

## Sequence analysis

# A linear programming approach for identifying a consensus sequence on DNA sequences

Han-Lin Li\* and Chang-Jui Fu

Institute of Information Management, National Chiao Tung University, Taiwan, China

Received on August 6, 2003; revised on January 11, 2005; accepted on January 18, 2005

Advance Access publication January 25, 2005

**ABSTRACT**

**Motivation:** Maximum-likelihood methods for solving the consensus sequence identification (CSI) problem on DNA sequences may only find a local optimum rather than the global optimum. Additionally, such methods do not allow logical constraints to be imposed on their models. This study develops a linear programming technique to solve CSI problems by finding an optimum consensus sequence. This method is computationally more efficient and is guaranteed to reach the global optimum. The developed method can also be extended to treat more complicated CSI problems with ambiguous conserved patterns.

**Results:** A CSI problem is first formulated as a non-linear mixed 0-1 optimization program, which is then converted into a linear mixed 0-1 program. The proposed method provides the following advantages over maximum-likelihood methods: (1) It is guaranteed to find the global optimum. (2) It can embed various logical constraints into the corresponding model. (3) It is applicable to problems with many long sequences. (4) It can find the second and the third best solutions. An extension of the proposed linear mixed 0-1 program is also designed to solve CSI problems with an unknown spacer length between conserved regions. Two examples of searching for CRP-binding sites and for FNR-binding sites in the *Escherichia coli* genome are used to illustrate and test the proposed method.

**Availability:** A software package, Global Site Seer for the Microsoft Windows operating system is available by <http://www.iim.nctu.edu.tw/~cjfu/gss.htm>

**Contact:** hlli@cc.nctu.edu.tw

**INTRODUCTION**

The methods for determining a consensus pattern can be split into two parts. The first part is the model for describing the shared pattern; the second part is the algorithm for identifying the optimal common site according to its shared pattern. This study belongs to the second part. A consensus sequence identification (CSI) problem is, given a set of sequences known to contain binding sites for a common factor but not knowing where the sites are, discovers the location of the sites in each sequence (Stormo, 2000).

The CSI problem is critical in research on gene expression such as the protein-binding site in a DNA strand. For the last decade several good methods have been developed for solving such problems (Brazma *et al.*, 1998). Of these, the maximum-likelihood approach (Stormo and Hartzell, 1989; Hertz *et al.*, 1990) is the best known.

The traditional maximum-likelihood approach, which measures information content to determine alignments, works fairly well and can be relied upon to discover the common sites. However, they are still not able to determine the complete set of regulatory interactions for complicated promoters typical of metazoans (Stormo, 2000).

Recently, Ecker *et al.* (2002) utilized optimization techniques to reformulate the maximum-likelihood approach for solving CSI problems. They adopted a probabilistic model and formulated a well-designed non-linear model with reference to the expectation maximization algorithm of Lawrence and Reilly (1990). Their method, however, occasionally only finds a feasible solution or a local optimum: which means the best solution may not be found. Additionally, no further structural feature in a CSI problem can be embedded conveniently in their model.

This study proposes a linear programming method for solving a CSI problem to reach the globally optimal consensus sequence. Two examples of searching for CRP-binding sites and for FNR-binding sites in the *Escherichia coli* genome are used to illustrate the proposed method. The CSI problem is first formulated as a non-linear mixed 0-1 program for alignment of DNA sequences; each of the four bases are coded with two binary variables and a matching score is designed. This non-linear mixed 0-1 program is then converted into a linear mixed 0-1 program by linearization techniques. This study decomposes a CSI problem into several subprograms to be solved by a set of distributed computers linked via the internet. Owing to some special features of the binary relationships, this linear 0-1 program includes  $2m$  binary variables where  $m$  is the number of active letters in the common site. Some very attractive properties of this method are:

- (1) The required number of binary variables is independent of the number of sequences and the size of each sequence. That means, the proposed method is computationally efficient in solving a CSI problem with a large data size.
- (2) The proposed method is guaranteed to find the global optimum instead of a local optimum.
- (3) Many kinds of specific features accompanied with a CSI problem can be formulated straightforwardly as logical constraints and embedded into the linear program.

An example of searching CRP-binding sites, as discussed in Stormo and Hartzell (1989) and Ecker *et al.* (2002), is described as follows. Given 18 letter sequences, each 105 positions long, where each position contains a letter from the set {A, T, C, G}, find a common site of length 16 with the pattern

$$L_1 L_2 L_3 L_4 L_5 \square \square \square \square \square L_6 L_7 L_8 L_9 L_{10}$$

\*To whom correspondence should be addressed.

where  $L_i, \square \in \{A, T, C, G\}$  and  $\square$ s mean the positions of ignored letters.

Restated, the problem is to specify

- (1) the  $L_i$ s of the common site pattern and
- (2) the location of the site in each given sequence, which can fit most closely the common site.

The following are difficulties associated with the method of Ecker *et al.* (2002) and other maximum-likelihood methods (as reviewed in Brazma *et al.*, 1998) for solving a CSI problem:

(1) *Only a local optimal or feasible solution is obtained.* Since Ecker *et al.* (2002) formulated a CSI problem as a non-convex non-linear program, their method may only find local optima, as has been acknowledged. Other maximum-likelihood methods, which intend to maximize the probability of binding to the promoters in the sequences, may only find a feasible solution instead of finding a local optimal solution. It is not guaranteed that current maximum-likelihood methods can reach the global optimum for general CSI problems.

(2) *Heavy computational burden.* The non-linear program in Ecker *et al.* (2002) contains too many non-linear terms. The heavy computational burden in their method prohibits it from treating a CSI problem with a large number of sequences.

(3) *Difficulty of adding logical constraints.* When identifying protein binding sites, there usually exists some specific features to be considered as logical constraints. For example, the letters of position  $L_i$  and  $L_{11-i}$  are expected to be complementary (i.e. G–C and A–T). Formulating such a constraint in maximum-likelihood approaches is a complex task. It is even impossible to formulate even more complicated logical constraints (e.g. those with some ambiguity) when applying these approaches.

(4) *Fixed number of ignored letters.* Maximum-likelihood methods are mainly used to solve CSI problems with fixed number of ignored letters (e.g. six in the above example). However, in the real world this number is unknown and needs to be found by some preliminary processes.

