \mathcal{C}$, an element $w \in \Sigma^*$ is called a positive example (negative example) of c if w is in c (is not in c). 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 mapping $R : \mathcal{C} \to 2^{\Lambda^*}$ 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.

We do not give any formal definition of PAC-learnability (see (Blumer et al. 1989; Natarajan 1989; Valiant 1984) for definition). Instead, we mention a very useful theorem for practical applications. A concept class is known to be polynomial dimension if there is a polynomial $d(n)$ such that $\log |\{c \cap \Sigma^n \mid c \in \mathcal{C}\}| \leq d(n)$ for all $n \geq 0$. This is a notion independent of the representation of the concept class. A polynomial-time fitting for $\mathcal{C}$ is a deterministic polynomial-time algorithm that takes a finite set S of examples as input and outputs a representation of a concept c in $\mathcal{C}$ which is consistent with S, if any. Thus this depends on the representation of the concept class. The following result is a key to the design of a polynomial-time PAC-learning algorithm for a concept class, where the size parameter s (Natarajan 1989) for the minimum size representation of a concept is not considered. In Theorem 1, the polynomial-time fitting is the required learning algorithm for the concept class.

Theorem 1. (Blumer et al. 1989; Natarajan 1989) A concept class $\mathcal{C}$ is polynomial-time learnable if $\mathcal{C}$ is of polynomial dimension and there is a polynomial-time fitting for $\mathcal{C}$.

For knowledge discovery from amino acid sequences, we introduced in (Arikawa et al. 1993) a class $DTRP(d, k)$ of sets defined by decision trees over k-variable regular patterns with depth at most d ($k, d \geq 0$).

Theorem 2. (Arikawa et al. 1993) $DTRP(d, k)$ is polynomial-time learnable for all $d, k \geq 0$.

The above theorem is easily shown by proving the conditions of Theorem 1. But the result is especially important in practice of machine discovery because it gives us a guarantee for discovery when the target concept can be captured as a decision tree over regular patterns. In (Arikawa et al. 1993; Shimozono et al. 1994), we have shown the usefulness of the class $DTRP(d, k)$ by experiments.

We relate some complexity issues. We want to find a small decision tree but it is known that the problem of finding a minimum size decision tree is NP-complete (Hyafil & Rivest 1976). Moreover, we should mention

that the polynomial-time fitting in Theorem 2 does not have any sense in practice. We have also shown that the problem of finding a regular pattern which is consistent with given positive and negative examples is NP-complete (Miyano, Shinohara, & Shinohara 1991; 1993). As to the alphabet indexing problem, we have also shown in (Shimozono & Miyano 1995) that the problem is NP-complete. These computational difficulties are solved practically in the design of BONSAI.

BONSAI System

Overview of BONSAI

BONSAI system (Fig. 4) is designed based on the notions and results in machine learning paradigms. BONSAI assumes two sets POS and NEG of positive examples and negative examples. In order to discover knowledge, BONSAI will take training examples from POS and NEG randomly. The sets P and N consist of positive and negative training examples, respectively. The window size of positive (negative) training examples is the cardinality $|P|$ ($|N|$). From these sets P and N, BONSAI shall find a hypothesis (T, ψ) that may explain the unknown concept provided as POS and NEG with high accuracy. In the design of BONSAI, we had to solve the difficulties mentioned in the previous section in a practical way. Three problems arise for this purpose. The first is the problem of constructing efficiently small decision trees over regular patterns. The second is the problem of finding good alphabet indexings. The third is how to combine the process for decision trees with the alphabet indexing process. A sketch is given in Fig. 4 (see (Shimozono et al. 1994) for more detail).

