行政院國家科學委員會專題研究計畫　期中進度報告

以全域最佳化方法求解 DNA 序列之共同區間定址問題(2/3)

計畫類別：個別型計畫
計畫編號：NSC94-2213-E-009-040-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：國立交通大學資訊管理研究所

計畫主持人：黎漢林

計畫參與人員：傅昶瑞

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 95 年 5 月 22 日

行政院國家科學委員會專題研究計畫期中進度報告

以全域最佳化方法求解 DNA 序列之共同區間定址問題 (2/3)

A Linear Programming Approach for Identifying a Consensus Sequence on DNA Sequences (2/3)

計畫編號：NSC 94-2213-E-009-040

執行期限：94 年 8 月 1 日至 95 年 7 月 31 日

主持人：黎漢林　　國立交通大學資訊管理研究所

本研究為三年期計畫，目的在於發展全域最佳化技術來解決基因序列共同區間的尋找問題。在多組基因序列間尋找共同區間的問題，在生物資訊的分析上是相當常見的。部分案例中這個共同區間還需要包含結構性的限制，mRNA 中 stem-loop motifs 的尋找即為其中一例。

在本計畫中這類問題首先將被轉換成非線性 0-1 的數學模式。該模式經過轉換後可以變成包含有限個 0-1 變數的線性模式，且可利用分散式計算來快速求解。此方法保證可以找到全域最佳解，其運算速度也遠優於任何現今被提出的方法。

第一年本計畫將進行該方法的發展及驗證，第二年與第三年針對各種在生物資訊分析方面可能的應用來延伸開發其應用的作法，且發展自動化的應用程式系統來完成分散運算的能力。

關鍵詞：最佳化、生物資訊、分子生物學、蛋白質鍵結

Abstract

This plan is about to develop a global optimization method in identifying a set of protein binding sites in a set of unaligned DNA fragments. The identification of common sites in multiple sequences is frequently encountered in the analysis of biopolymer sequence data. In several cases there exists various kind of structural constraints in such a problem. An example is the stem-loop motifs in mRNA molecules.

Firstly the problem is formulated as a nonlinear 0-1 optimization model. This model is then converted into a linear 0-1 problem solvable by distributed computation system. The technique is guaranteed to find a global optimum. In additional, the computational speed of this technique is much faster than current methods in literature.

In first year we develop this methodology and verify the performance. And in the following two years we will apply this method to various types of practical use. Meanwhile a distributed computation system will developed to enhance the computation.

Keywords: Optimization, Bioinformatics, Molecular biology, Protein binding

1. 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 site are, discover 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 those methods, the maximum likelihood approach (Stormo *et al*., 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 is reliable on discovering 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 nonlinear 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 firstly formulated as a nonlinear 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 nonlinear 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 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 firstly that 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. Secondly, the proposed method is guaranteed to find the global optimum instead of a local optimum.

Thirdly, many kinds of specific features accompanied with a CSI problem can be formulated straight forwardly as logical constraints and embedded into the linear program.

An example of searching CRP-binding sites, as discussed in Stormo *et al*. (Stormo *et al*., 1989) and Ecker *et al*. (Ecker *et al*., 2002), is described as follows. Given eighteen letter sequences each 105 positions long, where each position contains a letter from the set {A, T, C, G}, find a common site of length16 with the pattern

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

where L_i , $\square$ $\in$ {A, T, C, G}and $\square$'s mean the positions of ignored letters.

Restated, the problem is to specify

(i) the L_i 's of the common site pattern

(ii) 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:

(i) Only a local optimal or feasible solution is obtained

Since Ecker *et al*. (2002) formulated a CSI problem as a non-convex nonlinear program, their method may only find local optima, as has been acknowledged (Ecker *et al*., 2002). 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.

(ii) Heavy computational burden

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

(iii) 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 complement (i.e. G with C and A with T). Formulating such a constraint in maximum likelihood approaches is a complex task. It is even impossible to formulate more complicated logical constraints (e.g. those with some ambiguity) when applying these approaches.

(iv) 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 real world this number is unknown and need to be found by some preliminary processes.

(v) 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:

(i) 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.

(ii) It can effectively solve a CSI problem by a set of on-line computers as illustrated by our numerical experiments.

(iii) It is convenient to add logical constraints. Since the binary variables are very suitable to express logical relationship, various complicated constraints can be embedded directly into the proposed method.

(iv) It can be extended to treat CSI problems with unknown number of ignored letters.

(v) It is very straight forward 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.

2. Proposed Method

This study firstly formulates a CSI problem as a nonlinear mixed 0-1 program. Then it converts this nonlinear 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 improves greatly the computational efficiency of this program.