(5) *Difficulty of finding the second and the third best solutions.* Since current methods may only find a local optimum, it is hard to find other solutions next to the best solution.

In order to overcome the above difficulties of solving a CSI problem, this study proposes a novel method to treat the same problem that molecular biologists actually are interested in solving. We formulate a CSI problem as the identification of a consensus sequence that minimizes the number of differences between the proposed sites. Our basic concept is to reformulate a CSI problem as a mixed 0-1 linear program which only contains a limited number of 0-1 variables and most variables are continuous. Such a mixed 0-1 linear program can be solved effectively by commonly used branching-and-bound algorithms or a branch-cut algorithm (Balas *et al.*, 1996). The advantages of the proposed method are listed below:

- (1) It is guaranteed to find the globally optimal solution. Since the objective function and constraints are all linear, the program should converge to the global optimum.
- (2) It can effectively solve a CSI problem by a set of on-line computers as illustrated by our numerical experiments.

**Table 1.** Base code in the determined common site

Base	$u_i$	$v_i$	$a_i$	$t_i$	$c_i$	$g_i$
A	0	0	1	0	0	0
T	1	1	0	1	0	0
C	0	1	0	0	1	0
G	1	0	0	0	0	1

- (3) It is convenient to add logical constraints. Since the binary variables are suited to express logical relationship, various complicated constraints can be embedded directly into the proposed method.
- (4) It can be extended to treat CSI problems with unknown number of ignored letters.
- (5) It is very straightforward to find the complete set of the second, third, etc. best consensus sequences.

In the following section we will discuss the linear programming technique for solving a CSI problem.

## PROPOSED METHOD

This study first formulates a CSI problem as a non-linear mixed 0-1 program. Then it converts this non-linear mixed 0-1 program into a linear mixed 0-1 program using linearization techniques. To reduce the computational burden, many 0-1 variables in this linear mixed 0-1 program can actually be solved as continuous variables by an all or nothing assignment technique which greatly improves the computational efficiency of this program.

### Non-linear mixed 0-1 program

Here we use the example data in Stormo and Hartzell (1989), as listed in Appendix section, to describe the proposed method. First, represent the data in Appendix section as an  $18 \times 105$  data matrix  $D$ :

$$D = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,105} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,105} \\ \vdots & \vdots & \ddots & \vdots \\ b_{18,1} & b_{18,2} & \cdots & b_{18,105} \end{bmatrix}, \quad (1)$$

where  $b_{l,p}$  is the letter in the position  $p$  of the sequence  $l$ .

Recall the example discussed in the previous section. The common site we want to find has 16 positions (10  $L_i$ s and 6 ignored letters), a sequence has 90 corresponding sites, so an  $18 \times 900$  data matrix  $D'$  is generated from  $D$ :

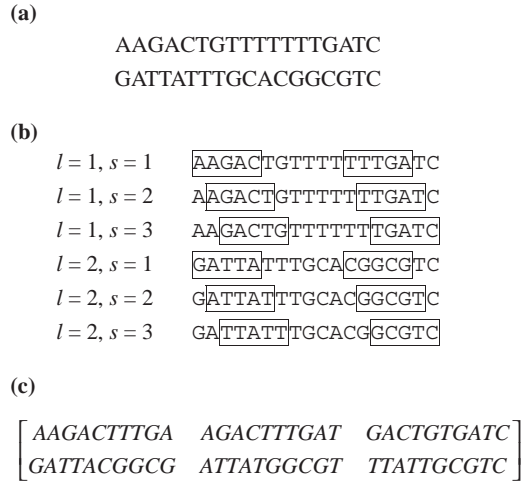
$$D' = \begin{bmatrix} d_{1,1}^1 & \cdots & d_{1,1}^{10} & d_{1,2}^1 & \cdots & d_{1,2}^{10} & \cdots & d_{1,90}^1 & \cdots & d_{1,90}^{10} \\ d_{2,1}^1 & \cdots & d_{2,1}^{10} & d_{2,2}^1 & \cdots & d_{2,2}^{10} & \cdots & d_{2,90}^1 & \cdots & d_{2,90}^{10} \\ \vdots & & \vdots & & & \ddots & & \vdots & & \vdots \\ d_{18,1}^1 & \cdots & d_{18,1}^{10} & d_{18,2}^1 & \cdots & d_{18,2}^{10} & \cdots & d_{18,90}^1 & \cdots & d_{18,90}^{10} \end{bmatrix}, \quad (2)$$

where

$$d_s^i = \begin{cases} b_{l,i+s-1} & (\text{for } i = 1, 2, \dots, 5) \\ b_{l,i+s+5} & (\text{for } i = 6, 7, \dots, 10), \end{cases}$$

and  $s = 1, \dots, 90$  is the starting position of each candidate site.

For  $L_i \in \{A, T, C, G\}$ , two binary variables  $u_i$  and  $v_i$  can be used to express  $L_i$ , an element of the consensus sequence, as shown in Table 1.



**Fig. 1.** A small example of finding consensus sequence: (a) two sequences to be compared; (b) schematic representation of the candidate sites; (c) the associated  $D'$  matrix.

Table 1 indicates that if  $L_i$  is A, T, C or G, then  $a_i = 1$ ,  $t_i = 1$ ,  $c_i = 1$  or  $g_i = 1$ , which implies the following conditions:

$$\begin{aligned} a_i &= (1 - u_i)(1 - v_i) \\ t_i &= u_i v_i \\ c_i &= (1 - u_i)v_i \\ g_i &= u_i(1 - v_i). \end{aligned} \quad (3)$$

Now let  $\text{Score}_l$  be the degree of fitting to the common site found, specified as

$$\text{Score}_l = \sum_{s=1}^{90} z_{l,s} (\theta_{l,s}^1 + \theta_{l,s}^2 + \cdots + \theta_{l,s}^{10}), \quad (4)$$

where  $\theta_{l,s}^i$  is the element of candidate sites extracted from  $D'$ . The constraints associated with Equation (4) are:

$$\begin{aligned} \sum_{s=1}^{90} z_{l,s} &= 1, \quad z_{l,s} \in \{0, 1\} \quad \text{for all } l \text{ and } s. \\ \theta_{l,s}^i &= \begin{cases} a_i & \text{if } d_{l,s}^i = A \\ t_i & \text{if } d_{l,s}^i = T \\ c_i & \text{if } d_{l,s}^i = C \\ g_i & \text{if } d_{l,s}^i = G. \end{cases} \end{aligned} \quad (5)$$

Clearly,  $0 \leq \text{Score}_l \leq 10$ . And the objective is to maximize the total sum of  $\text{Score}_l$ .