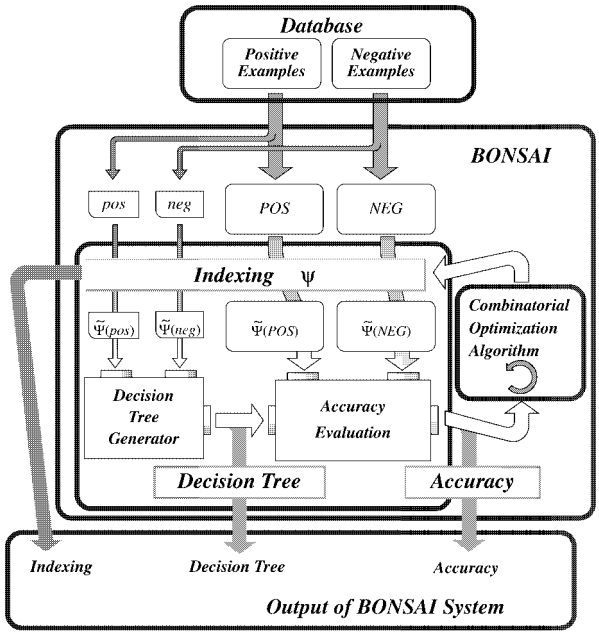

Figure 4: BONSAI

View Design and Decision Tree Algorithm

We employed the idea of ID3 (Quinlan 1986) for constructing a decision tree because, empirically, ID3 produces small enough decision trees very efficiently. ID3 assumes a set Π of attributes and a set D of data specified by the values of the attributes in advance.

However, we are just given only sequences called positive and negative examples and no attributes are provided explicitly. For utilizing the ID3 algorithm for knowledge discovery, we employ the view of regular patterns, each of which is regarded as an attribute that takes values in $\{\mathrm{P}, \mathrm{N}\}$. Then by specifying a class Π of regular patterns as attributes, we can apply the idea of ID3 to constructing a decision tree as in Fig. 5. Thus the choice of Π and the method for finding π in Π minimizing $E(\pi, P, N)$ in Fig. 5 are important for knowledge discovery. From a practical point, the following strategies are considered and some of them are implemented in BONSAI.

1. Π consists of all regular patterns of the form $x_0 \alpha x_1$, where α is a substring of a string in $P \cup N$. The process for finding π in Π minimizing $E(\pi, P, N)$ is an exhaustive search in Π. BONSAI System in the former version implemented this method. Very successful results are reported in (Arikawa et al. 1993; Shimozono et al. 1994).
2. In order to deal with regular patterns having more variables, we developed a heuristic algorithm by using a polynomial-time algorithm for finding a longest common substring of two strings. Our experimental results show that this heuristic method is very efficient and powerful. The new version of BONSAI also involves this strategy for finding a regular pattern for a node of a decision tree. The details of the algorithm are omitted.
3. We also made some experiments by using a genetic algorithm for finding regulars pattern with more variables. Although good regular patterns were found, the amount of time for computing was not acceptable in the experiments and this strategy is not accepted in BONSAI.

Heuristics for Alphabet Indexing

An alphabet indexing ψ with respect to P and N must satisfy $\tilde{\psi}(P) \cap \tilde{\psi}(N) = \emptyset$ and the problem of finding such an alphabet indexing is NP-complete (Shimozono & Miyano 1995). Therefore, in practice, we relax the condition by allowing overlaps for $\tilde{\psi}(P)$ and $\tilde{\psi}(N)$. For finding a good alphabet indexing, three methods have been developed and tested for BONSAI:

1. Local search method: Let $\Psi = \{\psi \mid \psi : \Sigma \to \Gamma\}$. For ψ and ϕ in Ψ, we define the distance by $d(\psi, \phi) = |\{a \mid \psi(a) \neq \phi(a)\}|$. Then the neighborhood of ψ is

```
function MakeTree( P, N : sets of strings ): node;
 begin
  if N = ∅ then
   return( Create("1", null, null) )
  else if P = ∅ then
   return( Create("0", null, null) )
  else begin
   Find a regular pattern π in Π
   minimizing 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 return( ( Create("1", null, null) )
   else
    return
    Create(π, MakeTree(P0, N0), MakeTree(P1, N1))
   end
 end
```

(a) 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.

