A Greedy Strategy for Finding Motifs from Yes-No Examples

Erika Tateishi*and Satoru Miyano†

*Department of Information Systems, Kyushu University 39, Kasuga 816, Japan

†Research Institute of Fundamental Information Science

Kyushu University 33, Fukuoka 812, Japan

Abstract

We define a motif as an expression $Z_1 \cdot Z_2 \cdots Z_n$ with sets $Z_1, Z_2, \ldots, Z_n$ of strings in a specified family Ω called the type. This notion can capture the most of the motifs in PROSITE as well as regular pattern languages. A greedy strategy is developed for finding such motifs with ambiguity just from positive and negative examples by exploiting the probabilistic argument. This paper concentrates on describing the idea of the greedy algorithm with its underling theory. Its experimental results on splicing sites and *E. coli* promoters are also presented.

1. Introduction

Technologies for discovering knowledge from nucleic acid and amino acid sequences are most expected in Genome Informatics/Molecular Bioinformatics. Various alignment techniques [8] have traditionally played a very important role in knowledge discovery from sequences. The knowledge on sequences is often expressed as a *motif* which is a pattern common to a family of sequences. PROSITE Database [3] collects such "motifs" of amino acid sequences of proteins which are expressed in a systematic way. For example, [AC]-$x(1)$-V-$x(4)$-{ED} is a motif representing [A or C]-any-V-any-any-any-any-{any but E or D}. In a motif C-$x(2,4)$-C-$x(12)$-H-$x(3,5)$-H, $x(2,4)$, $x(12)$, and $x(3,5)$ represent any sequence of length between 2 and 4, any sequence of length exactly 12, and any sequence of length between 3 and 5, respectively. Thus some kind of ambiguity is allowed in motifs since diversity and uncertainty are involved by nature.

*Corresponding author: Erika Tateishi, Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan. Email: tateishi@rifis.kyushu-u.ac.jp.

Finding such motifs from nucleic acid and amino acid sequences is a crucial problem since motifs provide biologically important knowledge expressed as sequences. The most powerful techniques are the finely tuned sequence alignment algorithms which assume in advance some knowledge such as the Dayhoff matrix. Recently, as to the practice of motif discovery, Wu and Brutlag [22] have taken an interesting approach and shown a very successful result on the subclass of retroviral and retrovirus-related reverse transcriptases by their heuristic search algorithm although no mathematical proof is supplied to the algorithm for showing its performance.

This paper presents a greedy strategy for finding such motifs with ambiguity just from positive and negative examples. The idea is based on the probabilistic argument invented for designing approximation algorithms for the maximum satisfiability problem [11, 23]. For motifs of a special type, Tateishi et al. [20] proved a lower bound of the performance of the algorithm. Thus some performance guarantee is provided for our method. We describe the details of algorithm together with its underling theory, and present its experimental results on splicing sites and *E. coli* promoters. We also provide a variation of the algorithm in order to handle a more general case though no mathematical results are yet shown for supporting its performance.

We define a motif as an expression $Z_1 \cdot Z_2 \cdots Z_n$, where $Z_1, Z_2, \ldots, Z_n$ are sets of strings in a specified family Ω called the type. When Z_i consists of several elements, such as [AC], the expression allows ambiguity. This notion completely captures the above cases and the case of regular patterns [1] in a uniform way.

We should note that the motif discovery involves computationally difficult obstacles. As well known, The longest common subsequence problem is NP-complete [12]. The complexity issues on the problem of finding a best consensus motif from positive and negative examples have been thoroughly investigated by Tateishi et al. [20]. It is shown in [20] that even a problem for a very simple type is NP-complete whether ambiguity is allowed in a motif or not since its proof works for both cases. Similar works related to the complexity issues on pattern languages are also found in Jiang and Li [10] and Miyano et al. [13].

2. Motifs and Complexity

For an alphabet Σ, we denote by Σ^* the set of all strings over Σ. The length of a string w in Σ^* is denoted by $|w|$. We denote $\Sigma^+ = \Sigma^* - \{\varepsilon\}$ (ε is the empty string) and $\Sigma^n = \{w \in \Sigma^* \mid |w| = n\}$ for an integer $n \geq 0$. For a set S, the number of elements in S is also denoted by $|S|$.

Definition 1 Let Ω be a family of subsets of Σ^* called a *type*. A *motif* π of type Ω is an expression of the form

$$Z_1 \cdots Z_n,$$

where $Z_1, \ldots, Z_n$ are elements in Ω. For a motif $\pi = Z_1 \cdots Z_n$, we denote by $L(\pi)$ the set of strings defined by $\{w_1 \cdots w_n \mid w_1 \in Z_1, \ldots, w_n \in Z_n\}$. For a string w and a motif π, we say that π *accepts* (*rejects*) w if $w \in L(\pi)$ ($w \notin L(\pi)$).

Example 1 A regular pattern [1] is an expression of the form $\pi = w_0 x_1 w_1 \cdots w_{n-1} x_n w_n$ consisting of strings $w_0, \ldots, w_n \in \Sigma^*$ and distinct variables $x_1, \ldots, x_n$. The pattern π defines a set $L(\pi)$ of strings in Σ^* obtained by substituting any strings in Σ^+ to the variables $x_1, \ldots, x_n$. Then any regular pattern is regarded as a motif of type $\Omega = \{\Sigma^+\} \cup \{\{w\} \mid w \in \Sigma^*\}$.