Consider the sample data in Figure 1 for instance.

$$\begin{aligned} \text{Score}_1 &= z_{1,1}(a_1 + a_2 + g_3 + a_4 + c_5 + t_6 + t_7 + t_8 + g_9 + a_{10}) \\ &\quad + z_{1,2}(a_1 + g_2 + a_3 + c_4 + t_5 + t_6 + t_7 + g_8 + a_9 + t_{10}) \\ &\quad + z_{1,3}(g_1 + a_2 + c_3 + t_4 + g_5 + t_6 + g_7 + a_8 + t_9 + c_{10}) \end{aligned} \quad (7)$$

$$\begin{aligned} \text{Score}_2 &= z_{2,1}(g_1 + a_2 + t_3 + t_4 + a_5 + c_6 + g_7 + g_8 + c_9 + g_{10}) \\ &\quad + z_{2,2}(a_1 + t_2 + t_3 + a_4 + t_5 + g_6 + g_7 + c_8 + g_9 + t_{10}) \\ &\quad + z_{2,3}(t_1 + t_2 + a_3 + t_4 + t_5 + g_6 + c_7 + g_8 + t_9 + c_{10}) \end{aligned} \quad (8)$$

All  $z_{l,s}$  in Equation (4) are binary variables. Equation (5) implies that for a sequence  $l$ , only one site is chosen and no other sites contribute to  $\text{Score}_l$ . Suppose the  $k$ -th site is selected, then  $z_{l,k} = 1$  and  $z_{l,s} = 0$  for all  $s \in \{1, 2, \dots, 90\}$ ,  $s \neq k$ . Since a huge amount of  $z_{l,s}$  (i.e.  $|l| \times |s|$ ) are involved, to treat  $z_{l,s}$  as binary variables would cause a heavy computational burden. Therefore  $z_{l,s}$  should be resolved as continuous variables rather than binary

variables. An important proposition is introduced below:

**PROPOSITION 1** (All or nothing assignment). Let  $z_{l,s} \geq 0$  be continuous variables instead of binary variables. If there is a  $k$ ,  $k \in \{1, 2, \dots, 90\}$ , such that  $\sum_{i=1}^{10} \theta_{l,k}^i = \max \left\{ \sum_{i=1}^{10} \theta_{l,s}^i \text{ for } s = 1, 2, \dots, 90 \right\}$ , then assigning  $z_{l,k} = 1$  and  $z_{l,s} = 0$  for all  $s \neq k$ ,  $s \in \{1, 2, \dots, 90\}$ , can maximize the value of  $\text{Score}_l$ .

**PROOF.** Since  $\sum_s z_{l,s} = 1$  and  $z_{l,s} \geq 0$ , it is true that

$$\begin{aligned} &\max \left\{ \sum_s \left( z_{l,s} \sum_i \theta_{l,s}^i \right) \right\} \\ &\leq \max \left\{ \sum_i \theta_{l,s}^i \text{ for } s = 1, 2, \dots, 90 \right\} = \sum_i \theta_{l,k}^i. \end{aligned}$$

**REMARK 1.** The objective function of a CSI problem  $f(x)$  can be rewritten as

$$\begin{aligned} f(x) &= \sum_{i=1}^{10} \left\{ a_i \sum_{(l,s) \in SA_i} z_{l,s} + t_i \sum_{(l,s) \in ST_i} z_{l,s} + c_i \sum_{(l,s) \in SC_i} z_{l,s} \right. \\ &\quad \left. + g_i \sum_{(l,s) \in SG_i} z_{l,s} \right\}, \end{aligned} \quad (9)$$

where  $SA_i = \{(l, s) | d_{l,s}^i = A\}$ ,  $ST_i = \{(l, s) | d_{l,s}^i = T\}$ ,  $SC_i = \{(l, s) | d_{l,s}^i = C\}$  and  $SG_i = \{(l, s) | d_{l,s}^i = G\}$  for  $i = 1, 2, \dots, 10$ .

This result implies that  $SA_i$  (or  $ST_i, SC_i, SG_i$ ) is a set composed of  $(l, s)$  in which the product term  $z_{l,s} a_i$  (or  $z_{l,s} t_i, z_{l,s} c_i, z_{l,s} g_i$ ) appears on the right hand side of Equation (4) because that  $\theta_{l,s}^i = a_i$ .

For instance, the sum of  $\text{Score}_1$  and  $\text{Score}_2$  in Equations (7) and (8) becomes

$$\begin{aligned} \text{Score}_1 + \text{Score}_2 &= a_1(z_{1,1} + z_{1,2} + z_{2,2}) + \cdots + a_{10}z_{1,1} \\ &\quad + \cdots + g_1(z_{1,3} + z_{2,1}) + \cdots + g_{10}z_{2,1}. \end{aligned} \quad (10)$$

Some logical constraints can be conveniently expressed by binary variables. For instance, the constraint that a CRP dimer binds a symmetrical site requires that

$$\text{if } L_i = \begin{cases} A & \text{then } L_{11-i} = T, \\ C & \text{then } L_{11-i} = G. \end{cases}$$

Such a logical structure can be formulated conveniently as the following constraints.

$$\begin{aligned} u_i + u_{11-i} &= 1 \\ v_i + v_{11-i} &= 1 \end{aligned} \quad \text{for } i = 1, 2, 3, 4, 5 \quad (11)$$

where  $u_i, v_i, u_{11-i}, v_{11-i} \in \{0, 1\}$ .