(b) $E(\pi, P, N) = \frac{p_1 + n_1}{|P| + |N|} I(p_1, n_1) + \frac{p_0 + n_0}{|P| + |N|} I(p_0, n_0)$, where $p_1 = |P \cap L(\pi)|$, $n_1 = |N \cap L(\pi)|$, $p_0 = |P \cap \overline{L(\pi)}|$, $n_0 = |N \cap \overline{L(\pi)}|$, $\overline{L(\pi)} = \Sigma^* - L(\pi)$, and $I(x, y) = -\frac{x}{x+y} \log \frac{x}{x+y} - \frac{y}{x+y} \log \frac{y}{x+y}$ (if $xy \neq 0$), $I(x, y) = 0$ (if $xy = 0$).

Figure 5: Algorithm for constructing a decision tree over regular patterns

the set $N(\psi) = \{\phi \mid d(\psi, \phi) = 1\}$. For a decision tree T, $Score(T, \phi)$ is used as the cost function. A simple local search strategy is implemented in BONSAI and the experiment in (Shimozono et al. 1994) shows that this local search strategy found good alphabet indexings.

2. Approximation algorithm: A polynomial-time approximation algorithm is also developed in Shimozono (Shimozono 1995) for which an explicit error ratio is proved. This algorithm has not yet been fully tested for its usefulness.

3. Cluster analysis: In (Nakakuni, Okazaki, & Miyano 1994), a method for finding an alphabet indexing by using a cluster analysis called Ward's method has been developed and tested. The experimental results are acceptable but this method is not yet installed in the current version of BONSAI.

BONSAI Garden: Knowledge Discovery from Hodgepodge

When we have a collection of sequences for BONSAI which may contain noises or is a hodgepodge of various kinds of sequences, it is not reasonable to explain the data by a single hypothesis produced by BONSAI. Coping with such situation, we have designed a system BONSAI Garden that runs several BONSAI's in parallel. The target of BONSAI Garden shall be the following:

- The system should be able to handle a hodgepodge of data and/or noisy data so that it classifies the data into some number of classes of sequences and simultaneously finds for each of these classes a hypothesis which explains the sequences in the class.

This section presents its idea and implementation of BONSAI Garden.

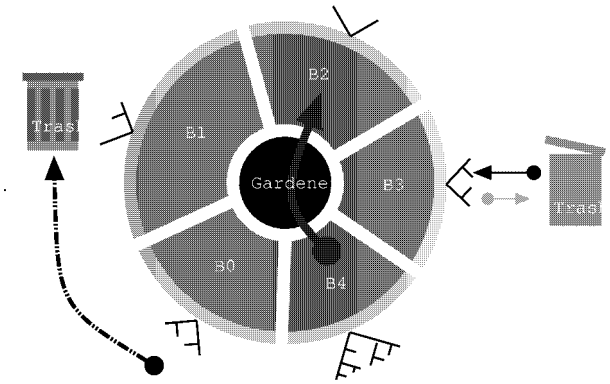

Figure 6: BONSAI Garden

BONSAI Garden consists of BONSAI's and a program Gardener (Fig. 6), which is not introduced in our former work (Shinohara et al. 1993). Let B_i be a BONSAI system with POS_i and NEG_i as the sets of positive and negative examples for $i = 0, ..., m-1$. BONSAI B_i with (POS_i, NEG_i) produces a hypothesis (T_i, ψ_i) and puts the sequences in POS_i and NEG_i misclassified by (T_i, ψ_i) into the sets $POS.TRASH_i$ and $NEG.TRASH_i$, respectively. POS_i and NEG_i are updated with the positive and negative examples correctly classified by (T_i, ψ_i). This is a single job cycle of BONSAI. Each BONSAI repeats this process under the control of Gardener.