Nonlinear mixed 0-1 program

Here we use the example data in Stormo (1989), as listed in Appendix, to describe the proposed method. Firstly, represent the data in Appendix as an 18*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 previous section, the common site we want to find has 16 positions (ten L_i's and six ignored letters), a sequence has 90 corresponding sites, so an 18*900 data matrix *D'* is generated from *D*.

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

where

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

and *s* = 1…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 Tab. 1.

Tab. 1 indicates that if L_i is A, T, C, or G respectively, then a_i = 1, t_i = 1, c_i = 1 or g_i = 1, which implies following conditions.

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

Now let $Score_l$ be the degree of fitting to the found common site, specified as

$$Score_l = \sum_{s=1}^{90} z_{l,s}(\theta_{l,s}^1 + \theta_{l,s}^2 + ... + \theta_{l,s}^{10}) \tag{4}$$

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

(a)

AAGACTGTTTTTTTGATC
GATTATTTGCACGGCGTC

(b)

$l = 1, s = 1$ AAGACTGTTTTTTTGATC
$l = 1, s = 2$ AAGACTGTTTTTTTGATC
$l = 1, s = 3$ AAGACTGTTTTTTTGATC
$l = 2, s = 1$ GATTATTTGCACGGCGTC
$l = 2, s = 2$ GATTATTTGCACGGCGTC
$l = 2, s = 3$ GATTATTTGCACGGCGTC

(c)

$$\begin{bmatrix} AAGACTTTGA & AGACTTTGAT & GACTGTGATC \\ GATTACGGCG & ATTATGGCGT & TTATTGCGTC \end{bmatrix}$$

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

(i) $$\sum_{s=1}^{90} z_{l,s} = 1, \quad z_{l,s} \in \{0,1\} \text{ for all } l \text{ and } s. \tag{5}$$

(ii) $$\theta_{l,s}^{i} = \begin{cases} a_i & if\ d_{l,s}^{i} = \mathrm{A} \\ t_i & if\ d_{l,s}^{i} = \mathrm{T} \\ c_i & if\ d_{l,s}^{i} = \mathrm{C} \\ g_i & if\ d_{l,s}^{i} = \mathrm{G} \end{cases} \tag{6}$$

Clearly, $0 \le Score_l \le 10$. And the objective is to maximize the total sum of $Score_l$.

Consider the sample data in Fig. 1 for instance:

$$\begin{aligned} Score_1 = \; & z_{1,1}(a_1 + a_2 + g_3 + a_4 + c_5 + t_6 + t_7 + t_8 + g_9 + a_{10}) \\ & + z_{1,2}(a_1 + g_2 + a_3 + c_4 + t_5 + t_6 + t_7 + g_8 + a_9 + t_{10}) \\ & + z_{1,3}(g_1 + a_2 + c_3 + t_4 + g_5 + t_6 + g_7 + a_8 + t_9 + c_{10}) \end{aligned} \tag{7}$$

Tab. 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

$$Score_2 \;=\; z_{2,1}(g_1 + a_2 + t_3 + t_4 + a_5 + c_6 + g_7 + g_8 + c_9 + g_{10}) \tag{8}$$

$$+ z_{2,2}(a_1 + t_2 + t_3 + a_4 + t_5 + g_6 + g_7 + c_8 + g_9 + t_{10})$$

$$+ z_{2,3}(t_1 + t_2 + a_3 + t_4 + t_5 + g_6 + c_7 + g_8 + t_9 + c_{10})$$

All $z_{l,s}$ in (4) are binary variables. Equation (5) implies that for a sequence *l*, only one site is chosen and no other sites contribute to $Score_l$. Suppose the *k'*th site is selected, then $z_{l,k} = 1$ and $z_{l,s} = 0$ for all $s \in \{1, 2, ..., 90\}$, $s \neq k$. Since a huge amount of $z_{l,s}$ (i.e, $|l| * |s|$) 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:

<u>Proposition 1</u> (All or nothing assignment) Let $z_{l,s} \geq 0$ be continuous variables instead of binary variables. If there is a *k*, $k \in \{1, 2, ..., 90\}$, such that $\sum_{i=1}^{10} \theta_{l,k}^i = \max\{\sum_{i=1}^{10} \theta_{l,s}^i \; for\; s = 1,2,...,90\}$, then assigning $z_{l,k} = 1$ and $z_{l,s} = 0$ for all $s \neq k$, $s \in \{1,2,...,90\}$, can maximize the value of $Score_l$.