With reference to Table 1, clearly if  $L_i = A$  (i.e.  $u_i = 0$  and  $v_i = 0$ ) then  $L_{11-i} = T$  (i.e.  $u_{11-i} = 1$  and  $v_{11-i} = 1$ ) and vice versa; if  $L_i = C$  (i.e.  $u_i = 0$  and  $v_i = 1$ ) then  $L_{11-i} = G$  (i.e.  $u_{11-i} = 1$  and  $v_{11-i} = 0$ ) and vice versa. A CSI problem can then be formulated as a non-linear mixed 0-1 program below based on these constraints:

**Program 1** (Non-linear 0-1 CSI program)

$$\begin{aligned} \text{Maximize} \quad &\sum_{l=1}^{18} \text{Score}_l = \sum_{i=1}^{10} \left\{ a_i \sum_{(l,s) \in SA_i} z_{l,s} + t_i \sum_{(l,s) \in ST_i} z_{l,s} \right. \\ &\quad \left. + c_i \sum_{(l,s) \in SC_i} z_{l,s} + g_i \sum_{(l,s) \in SG_i} z_{l,s} \right\} \end{aligned} \quad (12)$$

$$\begin{aligned}
 &\text{subject to } \sum_{s=1}^{90} z_{l,s} = 1, \quad z_{l,s} \geq 0 \quad \text{for all } l, s \\
 &\left. \begin{aligned} a_i &= (1 - u_i)(1 - v_i) \\ t_i &= u_i v_i \\ c_i &= (1 - u_i)v_i \\ g_i &= u_i(1 - v_i) \end{aligned} \right\} \begin{aligned} &\text{Conservative constraints} \\ &\text{for } i = 1, 2, \dots, 10 \end{aligned} \\
 &\left. \begin{aligned} u_i + u_{11-i} &= 1 \\ v_i + v_{11-i} &= 1 \end{aligned} \right\} \begin{aligned} &\text{Logical constraints} \\ &\text{for } i = 1, 2, \dots, 5 \end{aligned} \\
 &u_i, v_i \in \{0, 1\} \quad \text{for } i = 1, 2, \dots, 5 \\
 &0 \leq u_i, v_i \leq 1 \quad \text{for } i = 6, 7, \dots, 10 \\
 &0 \leq a_i, t_i, c_i, g_i \leq 1 \quad \text{for } i = 1, 2, \dots, 10.
 \end{aligned}$$

This program intends to solve  $\{a_i, t_i, c_i, g_i\}$  for  $i = 1, 2, \dots, 10$  thus to maximize the total degree of fitting to the common site for the given 18 sequences, subjected to a possible logical constraint. A very important feature of Program 1 is that we can treat  $z_{l,s}$  as continuous variables rather than binary variables, which can improve the computational efficiency dramatically. We can ensure all found  $z_{l,s}$  will still have binary values as discussed in the next section.

### Linearization of Program 1

Program 1 is a mixed non-linear 0-1 program where  $q_i \sum z_{l,s}$  for  $q_i \in \{a_i, t_i, c_i, g_i\}$  and  $u_i v_i$  are product terms. These product terms can be linearized directly by the following propositions:

PROPOSITION 2. The product term  $\lambda_i = q_i \sum z_{l,s}$  where  $\lambda_i$  is to be maximized and  $q_i \in \{0, 1\}$  can be linearized as follows:

$$\begin{aligned}
 \lambda_i &\geq \sum z_{l,s} + M(q_i - 1) \\
 \lambda_i &\geq 0 \\
 \lambda_i &\leq \sum z_{l,s} \\
 \lambda_i &\leq Mq_i,
 \end{aligned} \tag{13}$$

where  $M$  is a big constant  $\geq$  the number of sequences.

PROOF. If  $q_i = 1$  then  $\lambda_i = \sum z_{l,s}$ ; and otherwise  $\lambda_i = 0$ .

PROPOSITION 3. The product term  $w_i = u_i v_i$  where  $u_i, v_i \in \{0, 1\}$  can be linearized as follows:

$$\begin{aligned}
 w_i &\leq u_i \\
 w_i &\leq v_i \\
 w_i &\geq 0 \\
 w_i &\geq u_i + v_i - 1.
 \end{aligned} \tag{14}$$

Denote  $Z(a_i) = a_i \sum_{(l,s) \in SA_i} z_{l,s}$ ,  $Z(t_i) = t_i \sum_{(l,s) \in ST_i} z_{l,s}$ ,  $Z(c_i) = c_i \sum_{(l,s) \in SC_i} z_{l,s}$  and  $Z(g_i) = g_i \sum_{(l,s) \in SG_i} z_{l,s}$ . Program 1 is then linearized into Program 2 based on Propositions 2 and 3.

Program 2 (Linear mixed 0-1 CSI program)

$$\begin{aligned}
 &\text{Maximize } \sum_{l=1}^{18} \text{Score}_l = \sum_{i=1}^{10} (Z(a_i) + Z(t_i) + Z(c_i) + Z(g_i)) \\
 &\text{subject to } \sum_{s=1}^{90} z_{l,s} = 1, \quad z_{l,s} \geq 0 \quad \text{for all } l, s \\
 &\left. \begin{aligned} a_i &= 1 - u_i - v_i + w_i \\ t_i &= u_i \\ c_i &= v_i - w_i \\ g_i &= u_i - w_i \\ w_i &\leq u_i \\ w_i &\leq v_i \\ w_i &\geq 0 \\ w_i &\geq u_i + v_i - 1 \end{aligned} \right\} \begin{aligned} &\text{Conservative constraints} \\ &\text{for } i = 1, 2, \dots, 10 \end{aligned} \\
 &\left. \begin{aligned} u_i + u_{11-i} &= 1 \\ v_i + v_{11-i} &= 1 \end{aligned} \right\} \begin{aligned} &\text{Logical constraints for } i = 1, 2, \dots, 5 \end{aligned}
 \end{aligned} \tag{15}$$

$$\left. \begin{aligned} &\sum_{(l,s) \in SA_i} z_{l,s} + M(a_i - 1) \leq Z(a_i) \leq \sum_{(l,s) \in SA_i} z_{l,s} \\ &0 \leq Z(a_i) \leq Ma_i \\ &\sum_{(l,s) \in ST_i} z_{l,s} + M(t_i - 1) \leq Z(t_i) \leq \sum_{(l,s) \in ST_i} z_{l,s} \\ &0 \leq Z(t_i) \leq Mt_i \\ &\sum_{(l,s) \in SC_i} z_{l,s} + M(c_i - 1) \leq Z(c_i) \leq \sum_{(l,s) \in SC_i} z_{l,s} \\ &0 \leq Z(c_i) \leq Mc_i \\ &\sum_{(l,s) \in SG_i} z_{l,s} + M(g_i - 1) \leq Z(g_i) \leq \sum_{(l,s) \in SG_i} z_{l,s} \\ &0 \leq Z(g_i) \leq Mg_i \\ &u_i, v_i \in \{0, 1\} \quad \text{for } i = 1, 2, \dots, 5 \\ &0 \leq u_i, v_i \leq 1 \quad \text{for } i = 6, 7, \dots, 10 \\ &0 \leq a_i, t_i, c_i, g_i \leq 1 \quad \text{for } i = 1, 2, \dots, 10. \end{aligned} \right\} \begin{aligned} &\text{Constraints for} \\ &\text{linearizing} \\ &\text{product terms} \end{aligned}$$