Example 2 Let Σ be the set of amino acid residues. For integers k, $i < j$, let $X(k) = \{w \mid w \in \Sigma^*,\ |w| = k\}$ and $X(i,j) = \{w \mid w \in \Sigma^*,\ i \leq |w| \leq j\}$. Let $\Omega = \{X(k) \mid k \geq 1\} \cup \{X(i,j) \mid 1 \leq i < j\} \cup \{Z \mid \emptyset \neq Z \subseteq \Sigma\}$. Then the zinc finger motif C-$x(2,4)$-C-$x(12)$-H-$x(3,5)$-H, the leucine zipper L-$x(6)$-L-$x(6)$-L-$x(6)$-L and a motif such as [AC]-$x(1)$-V-$x(4)$-{ED} in PROSITE can be regarded as motifs of type Ω, where [AC] represents "Ala or Cys" and {ED} represents "any but Glu or Asp."

A *yes-no example* is a pair (α, β) of strings in Σ^* with $\alpha \neq \beta$. For a motif π and a yes-no example (α, β), we say that (α, β) is *good* for π if π accepts α but rejects β. A *yes-no sample* is a set $S = \{(\alpha^{(1)}, \beta^{(1)}), \ldots, (\alpha^{(m)}, \beta^{(m)})\}$ of yes-no examples. We call strings $\alpha^{(1)}, \ldots, \alpha^{(m)}$ *positive examples* and strings $\beta^{(1)}, \ldots, \beta^{(m)}$ *negative examples*. Then, for a motif π and a yes-no sample S, we define $\mathrm{cost}(S, \pi)$ to be the number of pairs in S which are good for π. Note that $\mathrm{cost}(S, \pi) = |L(\pi) \cap P| \times |(\Sigma^* - L(\pi)) \cap N|$ if a yes-no sample S is provided as $P \times N$ with two disjoint sets P and N of strings.

Let Ω be a type. The *best consensus motif problem for type* Ω is, given a yes-no sample S, to find a motif π of type Ω that maximizes $\mathrm{cost}(S, \pi)$.

Tateishi et al. [20] have shown that the best consensus motif problem is computationally intractable by proving with a rather heavy argument the NP-completeness of the decision version of the problem. Therefore, we have to develop approximate/heuristic strategies which shall work in practice for the best consensus motif problem. The purpose of this paper is to give a strategy coping with this computational difficulty.

Theorem 1 *[20] The best consensus motif problem is NP-complete for the following type:*

(1) Ω_1*: all nonempty subsets of* Σ*.*

(2) Ω_+*:* Σ^+*, all nonempty subsets of* $X(k)$ *for all* $k \geq 1$*, all nonempty subsets of* $X(i,j)$ *for all* $j > i \geq 1$*.*

The above results also hold even if a yes-no sample S *is provided as* $P \times N$ *of two disjoint sets* P *and* N *of strings in* $\{0,1\}^*$*.*

It should be noticed that the problem of deciding if there is a motif π of type Ω_1 such that *all* yes-no examples are good for π is easily solved in polynomial time (see Section). Thus the maximization problem has a sense. Although Ω_+ includes Ω_1, different arguments are required in [20] for the proofs.

3. Greedy Strategy for Best Consensus Motif Problem

Let Ω be a type. For $1 \leq k \leq n$, let $p_k : \Omega \to [0,1]$ be a probability distribution on Ω and let π_k be a random variable taking values in Ω with the probability distribution p_k, i.e., the probability of $\pi_k = Z$ is given by $p_k(Z)$ for Z in Ω.

We call an expression

$$\mu(\pi_1, \ldots, \pi_n) = Z_1^1 \cdots Z_{k_1}^1 \cdot \pi_1 \cdot Z_1^2 \cdots Z_{k_2}^2 \cdot \pi_2 \cdots \pi_{n-1} \cdot Z_1^n \cdots Z_{k_n}^n \cdot \pi_n \cdot Z_1^{n+1} \cdots Z_{k_{n+1}}^{n+1}$$

with $Z_1^t, \ldots, Z_{k_t}^t \in \Omega$ $(1 \leq t \leq n+1, k_t \geq 0$ for $1 \leq t \leq n+1)$ a *random motif* with random variables $\pi_1, \ldots, \pi_n$. We denote by $\mu(Y_1, \ldots, Y_n)$ the (random) motif obtained by substituting Y_k to π_k for $1 \leq k \leq n$. For a random motif $\mu(\pi_1, \ldots, \pi_n)$, we denote by $P\{(\alpha, \beta)$ is good for $\mu(\pi_1, \ldots, \pi_n)\}$ or, more simply, $P((\alpha, \beta), \mu(\pi_1, \ldots, \pi_n))$, the probability that a yes-no example (α, β) is good for $\mu(\pi_1, \ldots, \pi_n)$. Formally, let

$$H((\alpha, \beta), \mu(\pi_1, \ldots, \pi_n)) = \{(Y_1, \ldots, Y_n) \in \Omega^n \mid (\alpha, \beta) \text{ is good for } \mu(Y_1, \ldots, Y_n)\}.$$