<u>Proof</u> Since $\sum_s z_{l,s} = 1$ and $z_{l,s} \geq 0$, it is true that

$$\max \{ \textstyle\sum_s (z_{l,s} \sum_i \theta_{l,s}^i) \} \leq \max \{ \sum_i \theta_{l,s}^i \text{ for } s = 1,2,...,90 \} = \sum_i \theta_{l,k}^i$$

<u>Remark 1</u> The objective function of a CSI problem *f*(x) can be rewritten as

$$f(x) = \sum_{i=1}^{10} \{a_i \sum_{(l,s)\in SA_i} z_{l,s} + t_i \sum_{(l,s)\in ST_i} z_{l,s} + c_i \sum_{(l,s)\in SC_i} z_{l,s} + g_i \sum_{(l,s)\in SG_i} z_{l,s}\} \tag{9}$$

where $SA_i = \{(l,s) \mid d_{l,s}^i = A\}$, $ST_i = \{(l,s) \mid d_{l,s}^i = T\}$, $SC_i = \{(l,s) \mid d_{l,s}^i = C\}$, and $SG_i = \{(l,s) \mid d_{l,s}^i = G\}$ for *i*=1,2,…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 $z_{l,s}t_i$, $z_{l,s}c_i$, $z_{l,s}g_i$ respectively) appears on the right hand side of (4) because that $\theta^{i}_{l,s} = a_i$.

For instance, the sum of $Score_1$ and $Score_2$ in (7) and (8) becomes

$$
\begin{aligned}
Score_1 + Score_2 \;=\; & a_1(z_{1,1} + z_{1,2} + z_{2,2}) + \ldots\ldots + a_{10} z_{1,1} \\
& + \ldots\ldots + g_1(z_{1,3} + z_{2,1}) + \ldots\ldots + g_{10} z_{2,1}
\end{aligned}
\tag{10}
$$

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

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

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

$$
\left.\begin{aligned} u_i + u_{11-i} &= 1 \\ v_i + v_{11-i} &= 1 \end{aligned}\right\} \text{for } i = 1, 2, 3, 4, 5 \tag{11}
$$
$$
\text{where } u_i, v_i, u_{11-i}, v_{11-i} \in \{0, 1\}.
$$

With reference to Tab. 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; (ii) 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 nonlinear mixed 0-1 program below based on these constraints:

<u>Program 1 (Nonlinear 0-1 CSI program)</u>

$$
\text{Maximize} \quad \sum_{l=1}^{18} Score_l = \sum_{i=1}^{10} \{ a_i \sum_{(l,s)\in SA_i} z_{l,s} + t_i \sum_{(l,s)\in ST_i} z_{l,s} + c_i \sum_{(l,s)\in SC_i} z_{l,s} + g_i \sum_{(l,s)\in SG_i} z_{l,s} \} \tag{12}
$$

subject to

$$
\sum_{s=1}^{90} z_{l,s} = 1, \quad z_{l,s} \geq 0 \text{ for all } l, s
$$

$$
\left.\begin{aligned} a_i &= (1-u_i)(1-v_i) \\ t_i &= u_i v_i \\ c_i &= (1-u_i) v_i \\ g_i &= u_i (1-v_i) \end{aligned}\right\} \begin{aligned} &\text{Conservative constraints} \\ &\quad \text{for } i = 1, 2, \ldots, 10 \end{aligned}
$$

$$
\left.\begin{aligned} u_i + u_{11-i} &= 1 \\ v_i + v_{11-i} &= 1 \end{aligned}\right\} \begin{aligned} &\text{Logical constraints} \\ &\quad \text{for } i = 1, 2, \ldots, 5 \end{aligned}
$$

$$
u_i, v_i \in \{0, 1\} \quad \text{for } i = 1, 2, \ldots, 5
$$

$$
0 \leq u_i, v_i \leq 1 \quad \text{for } i = 6, 7, \ldots, 10
$$

$$
0 \leq a_i, t_i, c_i, g_i \leq 1 \quad \text{for } i = 1, 2, \ldots, 10
$$

This program intends to solve { a_i, t_i, c_i, g_i } for i =1,2, …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 $z_{l,s}$ still have binary values as discussed in the next section.

Linearization of Program 1

Program 1 is a mixed nonlinear 0-1 program where $q_i \sum z_{l,s}$ for $q_i \in \{a_i, t_i, g_i, c_i\}$ 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 λ_i is to be maximized and $q_i \in \{0,1\}$ can be linearized as follows:

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

where M is a big constant larger than or equal to 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 below based on Proposition 2 and Proposition 3.