$z_{l,s}$ 's are treated as non-negative continuous variables for  $l = 1, 2, \dots, 18$  and  $s = 1, 2, \dots, 90$  where  $M$  can be any value greater than or equal to 18.

In Program 2, since  $u_i$  and  $v_i$  are binary variables,  $a_i, t_i, c_i$  and  $g_i$  should have binary values following Equation (3). Although  $z_{l,s}$  are treated as continuous variables, the values of  $z_{l,s}$  should be 0 or 1. This is because the optimal solution of a linear program should be a vertex point satisfying  $\sum_s z_{l,s} = 1$  for all  $l$ .

Consider the following proposition.

PROPOSITION 4. Let the optimal solution of Program 2 be  $x^* = (Z^*, u^*, v^*)$  and  $\sum_s z_{l,s} = 1$ . Assume that a sequence  $l$  contains sites  $s_1, s_2, \dots, s_k$  such that  $0 < z_{l,s_j}^* < 1$  for  $j = 1, 2, \dots, k$ , then,

$$\sum_i \theta_{l,s_1}^i = \sum_i \theta_{l,s_2}^i = \dots = \sum_i \theta_{l,s_k}^i = \max \left\{ \sum_i \theta_{l,s}^i \right\},$$

where  $\theta_{l,s_j}^i$  are specified in Equation (6).

PROOF. For  $\sum_s z_{l,s} = 1$ , if  $s_p, s_q \in \{s_1, s_2, \dots, s_k\}$  where  $\sum_i \theta_{l,s_p}^i > \sum_i \theta_{l,s_q}^i$ , then to maximize  $\text{Score}_l = \sum_{i,j} z_{l,s_j} \sum_i \theta_{l,s_j}^i$  requires  $z_{l,s_q} = 0$ . This conflicts with the observation that  $0 < z_{l,s_q} < 1$ , therefore  $\sum_i \theta_{l,s_1}^i = \sum_i \theta_{l,s_2}^i = \dots = \sum_i \theta_{l,s_k}^i$ .

After solving Program 2 we can obtain the globally optimum solution TGTGA□□□□□TCACA with objective value 147. The related non-zero  $z_{l,s}$  values indicate the starting positions of the binding sites in the 18 sequences, as listed below:

$$\begin{aligned}
 z_{1,64} &= z_{2,58} = z_{3,79} = z_{4,66} = z_{5,53} = z_{6,63} = z_{7,27} \\
 &= z_{8,42} = z_{9,12} = z_{10,17} = z_{11,64} = z_{12,44} = z_{13,51} \\
 &= z_{14,74} = z_{15,20} = z_{16,56} = z_{17,87} = z_{18,81} = 1.
 \end{aligned}$$

All other  $z_{l,s}$ s have value 0.

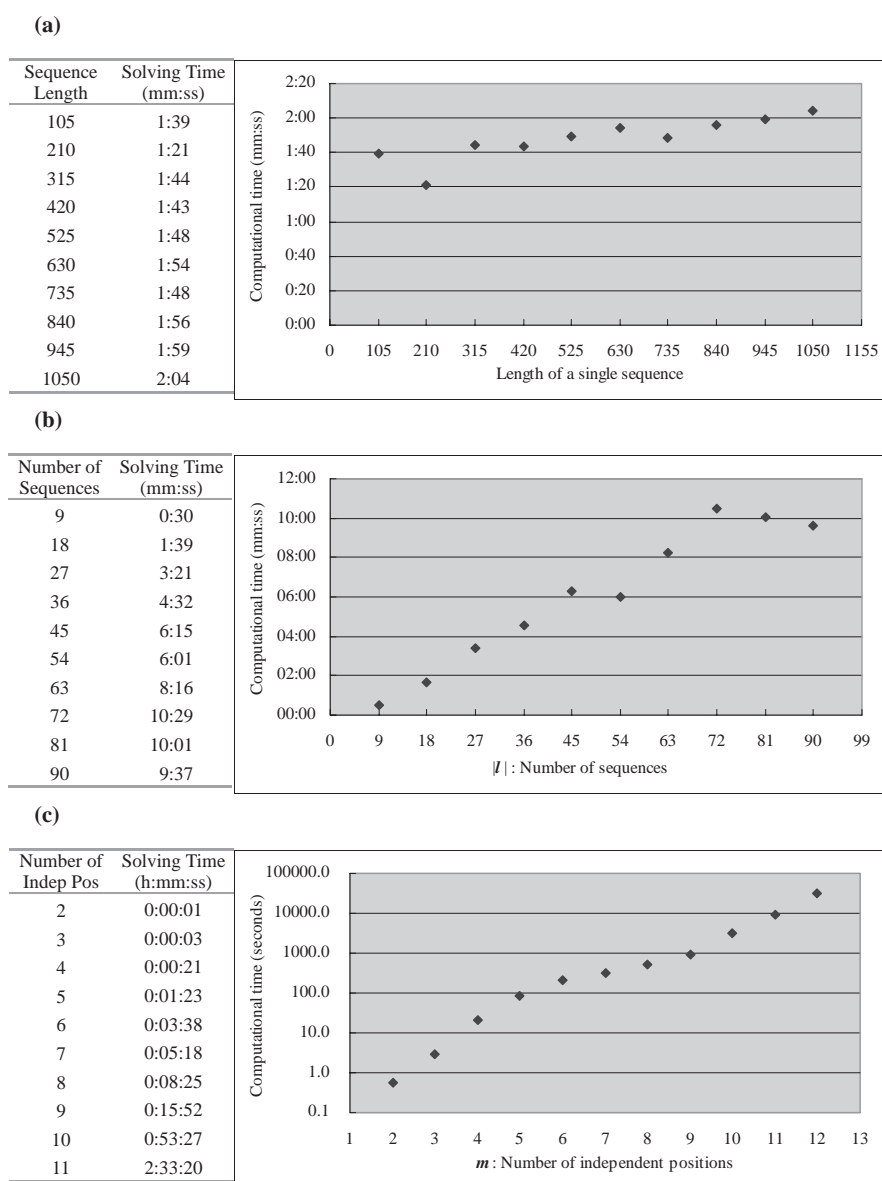
In Program 2 the total number of 0-1 variables is  $2m$  and the total number of the continuous variables is  $20m + |l| * |s|$ . Since the number of 0-1 variables is independent of the lengths of  $l$  and  $s$ , a CSI problem with many long sequences can be solved effectively.