Then $P((\alpha, \beta), \mu(\pi_1, \ldots, \pi_n))$ is given by

$$\sum_{(Y_1, \ldots, Y_n) \in H((\alpha,\beta), \mu(\pi_1, \ldots, \pi_n))} p_1(Y_1) \cdots p_n(Y_n).$$

For a yes-no sample $S = \{(\alpha^{(1)}, \beta^{(1)}), \ldots, (\alpha^{(m)}, \beta^{(m)})\}$ and a random motif $\mu(\pi_1, \ldots, \pi_n)$, the expected number $E(S, \mu(\pi_1, \ldots, \pi_n))$ of the yes-no examples in S which are good for $\mu(\pi_1, \ldots, \pi_n)$ is given by

$$E(S, \mu(\pi_1, \ldots, \pi_n)) = \sum_{i=1}^{m} P((\alpha^{(i)}, \beta^{(i)}), \mu(\pi_1, \ldots, \pi_n)).$$

Our greedy strategy shown in Fig. 1 assumes that the probability distributions $p_1, \ldots, p_n$ are known beforehand. Then it determines $Z_1, \ldots, Z_n$ in Ω consecutively in the following way: When $Z_1, \ldots, Z_{k-1}$ are determined, Z_k is set to be an element Z in Ω such that $E(S, \mu(Z_1, \ldots, Z_{k-1}, Z, \pi_{k+1}, \ldots, \pi_n)))$ is maximized, where $\pi_{k+1}, \ldots, \pi_n$ are random variables.

This greedy strategy requires (R1) and (R2) for effective implementation.

(R1) The expectation $E(S, \mu(Z_1, \ldots, Z_{k-1}, Z, \pi_{k+1}, \ldots, \pi_n))$ must be easily computable.

```
/* Let μ(π_1, ..., π_n) = π_1 ··· π_n be a random motif. */
/* This algorithm determines Z_1, ..., Z_n in Ω. */
for k ← 1 to n
   begin
      Find Z ∈ Ω maximizing E(S, μ(Z_1, ..., Z_{k-1}, Z, π_{k+1}, ..., π_n)));
      Z_k ← Z
   end
```

Figure 1: Greedy algorithm.

(R2) For Z in Ω, let $\mathrm{gain}(k, Z, S)$ be the difference

$$E(S, \mu(Z_1, \ldots, Z_{k-1}, Z, \pi_{k+1}, \ldots, \pi_n)) - E(S, \mu(Z_1, \ldots, Z_{k-1}, \pi_k, \ldots, \pi_n)).$$

In order to guarantee that the expectation does not decrease in each iteration, it must be shown that $\mathrm{gain}(k, Z, S) \geq 0$ for some Z in Ω for each $1 \leq k \leq n$. It is also a problem to give the probability distributions so that we can have such guarantee.

In Section 4, we deal with the case that this greedy strategy is applicable in practice by solving all these difficulties. Section 6 proves its usefulness with its experimental results. In Section 5, we consider a more general case and devise a heuristic algorithm for the best consensus motif problem although we have not yet succeeded in providing a sound theoretical basis.

4. Approximation Algorithm for Motifs of Type Ω_1

In this section we concentrate on the case that the type is $\Omega_1 = \{Z \mid \emptyset \neq Z \subseteq \Sigma\}$. For example, L-$x(6)$-L-$x(6)$-L-$x(6)$-L and [AC]-$x(1)$-V-$x(4)$-{ED} in Example 2 are motifs of type Ω_1. For a yes-no sample $S = \{(\alpha^{(1)}, \beta^{(1)}), \ldots, (\alpha^{(m)}, \beta^{(m)})\}$, we assume that

$$|\alpha^{(1)}| = |\beta^{(1)}| = \cdots = |\alpha^{(m)}| = |\beta^{(m)}|$$

since the length of a motif μ must be the same as the length of $\alpha^{(i)}$ if $(\alpha^{(i)}, \beta^{(i)})$ is good for μ.

The problem of finding, if any, a motif μ_0 of type Ω_1 such that *all* yes-no examples in S are good for μ_0 can be solved in polynomial time since μ_0 must be of the form $\mu_0 = \tilde{Z}_1 \cdots \tilde{Z}_n$ with $\tilde{Z}_k = \{\alpha_k^{(i)} \mid 1 \leq i \leq m\}$ for $1 \leq k \leq n$ and μ_0 must reject all negative examples from S, where $\alpha^{(i)} = \alpha_1^{(i)} \cdots \alpha_n^{(i)}$ with $\alpha_1^{(i)}, \ldots, \alpha_n^{(i)} \in \Sigma$. However, by Theorem 1 (1), the best consensus motif problem for Ω_1 is hardly solvable in polynomial time.

Let $\pi_1 \cdots \pi_n$ be a random motif with random variables $\pi_1, \ldots, \pi_n$ taking values in Ω_1. Let $p_k : \Omega_1 \to [0,1]$ be the probability distribution of π_k for $1 \leq k \leq n$. We first see that the expectation

$$E(S, Z_1 \cdots Z_{k-1} \pi_k \cdots \pi_n)$$

with $Z_1, \ldots, Z_{k-1}$ in Ω_1 is easily computable.