Program 2 (Linear mixed 0-1 CSI program)

$$\text{Maximize} \quad \sum_{l=1}^{18} Score_l = \sum_{i=1}^{10} (Z(a_i) + Z(t_i) + Z(c_i) + Z(g_i)) \tag{15}$$

subject to $\sum_{s=1}^{90} z_{l,s} = 1, \qquad z_{l,s} \geq 0 \quad \text{for all } l, s$

$$\left.\begin{array}{l} a_i = 1 - u_i - v_i + w_i \\ t_i = w_i \\ c_i = v_i - w_i \\ g_i = u_i - w_i \\ w_i \leq u_i \\ w_i \leq v_i \\ w_i \geq 0 \\ w_i \geq u_i + v_i - 1 \end{array}\right\} \begin{array}{l} \text{Conservative constraints} \\ \text{for } i = 1, 2, ..., 10 \end{array}$$

$$\left.\begin{array}{l} u_i + u_{11-i} = 1 \\ v_i + v_{11-i} = 1 \end{array}\right\} \text{Logical constraints for } i = 1, 2, ..., 5$$

$$\left.\begin{array}{l} \sum_{(l,s)\in SA_i} z_{l,s} + M(a_i - 1) \leq Z(a_i) \leq \sum_{(l,s)\in SA_i} z_{l,s} \\ 0 \leq Z(a_i) \leq M\, a_i \\ \sum_{(l,s)\in ST_i} z_{l,s} + M(t_i - 1) \leq Z(t_i) \leq \sum_{(l,s)\in ST_i} z_{l,s} \\ 0 \leq Z(t_i) \leq M\, t_i \\ \sum_{(l,s)\in SC_i} z_{l,s} + M(c_i - 1) \leq Z(c_i) \leq \sum_{(l,s)\in SC_i} z_{l,s} \\ 0 \leq Z(c_i) \leq M\, c_i \\ \sum_{(l,s)\in SG_i} z_{l,s} + M(g_i - 1) \leq Z(g_i) \leq \sum_{(l,s)\in SG_i} z_{l,s} \\ 0 \leq Z(g_i) \leq M\, g_i \end{array}\right\} \begin{array}{l} \text{Constraints for linearizing} \\ \text{product terms} \end{array}$$

$u_i, v_i \in \{0, 1\} \quad \text{for } i = 1, 2, ..., 5$

$0 \leq u_i, v_i \leq 1 \quad \text{for } i = 6, 7, ..., 10$

$0 \leq a_i, t_i, c_i, g_i \leq 1 \quad \text{for } i = 1, 2, ..., 10$

$z_{l,s}$'s are treated as non-negative continuous variables for l =1,2, … ,18 and s =1,2, … ,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 (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 $s_1, s_2, ..., s_k$ such that $0 < z^*_{l,s_j} < 1$ for j=1, 2, … k, then,

$$\sum_i \theta^i_{l,s_1} = \sum_i \theta^i_{l,s_2} = ... = \sum_i \theta^i_{l,s_k} = \max\{\sum_i \theta^i_{l,s}\},$$

where θ^i_{l,s_j} are specified in (6).

Proof For $\sum_s z_{l,s} = 1$, if $s_p, s_q \in \{s_1, s_2, ...s_k\}$ where $\sum_i \theta^i_{l,s_p} > \sum_i \theta^i_{l,s_q}$, then to maximize $Score_l = \sum_{l,j} z_{l,s_j} \sum_i \theta^i_{l,s_j}$ requires $z_{l,s_q} = 0$. This conflicts with the observation that $0 < z_{l,s_q} < 1$, therefore $\sum_i \theta^i_{l,s_1} = \sum_i \theta^i_{l,s_2} = ... = \sum_i \theta^i_{l,s_k}$.

After solving Program 2 we can obtain the globally optimum solution "TGTGA□□□□□□TCACA" with objective value 147. The related nonzero $z_{l,s}$ 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 solution 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:

$$\text{Maximize} \quad \sum_{l=1}^{18} Score_l \tag{16}$$

subject to

(i) The same constraints in Model 1

(ii) $t_1 + g_2 + t_3 + g_4 + a_5 + t_6 + c_7 + a_8 + c_9 + a_{10} \leq 9$ (new constraint)

The new constraint is used to force the program to find a new solution different from the solution of Program 2. The found second best common site is “TTTGA□□□□□□TCAAA” with score 129. Similarly we can find another solution by adding following constraint into (16).

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

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

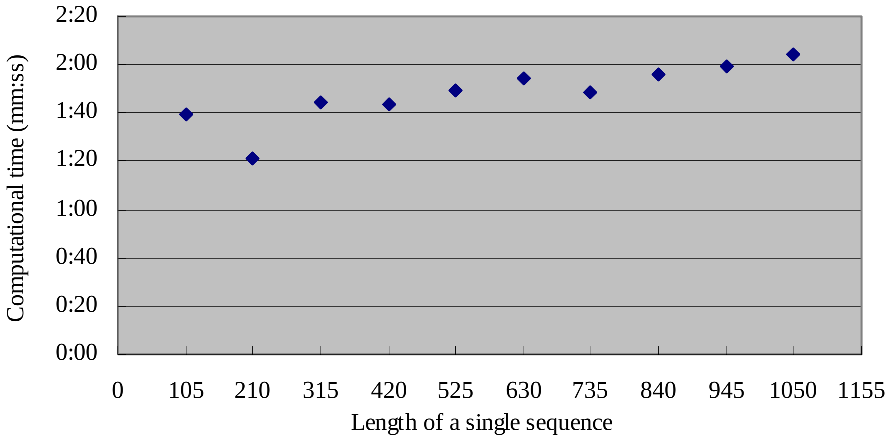
3. Implementation

Several experiments are tested here, using the example in the Appendix, 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.