### Suboptimal common sites

Program 2 can find the exact global optimum solution. Sometimes the second best and the third best solutions may also be useful. It is very convenient for the proposed method to find a complete set of common sites by adding some extra constraints. For instance, the second best solution of Program 2 can be obtained conveniently by solving the following program:

$$\begin{aligned}
 &\text{Maximize } \sum_{l=1}^{18} \text{Score}_l \\
 &\text{subject to } \begin{aligned} &\text{(i) The same constraints in Model 1} \\ &\text{(ii) } t_1 + g_2 + t_3 + g_4 + a_5 + t_6 + c_7 + a_8 + c_9 + a_{10} \leq 9 \\ &\quad \text{(new constraint).} \end{aligned}
 \end{aligned} \tag{16}$$

The new constraint is used to force the program to find a new solution different from the solution of Program 2. The second best common site found



**Fig. 2.** The relationship between computational time and various factors involved in a CSI problem. This figure illustrates the computational time of solving Program 2 with (a) various sequences sizes, (b) various number of sequences and (c) various independent positions.

is TTTGA□□□□□TCAAA with score 129. Similarly we can find another solution by adding following constraint into Equation (16).

$$t_1 + t_2 + t_3 + g_4 + a_5 + t_6 + c_7 + a_8 + a_9 + a_{10} \leq 9.$$

The third best common site found is AAATT□□□□□AATTT with score 129.

## IMPLEMENTATION

Several experiments are tested here, using the example in the Appendix section, to analyze the effect of sequence length and number of sequences on the computational time. All examples are solved by LINGO (Schrage, 1999), a widely used optimization software, on a personal computer with a Pentium 4 2.0G CPU. A software package named Global Site Seer is developed, based on

Program 2 for solving CSI problems. This software is available from <http://www.iim.nctu.edu.tw/~cjfu/gss.htm>

Figure 2 illustrates the experimental results for analyzing the time complexity. Figure 2a is the computational time given various sequence lengths, where the number of sequences is fixed at 18. The results show that the computational time changes slightly even if the sequence length is increased from 105 to 1050. Figure 2b is the computational time with various numbers of sequences. It shows that the solving time is roughly proportional to the number of sequences. The proposed model is quite promising for treating CSI problems with large sequence length and a large number of sequence numbers. Figure 2c shows that the computational time rises exponentially as the number of independent positions increases.



### Time complexity and distributed computing

From the results of Figure 2 we know that the time complexity is roughly proportional to the number of sequences and is influenced slightly by the length of the sequences. However, the computational time rises exponentially as the number of independent positions increases. The worst case of time complexity of solving Program 2 on a single machine is estimated as  $O(|I|2^{2m})$ , where  $|I|$  is the number of sequences and  $m$  is the number of independent positions.

To treat CSI problems with more independent positions, a method of distributed computing is discussed in this section. Suppose there are  $n$  PCs available for solving Program 2, we can decompose Program 2 into  $n$  subprograms by specifying different values on some  $u_i$ s and  $v_i$ s. For instance, if  $n = 32$  then the first subprogram can be formulated as

#### Subprogram 1

$$\text{Maximize } f(x) = \sum_l \text{Score}_l \quad (17)$$

subject to (i) The same constraint sets as in Program 2  
(ii)  $u_1 = v_1 = u_2 = v_2 = u_3 = 0$  (new constraint).

The new constraint (ii) is used to reduce the number of 0-1 variables from 10 to 5. Similarly, constraint (ii) for the second subprogram can be set as  $u_1 = 1$  and  $v_1 = u_2 = v_2 = u_3 = 0$ . Constraint (ii) for the 32nd subprogram could be  $u_1 = v_1 = u_2 = v_2 = u_3 = 1$ . All these 32 subprograms are solved simultaneously. Such a distributed computation algorithm can enhance the computational efficiency greatly. The computational time of Program 2 can be estimated as follows:

$$\text{Time}(l, m, n) = \alpha |l| 2^{\beta(2m - \lfloor \log_2 n \rfloor)} \quad (18)$$

where  $\alpha$  and  $\beta$  are parameters,  $m$  is the number of independent positions,  $n$  is the number of available PCs.

Figure 3 is the results of some experiments for solving Problem 2 with various  $m$  and  $n$  while  $|I| = 18$ . For the example of finding CRP-binding sites, the estimated  $\alpha$  and  $\beta$  values are  $\alpha = 0.014$  and  $\beta = 0.621$ .

### EXTEND TO FIND UNKNOWN BINDING SITES

A more complicated CSI problem is to search for the common site in an uncertain pattern format where the number of ignored letters between the two half sites is unknown. An example is to find a common site of length  $2 * 5 + k$  with the pattern

$$L_1 L_2 L_3 L_4 L_5 \square \cdots \square L_6 L_7 L_8 L_9 L_{10}$$

where  $k$ , the number of  $\square$ s, is an unknown integer between 0 and 10.