For $\sigma \in \Sigma$, let

$$S_\sigma = \{Z \in \Omega_1 \mid \sigma \in Z\}$$

and

$$p_k(S_\sigma) = \sum_{Z \in S_\sigma} p_k(Z).$$

For a string $\gamma = \gamma_1 \cdots \gamma_n$, let $\delta(\gamma, Z_1 \cdots Z_{k-1}) = 1$ if $\gamma_1 \cdots \gamma_{k-1} \in Z_1 \cdots Z_{k-1}$ else 0. Then for a yes-no example (α, β) ($\alpha = \alpha_1 \cdots \alpha_n$, $\beta = \beta_1 \cdots \beta_n$), the probability that (α, β) is good for $Z_1 \cdots Z_{k-1} \pi_k \cdots \pi_n$ is expressed as:

$$\begin{aligned} & P((\alpha, \beta), Z_1 \cdots Z_{k-1} \pi_k \cdots \pi_n) \\ &= \delta(\alpha, Z_1 \cdots Z_{k-1}) \cdot \left(\prod_{j=k}^{n} p_j(S_{\alpha_j}) - \delta(\beta, Z_1 \cdots Z_{k-1}) \cdot \prod_{j=k}^{n} p_j(S_{\alpha_j} \cap S_{\beta_j}) \right). \end{aligned}$$

This can be computed in polynomial time and therefore the expectation

$$E(S, \mu(\pi_1, \ldots, \pi_n)) = \sum_{i=1}^{n} P((\alpha^{(i)}, \beta^{(i)}), \mu(\pi_1, \ldots, \pi_n))$$

is also computable in polynomial time. This fulfills the requirement (R1) in Section 3. Since Ω_1 is a finite set, it is also trivial to find an element Z in Ω_1 that maximizes the expectation.

For satisfying the requirement (R2) in Section 3, Tateishi et al. [20] proved the following result:

Theorem 2 *[20] Let π_k be a random variable taking values in Ω_1 with a probability distribution p_k for $1 \leq k \leq n$. Let $s = |\Sigma|$. Assume that the probability distributions p_k $(1 \leq k \leq n)$ satisfy the following conditions:*

1. *$p_k(S_\sigma) \leq \frac{s+1}{2s}$ for all σ in Σ.*
2. *$p_k(S_\sigma \cap S_\tau) \geq \frac{s+2}{4s}$ for all σ, τ in Σ.*

Then, for each $1 \leq k \leq n$, there is some Z in Ω_1 such that $\mathrm{gain}(k, Z, S) \geq 0$, *where S is a yes-no sample.*

Theorem 2 has an advantage that it allows variations for motifs by specifying the probabilities for Z in Ω_1 as long as they satisfy the two conditions in Theorem 2. As a corollary of Theorem 2, we can prove the following lower bounds of the expectation:

Corollary 1 *Let $m = |S|$, $s = |\Sigma|$ and let n be the length of a motif.*

(1) If $p_k(Z) = \frac{|Z|}{s \cdot 2^{s-1}}$ for all Z in Ω_1 and for all $1 \leq k \leq n$, then

$$E(S, \pi_1 \cdots \pi_n) \geq \frac{m}{4} \cdot \left(\frac{s+1}{2s}\right)^{n-1}.$$

This is the case that any Z is allowed for a motif.

(2) If $p_k(\{\sigma\}) = \frac{1}{2s}$ for all σ in Σ, $p_k(\Sigma) = \frac{1}{2}$ and $p_k(Z) = 0$ for other Z in Ω_1 for all $1 \leq k \leq n$, then

$$E(S, \pi_1 \cdots \pi_n) \geq \frac{m}{2s} \cdot \left(\frac{s+1}{2s}\right)^{n-1}.$$

This is the case that only Σ and $\{\sigma\}$ for $\sigma \in \Sigma$ are allowed for a motif.

From Theorem 2, the greedy algorithm produces a motif $\pi = Z_1 \cdots Z_n$ with $\mathrm{cost}(S, \pi)$ at least as large as $E(S, \pi_1 \cdots \pi_n)$. The lower bounds of $E(S, \pi_1 \cdots \pi_n)$ in Corollary 1 are not good when n and s are larger. However, experiments in Section 6 show that our greedy strategy exhibits much better performance in a series of experiments on exon/intron splicing sites and promoter regions.

5. Heuristic Method for Finding More General Motifs

This section gives a greedy heuristic algorithm which can deal with a motif of the form

$$\Sigma^* Z_1 \cdots Z_n \Sigma^*$$

with $Z_1, \ldots, Z_n$ in Ω_1. For simplicity, we denote the above motif by $*Z_1 \cdots Z_n*$. Motifs of type Ω_1 in Section can cope with the case that the location of a segment of interest in a sequence is clear or determined beforehand. On the other hand, the motifs of the form $*Z_1 \cdots Z_n*$ are more flexible than the motifs of type Ω_1.

From a practical point of view, we consider how to apply the greedy strategy to this case by employing the random motif $\rho(\pi_1, \ldots, \pi_n) = *\pi_1 \cdots \pi_n*$ with probability distributions $p_1, \ldots, p_n$.