Gardener watches the behaviors of BONSAI's $B_0, \ldots, B_{m-1}$ and executes the following task (the parts marked with $*$ below provides the case that negative examples will be classified):

1. Watch: Gardener will find two BONSAI's, say B_i and B_j, which have finished one job cycle with

$$[(T_i, \psi_i), POS_i, NEG_i, POS.TRASH_i, NEG.TRASH_i],$$
$$[(T_j, \psi_j), POS_j, NEG_j, POS.TRASH_j, NEG.TRASH_j].$$

2. Compare: Gardener will compare the the sizes of hypotheses (T_i, ψ_i) and (T_j, ψ_j) and determine which is smaller. For this purpose, we must specify the size of a hypothesis. In BONSAI Garden, the number of symbols in the expression of a decision tree is used for the size of a hypothesis. Suppose that the hypothesis (T_i, ψ_i) produced by B_i is smaller than (T_j, ψ_j) by B_j.

3. Classify: Gardener will classify the sequences in POS_j by the smaller hypothesis (T_i, ψ_i). Let $POS.NEW_i$ be the set of examples in POS_j which are classified as positive by (T_i, ψ_i) and $POS.NEW_j$ be the set of examples in POS_j which are classified as negative by (T_i, ψ_i).
   ($*$: Let $NEG.NEW_i$ be the set of examples in NEG_j which are classified as negative by (T_i, ψ_i) and $NEG.NEW_j$ be the set of examples in NEG_j which are classified as positive by (T_i, ψ_i).)
4. Merge: Gardener will update POS_i and POS_j as follows:
   (a) $POS_i \leftarrow POS_i \cup POS.NEW_i$
   ($*$: $NEG_i \leftarrow NEG_i \cup NEG.NEW_i$)
   (b) $POS_j \leftarrow POS.NEW_j$
   ($*$: $NEG_j \leftarrow NEG.NEW_j$)

   Thus BONSAI with a smaller hypothesis will get more sequences while BONSAI with larger hypothesis will loose sequences.

After the above task by Gardener, BONSAI B_i updates his POS_i (NEG_i) as follows:

1. Distribute Trash: BONSAI B_i fetches all examples from the neighbor trashes $POS.TRASH_{i-1}$ ($NEG.TRASH_{i-1}$) and merge them into its POS_i (NEG_i). For $i = 0$, we assume that its neighborhood is BONSAI B_{m-1}.

   Then, B_i will start its next job cycle.

When all trashes $POS.TRASH_i$ ($NEG.TRASH_i$) become empty, BONSAI Garden halts. However, there is no guarantee of such termination. Thus we need to kill the process after some amount of execution. The Gardener with the above task is just a prototype designed for our current interest. By specifying the task of Gardener, we can design various BONSAI Gardens. The above idea is implemented on a network of workstations.

Experiments on BONSAI Garden

In (Shimozono et al. 1994), we collected signal peptide sequences from GenBank. A signal peptide is located at N-terminal region, that is at the initial segment of an amino acid sequence. As positive examples, the signal peptide sequences beginning with a Methionine (M) and of length at most 32 are collected. For the negative examples, we take N-terminal regions of length 30 obtained from complete sequences that have no signal peptide and begin with a Methionine. Table 1 shows the numbers of positive and negative examples in the families. By merging these files in Table 1, we made a hodgepodge of sequences. Let $SIGPOS$ and $SIGNEG$ be the sets of all positive and negative examples.

For experiment, we run BONSAI Garden with nine BONSAI's for these $SIGPOS$ and $SIGNEG$ by employing Gardener which does not exchange negative examples. The window size is set 4 and the alphabet indexing size is set 3. The negative examples in $SIGNEG$ are distributed to nine BONSAI's evenly. The experimental result shows that the signal peptide sequences are classified into one large class of 2205 sequences and two classes of 640 and 603 sequences as given in Table 2 together with small classes. Although we expected decision trees with two or three internal nodes, every result was a tree of a single internal node labeled with a regular pattern having four to six variables.