Program 2 can be modified slightly to treat this type of extended CSI problems. First we expand D in (1) as  $D'$  below:

$$D' = [D'(0) D'(1) D'(2) \cdots D'(10)]$$

in which

$$D'(k) = \begin{bmatrix} d_{1,1,k}^1 & \cdots & d_{1,1,k}^{10} & d_{1,2,k}^1 & \cdots & d_{1,2,k}^{10} & \cdots & d_{1,90,k}^1 & \cdots & d_{1,90,k}^{10} \\ d_{2,1,k}^1 & \cdots & d_{2,1,k}^{10} & d_{2,2,k}^1 & \cdots & d_{2,2,k}^{10} & \cdots & d_{2,90,k}^1 & \cdots & d_{2,90,k}^{10} \\ \vdots & & \vdots & & \vdots & & \ddots & & \vdots & \\ d_{18,1,k}^1 & \cdots & d_{18,1,k}^{10} & d_{18,2,k}^1 & \cdots & d_{18,2,k}^{10} & \cdots & d_{18,90,k}^1 & \cdots & d_{18,90,k}^{10} \end{bmatrix},$$

Computational time	$n$					
$m$	1	2	4	8	16	32
3	0:00:03	0:00:01	0:00:01	0:00:01	0:00:01	0:00:01
4	0:00:21	0:00:22	0:00:07	0:00:12	0:00:08	0:00:11
5	0:01:23	0:01:20	0:00:57	0:00:25	0:00:18	0:00:17
6	0:03:38	0:02:34	0:01:13	0:00:34	0:00:33	0:00:27
7	0:05:18	0:02:50	0:01:53	0:01:15	0:01:28	0:01:05
8	0:08:25	0:05:24	0:05:08	0:04:12	0:04:10	0:01:42
9	0:15:52	0:09:40	0:07:20	0:06:45	0:04:30	0:03:31
10	0:53:27	0:35:32	0:24:21	0:18:42	0:09:44	0:06:40
11	2:33:20	1:33:44	1:10:25	0:52:35	0:28:15	0:19:01
12			3:08:04	2:07:53	1:17:32	0:40:54
13					7:12:31	2:44:19

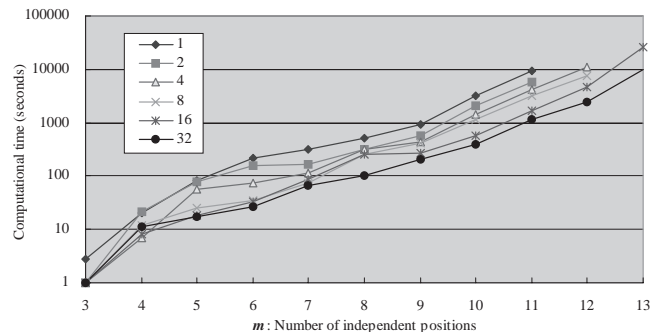


Fig. 3. Computational time of distributed computing with various  $m$  (independent positions) and  $n$  (number of available PCs).

where  $k \in \{0, 1, \dots, 10\}$ .

$$d_{l,s,k}^i = \begin{cases} b_{l,i+s-1} & (\text{for } i = 1, 2, 3, 4, 5) \\ b_{l,i+s+k-1} & (\text{for } i = 6, 7, 8, 9, 10) \end{cases}$$

$$\theta_{l,s,k}^i = a_i, t_i, c_i \text{ or } g_i \quad \text{when } d_{l,s,k}^i = A, T, C \text{ or } G.$$

The cases with  $k > 10$  are not considered since they are relatively rare. A linear mixed 0-1 program for solving this example is formulated below:

#### Program 3

$$\text{Maximize } \sum_{i=1}^{2m} (Z(a_i) + Z(t_i) + Z(c_i) + Z(g_i)) \quad (19)$$

$$\text{subject to (i) } \sum_{k=0}^{10} \sum_{s=1}^{96-k} z_{l,s,k} = 1, \quad z_{l,s,k} \geq 0 \quad \text{for all } l, s, k$$

$$\text{(ii) } \sum_s z_{1,s,k} = \sum_s z_{2,s,k} = \cdots = \sum_s z_{18,s,k} \\ \text{for } k \in \{0, 1, \dots, 10\}$$

(iii) the same conservative and logical constraints in Program 2

(iv) the same constraints for linearizing product terms in Program 2 but replace  $z_{l,s}$  by  $z_{l,s,k}$ .

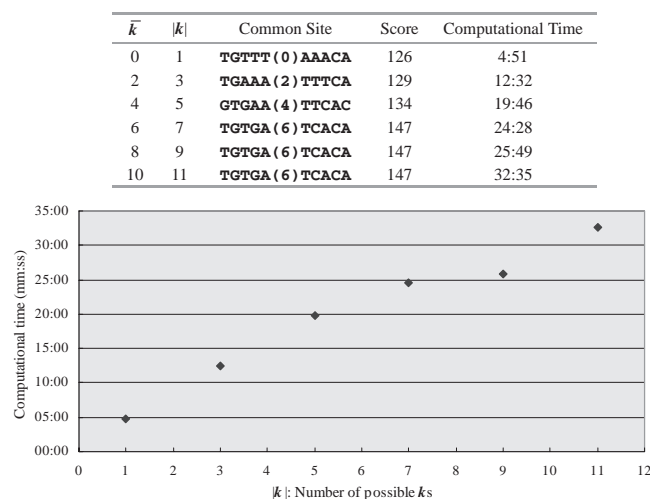
Constraints (i) and (ii) are used to ensure that when a specific  $k$  is chosen  $\sum_s z_{l,s,k} = 1$  and  $\sum_s z_{l,s,k'} = 0$  for  $k' \neq k$ .