First, let $Z_1, \ldots, Z_{k-1}$ be elements in Ω_1. For a yes-no example (α, β), it is, in general, hard to compute efficiently the probability that (α, β) is good for $\rho(Z_1, \ldots, Z_{k-1}, \pi_k, \ldots, \pi_n)$. In practice, instead of evaluating the exact probability, we shall give a lower bound of the probability and use it for obtaining a rough estimation of $E(S, \rho(\pi_1, \ldots, \pi_n))$ for a yes-no sample S.

Let $\alpha = \alpha_1 \cdots \alpha_p$ and $\beta = \beta_1 \cdots \beta_q$, where $\alpha_j \in \Sigma$ for $1 \leq j \leq p$ and $\beta_k \in \Sigma$ for $1 \leq k \leq q$. We assume $n \leq p, q$. We consider the segments of length n of α

and β. Let $\alpha^j = \alpha_j \cdots \alpha_{j+n-1}$ for $1 \le j \le p-n+1$ and $\beta^k = \beta_k \cdots \beta_{k+n-1}$ for $1 \le k \le q-n+1$. We denote $\alpha_i^j = \alpha_{j+i-1}$ and $\beta_i^k = \beta_{k+i-1}$ for $1 \le i \le n$. First note that

$$\begin{aligned} & P\{(\alpha,\beta) \text{ is good for } \rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)\} \\ & \quad \ge \max_{1\le j\le p-n+1} P\{(\alpha^j,\beta) \text{ is good for } \rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)\}. \end{aligned}$$

Then we shall give a lower bound of the probability that (α^j,β) is good for $\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n))$.

In the argument below, for a set $U \subseteq \Omega_1^{n-k}$, $P(U)$ represents the probability of U, i.e., $P(U) = \sum_{(Y_k,\ldots,Y_n)\in U} p_k(Y_k)\cdots p_n(Y_n)$. Let

$$E^k(j) = \prod_{i=k}^{n} S_{\alpha_i^j}$$

for $1 \le j \le p-n+1$ and

$$I_t = \{(r_1,\ldots,r_t) \mid 1 \le r_1 < \cdots < r_t \le q-n+1\}$$

for $1 \le t \le q-n+1$. Then for each $(r_1,\ldots,r_t) \in I_t$, let

$$F^k(j,(r_1,\ldots,r_t)) = \prod_{i=k}^{n} (S_{\alpha_i^j} \cap S_{\beta_i^{r_1}} \cap \cdots \cap S_{\beta_i^{r_t}}).$$

Note that $F^k(j,(r_1,\ldots,r_t)) \subseteq E^k(j)$ for any $(r_1,\ldots,r_t)$. Recall that

$$\begin{aligned} & P\{(\alpha^j,\beta^r) \text{ is good for } \mu(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)\} \\ & = \delta(\alpha^j, Z_1\cdots Z_{k-1}) \cdot \left(\prod_{j=k}^{n} p_j(S_{\alpha_i^j}) - \delta(\beta^r, Z_1\cdots Z_{k-1}) \cdot \prod_{j=k}^{n} p_j(S_{\alpha_i^j} \cap S_{\beta_i^r}) \right) \\ & = \delta(\alpha^j, Z_1\cdots Z_{k-1}) \cdot P(E^k(j) - \delta(\beta^r, Z_1\cdots Z_{k-1}) \cdot F^k(j,r)) \end{aligned}$$

for $1 \le j \le p-n+1$ and $1 \le r \le q-n+1$, where $1 \cdot F^k(j,r) = F^k(j,r)$ and $0 \cdot F^k(j,r) = \emptyset$. For convenience, let $\tilde{F}^k(j,r) = \delta(\beta^r, Z_1\cdots Z_{k-1}) \cdot F^k(j,r)$. Then let

$$\begin{aligned} Q_0 &= P(E^k(j)) \\ Q_t &= \sum_{(r_1,\ldots,r_t)\in I_t} P(\tilde{F}^k(j,r_1) \cap \cdots \cap \tilde{F}^k(j,r_t)) \quad \text{for } 1 \le t \le q-n+1. \end{aligned}$$

Let $\delta_0 = \delta(\alpha^j, Z_1\cdots Z_{k-1})$. Then we have

$$\begin{aligned} & P\{(\alpha^j,\beta) \text{ is good for } \rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n))\} \\ & = \delta_0 \cdot P((E^k(j) - \tilde{F}^k(j,1)) \cap \cdots \cap (E^k(j) - \tilde{F}^k(j,q-n+1))) \\ & = \delta_0 \cdot P(E^k(j) - (\tilde{F}^k(j,1) \cup \cdots \cup \tilde{F}^k(j,q-n+1))) \\ & = \delta_0 \cdot P(E^k(j)) - \delta_0 \cdot P(\tilde{F}^k(j,1) \cup \cdots \cup \tilde{F}^k(j,q-n+1)) \\ & = \delta_0 \cdot (Q_0 - Q_1 + Q_2 - Q_3 + \cdots + (-1)^t Q_t + \cdots + (-1)^{q-n+1} Q_{q-n+1}). \end{aligned}$$