Fig. 2 illustrates the experimental results for analyzing the time complexity. Fig. 2(a) 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. Fig. 2(b) 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 number. Fig. 2(c) shows that the computational time rises exponentially as the number of independent positions increases.

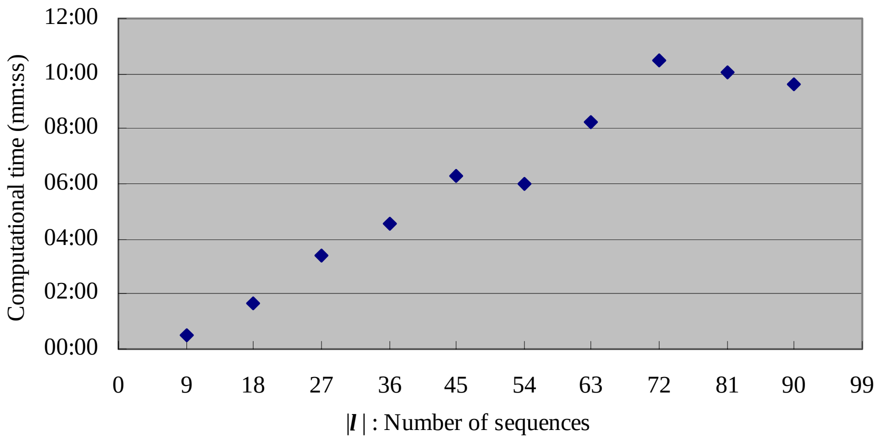
(a) Computational time versus sequence length

Sequence Length	Solving Time (mm:ss)
105	1:39
210	1:21
315	1:44
420	1:43
525	1:48
630	1:54
735	1:48
840	1:56
945	1:59
1050	2:04



(b) Computational time versus number of sequences

Number of Sequences	Solving Time (mm:ss)
9	0:30
18	1:39
27	3:21
36	4:32
45	6:15
54	6:01
63	8:16
72	10:29
81	10:01
90	9:37



(c) Computational time versus number of independent positions

Number of Indep Pos	Solving Time (h:mm:ss)
2	0:00:01
3	0:00:03
4	0:00:21
5	0:01:23
6	0:03:38
7	0:05:18
8	0:08:25
9	0:15:52
10	0:53:27
11	2:33:20

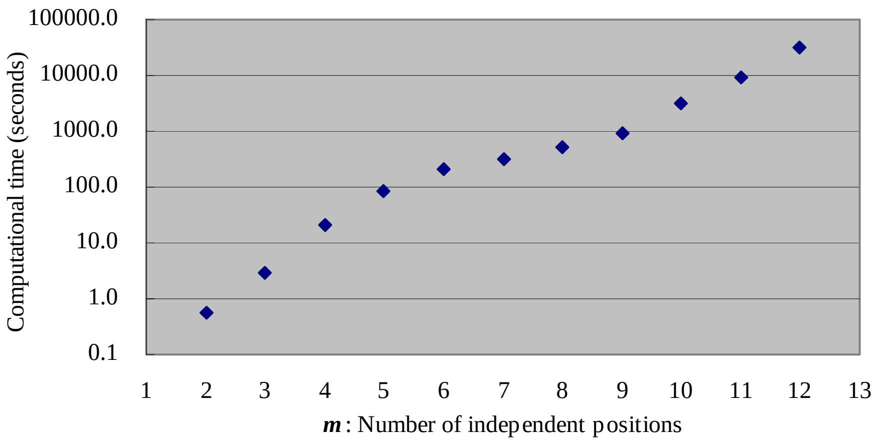


Fig. 2. The relationship between computational time and various factors involved in a CSI problem. This figure illustrates the computational time of solving Program 2 with (a) various sequences sizes; (b) various number of sequences and (c) various independent positions.

Time complexity and distributed computing

From the result of Fig. 2 we know that the time complexity is roughly proportional to the number of sequences and is influenced slightly by the length of 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($|l|\, 2^{2m}$), where $|l|$ 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

Maximize $$f(x) = \sum_l Score_l \tag{17}$$

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 32*th* 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:

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

where α and β are parameters, *m* is the number of independent positions, *n* is the number of available PCs.

Fig. 3 is the results of some experiments for solving Problem 2 with various *m* and *n* while $|l|$ = 18. For the example of finding CRP-binding sites, the estimated α and β values are α = 0.014 and β = 0.621.