**Table 2.** FNR binding sites found by Program 3

Operon	Sequence length	Site sequence found by Program 3	Predicted position	Score	Site sequence listed in RegulonDB <sup>a</sup>	Center position
<b>Common site:</b>		<b>TTGAT—ATCAA</b>				
narK	338	ATGAT—ATCAA	−86	9	actatgGGTA[ATGAT]AAAT[ATCAA]TGATagataa	−79.5
		TTGAT—ATCAA	−48	10	atcttaTCGT[TTGAT]TTAC[ATCAA]ATTGccttta	−41.5
ansB	345	TTGTT—GTCAA	−48	8	acgttgTAAA[TTGTT]TAAC[GTCAA]ATTTcccata	−41.5
		TTGTA—TCCAA	−81	6	gcctctAACT[TTGTA]GATC[TCCAA]AATAtattca	−74.5
		TTTAT—TTTAA	−123	7		
narG	525	TTGAT—ATCAA	−55	10	ctc[ttgAT]CGTT[ATCAA]TTCCACGCTGtttcag	−41.5
dmsA	325	TTGAT—AACAA	−48	9	ct[ttgaT]ACCG[AACAA]TAATTACTCCTCacttac	−33
fird	781	TTCAG—ATCCA	−37	7	AAAAATCGATCTCGTCAAAT[TTcag]act[atcca]	−47
		TTAAT—TTCAG	−98	7		
nirB	262	TTGAT—ATCAA	−48	10	aaaggtGAAT[TTGAT]TTAC[ATCAA]TAAGcggggt	−41.5
sodA	284	TTGAT—ATTTT	−42	7	agtacGCA[TTGAT]AATC[ATTTT]CAATatcatt	−34
fmr <sup>b</sup>	96	TTGAC—ATCAA	−7	9	atgttaAAA[TTGAC]AAAT[ATCAA]TTACGgcttga	1
					ccttaaCAACTTAAGGGTTTTCAAATAGatagac	−103.5
(cyoA)	599	CTTCT—ATCAA	−113	7	N/A	N/A
		TTGTT—TTCAC	−198	7		
(icdA)	290	ATGAC—AACAA	16	7	N/A	N/A
		TTGCT—AGCAT	73	7		
(sdhC)	708	TTGAT—AATAA	−330	8	N/A	N/A
(ulaA)	346	TCAAT—ATCAA	−278	8	N/A	N/A
		TTGGT—ATTAA	−257	8		

<sup>a</sup>For visualizing the comparison, the letters in uppercase represent the binding site listed in RegulonDB; the letter in bold face is the center of the site sequence; and the encompassed letters represent the exact binding site obtained by Program 3.

<sup>b</sup>The second site listed in RegulonDB is not contained in the sequence data, which is only 96 bases long, from GenBank.

**Fig. 4.** Computational time of Program 3 with various numbers of possible  $k$ 's. The number enclosed in the common site is the solution of  $k$ .

Using Program 3 to search CRP binding sites we obtain the globally optimal solution as TGTGA□□□□TCACA with score 147, which is exactly the solution found in Program 2. And the second best solution is GTGAA□□□□TTCAC with score 134. The relationship between the computational time and the number of possible  $k$ 's (i.e.  $|k|$ ) is linear, as shown in the experiment result listed in Figure 4.

The number of ignored letter  $k$  is between 0 and  $\bar{k}$ , the upper bound of  $k$ , and thus we have  $|k| = \bar{k} + 1$  in this experiment.

### Finding FNR-binding sites

Program 3 is also applied to solve an example of searching for binding sites of fumarate and nitrate reduction regulatory protein (FNR) in *E.coli*. Both CRP and FNR belong to the CRP/FNR helix-turn-helix transcription factor superfamily (Tan *et al.*, 2001). The sequence data, which is taken from GenBank, contains 12 DNA sequences with lengths varying from 96 to 781. Owing to the dimer structure of the binding protein, the common site in this example also has a constraint of inverse symmetry. The RegulonDB database (Huerta *et al.*, 1998) lists the regulatory binding sites found for 8 of these 12 sequences while the exact positions of the other 4 sequences are not listed yet. Solving this example by Program 3 we obtained the global optimal common site as TTGAT□□□□ATCAA with score 107, which is the same common site as indicated by Tan *et al.* (2001). Table 2 illustrates the result including the common site and the predicted binding sites for all of the 12 sequences. Some sites downstream of the transcription start (i.e. with positive indices) are also listed because there are a few known cases in which regulatory sites appear within transcription units (Tan *et al.*, 2001). The proposed method has found some sites not listed in RegulonDB but having scores higher than those listed in RegulonDB (e.g. the third solution in the Operon *ansB* row of Table 2). The best predicted sites in the four undetermined sequences are also listed in Table 2.

## DISCUSSION

This study proposes a linear mixed 0-1 programming approach for solving CSI problems. Compared to the widely used maximum-likelihood methods, the proposed method can reach a global optimum rather than finding a local optimum or a feasible solution. Additionally, by utilizing binary variables some logical constraints can be embedded into the models. It is also convenient to find the complete set of the second, third, etc. best common sites. Since the number of binary variables is fully independent of the number of sequences and the length of a sequence, the proposed method can treat a large CSI problem with many long sequences. For treating a CSI problem with many independent positions in an acceptable time, this study also proposes a method for distributed computing.

The proposed method can also be conveniently extended to treat more complicated CSI problems. In this study an extension of the linear program is designed to solve CSI problems with an unknown number of ignored letters between the two half sites. The results of searching for FNR-binding sites show that the extended model can not only find the locations of known binding sites listed in the RegulonDB database but also those not yet delimited.

There are two issues remaining for further study. The first is to extend this method to treat various practical CSI problems. The second is to develop a more refined distributed algorithm to solve some CSI problems by numerous computers via the internet.

## ACKNOWLEDGEMENT

The authors express deep appreciation for the referees' valuable comments on this paper. On the basis of these comments, this paper has been substantially improved.

## REFERENCES

- Balas,E. *et al.* (1996) Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Manage Sci.*, **42**, 1229–1246.
- Brazma,A. *et al.* (1998) Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.*, **5**, 279–305.
- Ecker,J.G. *et al.* (2002) An application of nonlinear optimization in molecular biology. *Eur. J. Oper. Res.*, **138**, 452–458.
- Hertz,G.Z. *et al.* (1990) Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.*, **6**, 81–92.
- Huerta,A.M. *et al.* (1998) RegulonDB: a database on transcriptional regulation in *Escherichia coli*. *Nucl. Acids Res.*, **26**, 55–59.
- Lawrence,C.E. and Reilly,A.A. (1990) An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, **7**, 41–51.
- Schrage,L. (1999) *Optimization Modeling with Lingo*, LINDO Systems Inc., Chicago, IL.
- Stormo,G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, **16**, 16–23.
- Stormo,G.D. and Hartzell,G.W. (1989) Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl Acad. Sci. USA*, **86**, 1183–1187.
- Tan,K. *et al.* (2001) A comparative genomics approach to prediction of new members of regulons. *Genome Res.*, **11**, 566–584.