Note that $F^k(j,r_1) \cap \cdots \cap F^k(j,r_t) = F^k(j,(r_1,\ldots,r_t))$. Thus for $t \geq 1$

$$\begin{aligned} Q_t &= \sum_{(r_1,\ldots,r_t)\in I_t} \prod_{l=1}^{t} \delta(\beta^{r_l}, Z_1 \cdots Z_{k-1}) \cdot P(F^k(j,(r_1,\ldots,r_t)) \\ &= \sum_{(r_1,\ldots,r_t)\in I_t} \prod_{l=1}^{t} \delta(\beta^{r_l}, Z_1 \cdots Z_{k-1}) \cdot \prod_{i=k}^{n} p_i(S_{\alpha_i^j} \cap S_{\beta_i^{r_1}} \cap \cdots \cap S_{\beta_i^{r_t}}) \\ &= \sum_{(r_1,\ldots,r_t)\in I_t} \prod_{l=1}^{t} \delta(\beta^{r_l}, Z_1 \cdots Z_{k-1}) \cdot \prod_{i=k}^{n} p_i(S_{\{\alpha_i^j,\beta_i^{r_1},\ldots,\beta_i^{r_t}\}}), \end{aligned}$$

where $S_{\{\alpha_i^j,\beta_i^{r_1},\ldots,\beta_i^{r_t}\}} = \{Z \in \Omega_1 \mid \{\alpha_i^j,\beta_i^{r_1},\ldots,\beta_i^{r_t}\} \subseteq Z\}$. Q_t contains $O(n^t)$ terms. Thus, the total computation of $\sum_{t=0}^{q-n+1}(-1)^t Q_t$ requires exponential time. Therefore, we may take an odd constant integer $K > 0$ and let

$$\begin{aligned} &\tilde{P}((\alpha^j,\beta),\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)) \\ &\quad = \delta_0(Q_0 - Q_1 + Q_2 - Q_3 + \cdots - Q_K). \end{aligned}$$

Then we use

$$\begin{aligned} &\tilde{P}((\alpha,\beta),\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)) \\ &\quad = \max_{1\leq j\leq p-n+1} \tilde{P}((\alpha^j,\beta),\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)) \end{aligned}$$

as a lower bound for the estimation. The greedy algorithm in Fig. 1 uses

$$\begin{aligned} &\tilde{E}(S,\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)) \\ &\quad = \sum_{(\alpha,\beta)\in S} \tilde{P}((\alpha,\beta),\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n)) \end{aligned}$$

instead of the exact expectation which requires exponential time to compute.

No mathematical proofs are yet provided for the requirements (R1) and (R2) though Theorem 2 solves a special case of the problem. We implemented this heuristic algorithm by modifying $\tilde{P}((\alpha^j,\beta),\rho(Z_1,\ldots,Z_{k-1},\pi_k,\ldots,\pi_n))$ into $\delta_0(\sum_{(r_1)\in I_1} Q_0 - Q_1)$ instead of taking a large $K \geq 1$ so that it shall work in practice. Some experimental results shall be given in Section 6.

6. Results

The purpose of this section is to evaluate the performance of the greedy strategy in Fig. 1 by implementing two kinds of algorithms. The first is the greedy algorithm for type Ω_1 discussed in Section 4. We denote this algorithm by GREEDY[Ω_1]. We use the probability distributions $p_1,\ldots,p_n$ given in Corollary 1 (1). The second is the heuristic algorithm for finding a motif of

the form $\Sigma^* Z_1 \cdots Z_n \Sigma^*$ discussed in Section 5. This algorithm is denoted by GREEDY$[*\Omega_1*]$.

6.1. Experiments by GREEDY$[\Omega_1]$

For testing GREEDY$[\Omega_1]$, we use data on exon/intron splicing sites and *E.coli* promoters since good statistics are known and the problem is very suited for the motifs of type Ω_1. The approach with HMM [7, 15] also seems very suited for the best consensus motif problem for type Ω_1 by definition. We ha not compared with our strategy with the HMM approach on the same data.

6.1.1. Exon/intron splicing sites

We shall present some experimental results of the algorithm on exon/intron splicing sites. For the coding region identification problem, there are many papers aiming at predicting splicing sites. For example, Brunak et al. [4] applied neural networks to predict splicing site locations in human pre-mRNA. Some software is also available [19].

Examples are taken from GenBank. Positive examples are sequences of length 60 each of which comprises a segment of length 30 in the exon and a segment of length 30 in the intron. Negative examples are sequences which are known not to include any splicing sites.

The first sample S_{ei300} consists of 300 yes-no examples randomly generated from the above positive and negative examples. The motif length is 60. From 300 yes-no examples, 269 yes-no examples are good for a very (too) simple motif (Fig. 2). It accepts all positive examples but does not reject 31 negative examples. This motif is consistent with the GT-AG rule [21].

The second sample $S_{ei50:100}$ was constructed in the following way: First, we took 50 positive examples $\alpha_1, \ldots, \alpha_{50}$ and 100 negative examples $\beta_1, \ldots, \beta_{100}$. Then for each positive example α_i, two yes-no examples $(\alpha_i, \beta_{10(i-1)+j})$ $(j = 1, 2)$ are generated. Thus in total, 100 yes-no examples are generated. Fig. 3 shows the result. GREEDY$[\Omega_1]$ found a motif for which 97 yes-no examples are good (97%).

The third sample $S_{ei10:100}$ with 100 yes-no examples was constructed from 10 positive examples and 100 negative examples in the same way. Fig. 4 shows the result. GREEDY$[\Omega_1]$ found a motif for which all yes-no examples are good (100%).