4. Extend to Find Unknown Binding Sites

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

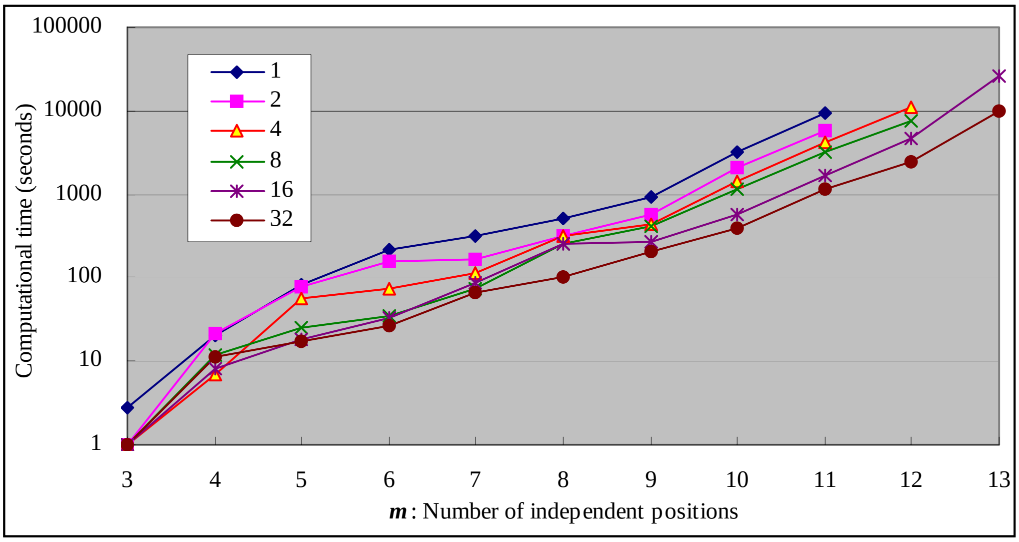


Fig. 3. Computational time of distributed computing with various ***m*** (independent positions) and ***n*** (number of available PCs)

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

$L_1L_2L_3L_4L_5$ □...... □ $L_6L_7L_8L_9L_{10}$

where k, the number of □'s, is an unknown integer between 0 and 10.

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

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

in which

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

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

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

$\theta^{i}_{l,s.k} = a_i$, t_i, c_i, or g_i when $d^{i}_{l,s.k}$ = ‘A’, ‘T’, ‘C’, or ‘G’ respectively.

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

<u>Program 3</u>

Maximize $\quad \sum_{i=1}^{2m} (Z(a_i) + Z(t_i) + Z(c_i) + Z(g_i))$ (15)

subject to

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

(ii) $\sum_{s} z_{1,s,k} = \sum_{s} z_{2,s,k} = ... = \sum_{s} z_{18,s,k}$ for $k \in \{0, 1, ..., 10\}$

(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 then $\sum_s z_{l,s,k} = 1$ and $\sum_s z_{l,s,k'} = 0$ for $k' \neq k$.

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 Fig. 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.

$\bar{k}$	$\|k\|$	Common Site	Score	Computational Time
0	1	**TGTTT(0)AAACA**	126	4:51
2	3	**TGAAA(2)TTTCA**	129	12:32
4	5	**GTGAA(4)TTCAC**	134	19:46
6	7	**TGTGA(6)TCACA**	147	24:28
8	9	**TGTGA(6)TCACA**	147	25:49
10	11	**TGTGA(6)TCACA**	147	32:35

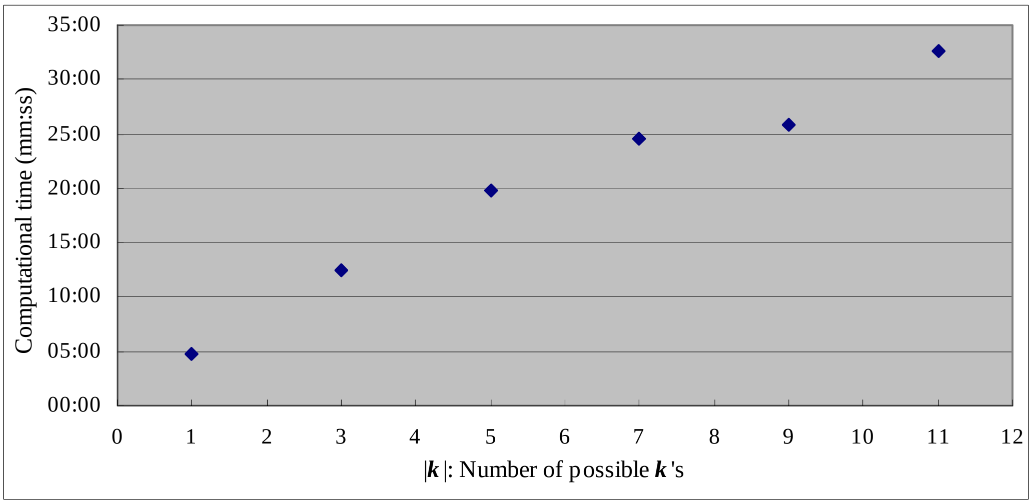


Fig. 4. Computational time of Program 3 with various numbers of possible *k*'s. The number enclosed in the common site is the solution of *k*.