Sequences	Positive	(%)	Negative
Primate	1032	(27.2%)	3162
Rodent	1018	(26.8%)	3158
Bacterial	495	(13.0%)	7330
Plant	370	(9.8%)	3074
Invertebrate	263	(6.9%)	1927
Other Mammalian	235	(6.2%)	588
Other Vertebrate	207	(5.5%)	1056
Viral	120	(3.2%)	4882
Others	56	(1.4%)	2775
TOTAL	3796	(100.00%)	27952

Table 1: Numbers of positive and negative examples in files

We also made an experiment by using transmembrane data (Arikawa et al. 1992; 1993; Shimozono et al. 1994) MEMPOS (689 sequences) and MEMNEG (19256 sequences) on a single BONSAI with the new regular pattern searching algorithm. It found an alphabet indexing and a regular pattern (Table 3 (a)) which achieves the same accuracy as that found in (Arikawa et al. 1993; Shimozono et al. 1994). Thus a single hypothesis can explain (MEMPOS,MEMNEG) with very high accuracy (95%). However, by a similar experiment by using $SIGPOS$ and $SIGNEG$, it does not seem reasonable to explain the signal peptide sequences by a single hypothesis (Table 3 (b)).

Conclusion

We have presented the design concept of BONSAI Garden for knowledge discovery from hodgepodge of sequences. The system is designed for classifying a hodgepodge of sequences into some number of classes together with hypotheses explaining these classes. Some experiments on BONSAI Garden with signal peptide sequences proved the potential ability of BONSAI Garden. In the future, it may be possible to report more experimental results on protein data with further discussions on learning aspects.

Acknowledgment

This work is supported in part by Grant-in-Aid for Scientific Research on Priority Areas "Genome Informatics" from the Ministry of Education, Science and Culture, Japan.

BONSAI	Size	Alphabet Indexing ACDEFGHI KLMNPQRSTVWXY	Decision tree (regular pattern)
B_2	2205	012010221000200112112	*2*02*1*02*(P, N)
B_6	640	112200101122021020112	*10*0*21*0*(P, N)
B_8	603	021022212212221102022	*222*12*12*(P, N)
B_0	119	112102101021110222202	2*21*12*1*2*(P, N)
B_1	76	120111112202222010220	*1*12*20*12*(P, N)
B_4	69	111020102222122221222	*0*02*2*1*22*(P, N)
B_7	58	022212121002011012021	*20*2*12*1*1*(P, N)
B_3	22	220021220121122221200	*22*2*1*0*1*1*(P, N)
B_5	4	102102101120200212222	*122*0001*21*(N, P)

Table 2: For example, *2*02*1*02* (P,N) represents a decision tree whose root is labeled with *2*02*1*02* and the left (no) and right (yes) leaves are P and N, respectively. 3,796 sequences are classified into three main classes. The accuracy is 100% for positive examples in each class. Only three negative examples are misclassified from 27,952 sequences. The percentages of primate, rodent, and bacterial sequences in B_2 are 28.4%, 25.4%, and 13.0%, respectively, which are almost the same as those in Table 1. The same is observed for B_6 and B_8.

	Alphabet Indexing ACDEFGHI KLMNPQRSTVWXY	Decision tree (regular pattern)	Score
(a)	100011010110100111101	*0*0*01*0*0*(P,N)	$94.6\% = \sqrt{93.9 \times 95.3}$
(b)	120010000211120010000	*10*222*2*10*0* $\xrightarrow{\text{yes}}$ P $\downarrow$no *10*0*0*0*00*0*0*0*(P, N)	$80.9\% = \sqrt{85.4 \times 76.6}$

Table 3: Results on transmembrane domain sequences and signal peptide sequences by a single BONSAI.

BONSAI	Size	Alphabet Indexing ACDEFGHI KLMNPQRSTVWXY	Decision tree (regular pattern)
B_3	2175	200021110212201212221	*10*(P, *20*(P, N))
B_0	429	110210202120000211120	*200*(*1000*(P, N), N)
B_4	190	120011220210122212200	*10*(P, *220*(*00*(P, N), N)
B_1	37	120210222102020212020	*202*(P, *02022*(N, *111*(N, P)))
B_2	30	110210222122020212200	*111*(N, *022*(P, *20*(P, N)))
B_6	26	120112021200001111200	*2010*(*110*(P, *00*(*22*(N, P), N)), P)
B_5	17	020022021001121210020	*102*(*00202*(*00021*(P, *211*(N, P)), N), N)

Table 4: Results on a hodgepodge of various signal peptide sequences. The number of BONSAI's which were simultaneously running is 7. Each regular expressions on internal nodes of decision trees are of the type $x\alpha y$. The numbers of classified positive examples in this result are given in Table 5.