6.1.2. *E. coli* promoters

For testing GREEDY$[\Omega_1]$, the *E. coli* promoters are also good examples since

```
-30        -20        -10          1        10         20         30
  ..............................gc.............................
                                 t
```

Figure 2: Motif $x(30)$-g-[ct]-$x(28)$ for exon/intro splicing sites. From 269 yes-no examples in S_{ei300} are good for this motif.

```
-30        -20        -10          1        10         20         30
  ..............................gta.a.........................
                                   c c
                                   g g
```

Figure 3: 97 yes-no examples in $S_{ei50:100}$ are good for the above motif. Letters in vertical line, for example a, c, g at position 3, represent the set consisting of these letters.

those sites are very well characterized [9, 14, 6]. Examples are again taken from GenBank. Positive and negative examples are taken from a sequence as shown in Fig. 5.

$S_{e.coli10:100}$ are generated in the same way as $S_{ei10:100}$ by taking 10 positive examples and 100 negative examples. Fig. 6 shows the result. GREEDY[Ω_1] found a motif for which all yes-no examples are good (100%).

6.2. Experiments by GREEDY[$*\Omega_1*$]

Eukaryotic promoters are more complex and larger than *E. coli* promoters. No definite consensus has been built for eukaryotic promoters although some major elements are known, such as CCAAT, GC and TATA boxes.

Positive and negative examples for eukaryotic promoters are collected in the same way as *E. coli* promoters. We took three samples S_1, S_2 and S_3 shown in Table 1.

Experiments were done for these samples S_1, S_2 and S_3 by changing motif the motif length from 10 to the maximum length in step 10. Fig. 7 shows an example of a motif of length 40 that GREEDY[$*\Omega_1*$] found from S_1. Only 6 yes-no examples are good for this motif though there seems to be a TATA-like region

```
-30        -20        -10          1        10         20         30
  .a................a...a....aacgtaag...ca.a...cccc.a.c.......
   c               c   c    ccg  cc    gc c   gggg c g
   g               g   g    gg   gt     g g   tttt g t
```

Figure 4: All (100) yes-no examples in $S_{ei10:100}$ are good for the above motif.

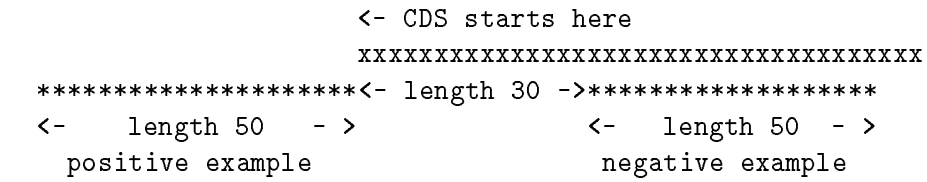

Figure 5: Positive and negative examples for $S_{e.coli10:100}$.

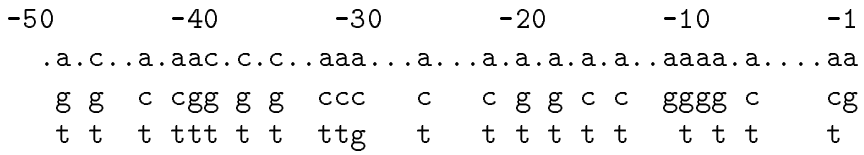

Figure 6: All (100) yes-no examples in $S_{e.coli10:100}$ are good for the above motif.

Table 1: Three samples for eukaryotic promoters. For example, S_1 consists of 20 yes-no examples constructed from 10 positive and 20 negative examples whose length is 50.

Sample	Yes-No Examples	Positive	Negative	Length
S_1	20	10	20	50
S_2	20	20	20	100
S_3	50	50	50	50

```
..............a.a.aaa.........aaaa.aa..t
              c    t           cccc cc
              t                tgtg tt
```

Figure 7: A motif of length 40 found from S_1.

in the motif. As a whole, experimental results by GREEDY[$*\Omega_1*$] on eukaryotic promoters are not attractive in any case. It seems difficult for GREEDY[$*\Omega_1*$] to find interesting motifs for eukaryotic promoters. Eukaryotic promoters are less well characterized [5]. It seems more reasonable to find motifs of the form $*\Omega_1 * \Omega_1 * \cdots * \Omega_1*$. However, currently, we do not have any efficient strategy for finding such motifs.

7. Conclusion

Motivated by the problem of extracting motifs, such as in PROSITE, we have developed a greedy strategy for finding motifs from positive and negative sequences by exploiting probabilistic arguments. We presented some of the experimental results on *E. coli* and eukaryotic promoters and splicing sites. The results by GREEDY[Ω_1] are not so much astonishing, but reasonable knowledge are found by our strategy. The lower bounds in Corollary 1 are not good but these experiments showed that the performance of our greedy algorithm is much better. The experimental results by GREEDY[$*\Omega_1*$] on eukaryotic promoters are unfortunately not successful. In order to characterize eukaryotic promoters, we need another strategy that can cope with more complicated motifs. In this paper, we have dealt with only DNA sequences and ignored amino acid sequences. This is simply because the alphabet of 20 symbols is too large for our algorithm since the time complexity increases exponentially with respect to the size of the alphabet. The difference between 2^4 and 2^{20} is very serious in our algorithm. From both theory and practice, it is an challenging open problem to devise efficient approximation algorithms for finding such general motifs together with proofs guaranteeing their performance.