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 varied 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 found regulatory binding sites for eight of these twelve sequences while the exact positions of other four 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). Tab.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 Tab.2). The best predicted sites in the four undetermined sequences are also listed in Tab.2.

5. Discussion

This study proposes a linear mixed 0-1 programming approach for solving CSI problems. Comparing with 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 result of searching for FNR-binding sites shows that the extended model can find not only the locations of known binding sites listed in the RegulonDB database but also those not yet

Tab. 2. FNR binding sites found by Program 3

Operon	Seq. length	Site seq. found by Program 3	Predicted Position	Score	Site seq. listed in RegulonDB*	Center Position
Common site:		**TTGAT----ATCAA**				
narK	338	ATGAT----ATCAA	-86	9	actatgGGTAATGATAA**A**TATCAATGATagataa	-79.5
		TTGAT----ATCAA	-48	10	atcttaTCGTTTGATTT**A**CATCAAATTGccttta	-41.5
ansB	345	TTGTT----GTCAA	-48	8	acgttgTAAATTGTTTA**A**CGTCAAATTTcccata	-41.5
		TTGTA----TCCAA	-81	6	gcctctAACTTTGTAGA**T**CTCCAAAATAtattca	-74.5
		TTTAT----TTTAA	-123	7		
narG	525	TTGAT----ATCAA	-55	10	ctcttgATCGTTATCAA**T**TCCCACGCTGtttcag	-41.5
dmsA	325	TTGAT----AACAA	-48	9	ctttgaTACCGAACAAT**A**ATTACTCCTCacttac	-33
frd	781	TTCAG----ATCCA	-37	7	AAAAATCGATC**T**CGTCAAATTTcagacttatcca	-47
		TTAAT----TTCAG	-98	7		
nirB	262	TTGAT----ATCAA	-48	10	aaaggtGAATTTGATTT**A**CATCAATAAGcggggt	-41.5
sodA	284	TTGAT----ATTTT	-42	7	agtacgGCATTGATAAT**C**ATTTTCAATAtcattt	-34
fnr**	96	TTGAC----ATCAA	-7	9	atgttaAAATTGACAAA**T**ATCAATTACGgcttga	1
					ccttaaCAACTTAAGGG**T**TTTCAAATAGatagac	-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		

* 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.

** The second site listed in RegulonDB is not contained in the sequence data, which is only 96 bases long, from GenBank.

delimitated.

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 internet.