Sequences	BONSAI								
	B_0	B_1	B_2	B_3	B_4	B_5	B_6	Trash	
Bacterial	56	7	4	330	16	4	4	1	422
Invertebrate	22	3	1	167	18	0	1	0	212
Other Mammalian	33	4	2	145	14	1	3	1	203
Organelle	0	0	0	5	0	0	0	1	6
Phage	0	0	0	11	0	0	0	0	11
Plant	52	4	5	237	17	3	3	3	324
Primate	101	6	6	523	42	2	10	0	690
Rodent	122	4	6	549	59	3	1	1	745
Synthetic	2	0	0	21	3	0	0	0	26
Viral	16	7	1	63	5	3	1	1	97
Other Vertebrate	25	2	5	124	16	1	3	1	177
Size	429	37	30	2175	190	17	26	9	2913

Table 5: Numbers of classified sequences by 7 BONSAI's

References

Angluin, D. 1980. Finding patterns common to a set of strings. J. Comput. System Sci. 21:46–62.

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. In Proc. 25th Hawaii International Conference on System Sciences, 675–684.

Arikawa, S.; Kuhara, S.; Miyano, S.; Mukouchi, Y.; Shinohara, A.; and Shinohara, T. 1993. A machine discovery of a negative motif from amino acid sequences by decision trees over regular patterns. New Generation Computing 11:361–375.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. K. 1989. Learnability and the Vapnik-Chervonenkis dimension. JACM 36:929–965.

Hyafil, L., and Rivest, R. 1976. Constructing optimal binary decision trees is NP-complete. Inf. Process. Letter 5:15–17.

Miyano, S.; Shinohara, A.; and Shinohara, T. 1991. Which classes of elementary formal systems are polynomial-time learnable? In Proc. 2nd Algorithmic Learning Theory, 139–150.

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.

Miyano, S. 1993. Learning theory towards Genome Informatics. In Proc. 4th Workshop on Algorithmic Learning Theory (Lecture Notes in Artificial Intelligence 744), 19–36.

Nakakuni, H.; Okazaki, T.; and Miyano, S. 1994. Alphabet indexing by cluster analysis: a method for knowledge acquisition from amino acid sequences. In Proc. Genome Informatics Workshop V, 176–177.

Natarajan, B. 1989. On learning sets and functions. Machine Learning 4:67–97.

Quinlan, J. 1986. Induction of decision trees. Machine Learning 1:81–106.

Shimozono, S., and Miyano, S. 1995. Complexity of finding alphabet indexing. IEICE Trans. Information and Systems E78-D:13–18.

Shimozono, S.; Shinohara, A.; Shinohara, T.; Miyano, S.; Kuhara, S.; and Arikawa, S. 1994. Knowledge acquisition from amino acid sequences by machine learning system BONSAI. Trans. Information Processing Society of Japan 35:2009–2018.

Shimozono, S. 1995. An approximation algorithm for alphabet indexing problem. Technical Report RIFIS-TR-CS-96, Research Institute of Fundamental Information Science, Kyushu University.

Shinohara, A.; Shimozono, S.; Uchida, T.; Miyano, S.; Kuhara, S.; and Arikawa, S. 1993. Running learning systems in parallel for machine discovery from sequences. In Proc. Genome Informatics Workshop IV, 74–83.

Shinohara, T. 1983. Polynomial time inference of extended regular pattern languages. In Proc. RIMS Symp. Software Science and Engineering (Lecture Notes in Computer Science 147), 115–127.

Valiant, L. 1984. A theory of the learnable. Commun. ACM 27:1134–1142.