Aiming at knowledge discovery from amino acid sequences, the second author's research group has developed a system called BONSAI [2, 16, 18] that produces, from positive and negative examples, a mapping ψ called an *alphabet indexing* which classifies twenty amino acid residues into a smaller categories and a decision tree whose internal nodes are labeled with regular patterns. Since regular patterns are used for making decisions at nodes, only exact pattern matching is allowed. We are planning to implement the strategy developed in this paper in a forthcoming version of BONSAI so that it can cope with sequences with ambiguity.

Acknowledgments

The authors are very grateful to Naohiro Furukawa, Daisuke Ikeda and Ayumi Shinohara for their excellent advises. The authors would also thank Satoru Kuhara for his guidance in Genome Informatics. This work is partly supported by Grant-in-Aid for Scientific Research on Priority Area "Genome Informatics," No. 06558047 and No. 06680326 from Ministry of Education, Science and Culture, Japan.

References

1. Angluin, D., Finding patterns common to a set of strings, *J. Comput. System Sci.* **21** (1980) 46–62.
2. Arikawa, S., Miyano, S., Shinohara, A., Kuhara, S., Mukouchi, Y., and Shinohara, T., A machine discovery from amino acid sequences by decision trees over regular patterns, *New Generation Computing* **11** (1993) 361–375.
3. Bairoch, A., PROSITE: a dictionary of sites and patterns in proteins, *Nucleic Acids Res.* **19** (1991) 2241–2245.
4. Brunak, S., Engelbrecht, J., and Knudsen, S., Prediction of human mRNA donar and acceptor sites from the DNA sequence, *J. Mol. Biol.* **220** (1991) 49–65.
5. Bucher, P., Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences, *J. Mol. Biol.* **212** (1990) 563–578.
6. Cooke, D.E. and Hunt, J.E., Recognising promoter sequences using an artificial immune system, *Proc. Third Int. Conf. Intelligent Systems for Molecular Biology*, 1995, 89–97.
7. Fujiwara, Y., Asogawa, M. and Konagaya, A., Stochastic motif extraction using hidden Markov model, *Proc. Second Int. Conf. Intelligent Systems for Molecular Biology*, 1994, 121–129.
8. Gribskov, M. and Devereux, J., *Sequence Analysis Primer*, Stockholm Press, 1991.
9. Hawley, D.K. and McClure, W.R., Compilation and analysis of *Escherichia coli* promoter sequences, *Nucleic Acids Res.* **11** (1983) 2237–2255.
10. Jiang, T. and Li, M., On the complexity of learning strings and sequences, *Proc. 4th Workshop on Computational Learning Theory*, 1991, 367–371.
11. Johnson, D.S., Approximation algorithms for combinatorial problems, *J. Comput. System Sci.* **9** (1974) 256–278.
12. Mair, D., The complexity of some problems on subsequences and super-

sequences, *J. Assoc. Comput. Mach.* **25** (1977) 322–336.
13. Miyano, S., Shinohara, A. and Shinohara, T., Which classes of elementary formal systems are polynomial-time learnable?, *Proc. Second Workshop on Algorithmic Learning Theory*, 1991, 139–150.
14. Oliphant, A.R. and Struhl, K., Defining the consensus sequences of *E. coli* promoter elements by random selection, *Nucleic Acids Res.* **16** (1988) 7673–7683.
15. Raman, R. and Overton, G.C., Application of hidden Markov model to the characterization of transcription factor binding sites, *Proc. 26th Hawaii Int. Conf. System Sciences*, Vol. V, 1994, 275–283
16. Shimozono, S., Shinohara, A., Shinohara, T., Miyano, S., Kuhara, S., and Arikawa, S., Knowledge acquisition from amino acid sequences by machine learning system BONSAI, *Transactions of Information Processing Society of Japan* **35** (1994) 2009–2018.
17. Shinohara, T., Polynomial time inference of extended regular pattern languages, *Lecture Notes in Computer Science* **147** (1983) 115–127.
18. Shoudai, T., Lappe, M., Miyano, S., Shinohara, A., Okazaki, T., Arikawa, S., Uchida, T., Shimozono, S., Shinohara, T., and Kuhara, S., BONSAI Garden: parallel knowledge discovery system for amino acid sequences, In *Proc. Third International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, 1995, 359–366.
19. Staden, R., Finding protein coding regions in genomic sequences, *Meth. Enzym.* **183** (1990) 163–180.
20. Tateishi, E., Maruyama and Miyano, S., Extracting motifs from positive and negative sequence data, RIFIS-TR-CS-115, Research Institute of Fundamental Information Science, Kyushu University, May, 1995.
21. Watson, J.D., Hopkins, N.H., Roberts, J.W., Steitz, J.A. and Weiner, A.M., *Molecular Biology of the Gene*, Fourth Edition, The Benjamin/Cummings Pub. Co., Inc., 1987.
22. Wu, T.D. and Brutlag, D.L., Identification of protein motifs using conserved amino acid properties and partitioning techniques, In *Proc. Third International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, 1995, 402–410.
23. Yannakakis, M., On the approximation of maximum satisfiability, *J. Algorithms* **17** (1994) 475–502.