Appendix

The 18 unaligned DNA sequences containing CRP binding sites are used as the example throughout this paper. It is taken from Stormo *et al*. (1989)

```
cole1     TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATAGCGCGTGGTGTGAAAGACTGTTTTTTTGATCGTTTTCACAAAAATGGAAGTCCACAGTCTTGACAG

ecoarabop GACAAAAACGCGTAACAAAAGTGTCTATAATCACGGCAGAAAAGTCCACATTGATTATTTGCACGGCGTCACACTTTGCTATGCCATAGCATTTTTATCCATAAG

ecobglr1  ACAAATCCCAATAACTTAATACTTGACATTTGTTATATATAACTTTATAAATTCCTAAAATTACACAAAGTTAATAACTGTGAGCATGGTCATATTTTTATCAAT

ecocrp    CACAAAGCGAAAGCTATGCTAAAACAGTCAGGATGCTACAGTAATACATTGATGTACTGCATGTATGCAAAGGACGTCACATTACCGTGCAGTACAGTTGATAGC

ecocya    ACGGTGCTACACTTGTATGTAGCGCATCTTTCTTTACGGTCAATCAGCAAGGTGTTAAATTGATCACGTTTTAGACCATTTTTTCGTCGTGAAACTAAAAAAACC

ecodeop   AGTGAATTATTTGAACCAGATCGCATTACAGTGATGCAAACTTGTAAGTAGATTTCCTTAATTGTGATGTGTATCGAAGTGTGTTGCGGAGTAGATGTTAGAATA

ecogale   GCGCATAAAAAACGGCTAAATTCTTGTGTAAACGATTCCACTAATTTATTCCATGGCACACTTTTCGCATCTTTGTTATGCTATGGTTATTTCATACCATAAGCC

ecoilvbpr GCTCCGGCGGGGTTTTTTGTTATCTGCAATTCAGTACAAAACGTGATCAACCCCTCAATTTTCCCTTTGCTGAAAAATTTTCCATTGTCTCCCCTGTAAAGCTGT

ecolac    AACGCAATTAATGTGAGTTAGCTCACTCATTAGGCACCCCAGGCTTTACACTTTATGCTTCCGGATCGTATGTTGTGTGGAATTGTGAGCGGATAACAATTTCAC

ecomale   ACATTACCGCCAATTCTGTAACAGAGATCACACAAAGCGACGGTGGGGCGTAGGGGCAAGGAGGATGGAAAGAGGTTGCCGTATAAAGAAACTAGAGTCCGTTTA

ecomalk   GGAGGAGGCGGGAGGATGAGAACACGGCTTCTGTGAACTAAACCGAGGTCATGTAAGGAATTTCGTGATGTTGCTTGCAAAAATCGTGGCGATTTTATGTGCGCA

ecomalt   GATCAGCGTCGTTTTAGGTGAGTTGTTAATAAAGATTTGGAATTGTGACACAGTGCAAATTCAGACACATAAAAAAACGTCATCGCTTGCATTAGAAAGGTTTCT

ecoompa   GCTGACAAAAAAGATTAAACATACCTTATACAAGACTTTTTTTTCATATGCCTGACGGAGTTCACACTTGTAAGTTTTCAACTACGTTGTAGACTTTACATCGCC

ecotnaa   TTTTTTAAACATTAAAATTCTTACGTAATTTATAATCTTTAAAAAAAGCATTTAATATTGCTCCCCGAACGATTGTGATTCGATTCACATTTAAACAATTTCAGA

ecouxul   CCCATGAGAGTGAAATTGTTGTGATGTGGTTAACCCAATTAGAATTCGGGATTGACATGTCTTACCAAAAGGTAGAACTTATACGCCATCTCATCCGATGCAAGC

pbr-p4    CTGGCTTAACTATGCGGCATCAGAGCAGATTGTACTGAGAGTGCACCATATGCGGTGTGAAATACCGCACAGATGCGTAAGGAGAAAATACCGCATCAGGCGCTC

trn9cat   CTGTGACGGAAGATCACTTCGCAGAATAAATAAATCCTGGTGTCCCTGTTGATACCGGGAAGCCCTGGGCCAACTTTTGGCGAAAATGAGACGTTGATCGGCACG

(tdc)     GATTTTTATACTTTAACTTGTTGATATTTAAAGGTATTTAATTGTAATAACGATACTCTGGAAAGTATTGAAAGTTAATTTGTGAGTGGTCGCACATATCCTGTT
```

References

Balas,E., Ceria,S. and Cornuéjols,G. (1996) Mixed 0-1 Programming by Lift-and-Project in a Branch-and-Cut Framework. *Management Science*, **42**, 1229-1246.

Brazma,A., Jonassen,I., Eidhammer,I. and Gilbert,D. (1998) Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.*, **5**, 279-305.

Ecker,J.G., Kupferschmid,M., Lawrence,C.E., Reilly, A.A. and Scott, A.C.H. (2002) An application of nonlinear optimization in molecular biology. *European Journal of Operational Res.,* **138**, 452-458.

Hertz,G.Z., Hartzell,G.W. and Stormo,G.D. (1990) Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Computa. Appl. Biosci.*, **6**, 81-92.

Huerta,A.M., Salgado,H., Thieffry,D. and Collado-Vides,J. (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: Structure, Function, and Genetics*, **7**, 41-51.

Schrage,L. (1999) *Optimization Modeling With Lingo*, LINDO Systems Inc., Chicago.

Stormo,G.D. and Hartzell,G.W. (1989) Identifying protein-binding sites from unaligned DNA fragments. *Proceedings of the National Academy of Sciences of the USA*, 86, 1183-1187.

Stormo,G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, **16**, 16-23.

Tan,K., Moreno-Hagelsieb,G., Collado-Vides,J. and Stormo,G.D. (2001) A comparative genomics approach to prediction of new members of regulons. *Genome Research*, **11**, 566-584.

八、計畫成果自評

1、本研究已發展以全域最佳化方法求解 DNA 序列之共同區間定址問題，研究成果也已發表於生物資訊之國際知名頂級期刊 Bioinformatics：

Han-Lin Li, and Chang-Jui Fu, “A Linear Programming Approach for Identifying a Consensus Sequence on DNA Sequences”, Bioinformatics, 21, 1838-1845, May 2005.

2、本研究正陸續求解更深入的課題，並將於 INFORMS 2006 H.K. Conference 發表：

Han-Lin Li, and Chang-Jui Fu, “Identifying DNA Consensus Sequence Without Shared Patterns Using Linear Programs”, INFORMS International Conference, Hong Kong, June 25-28, 2006.

3、本研究已訓練數位博士研究生完成撰寫論文中。