Available online at www.sciencedirect.com
SCIENCE DIRECT®
Fuzzy Sets and Systems 152 (2005) 83–102
FUZZY sets and systems
www.elsevier.com/locate/fss

A hybrid promoter analysis methodology for prokaryotic genomes

V. Cotik[a], R. Romero Zaliz[a, c], I. Zwir[b, c, *]

[a] *Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina*
[b] *Department of Molecular Microbiology, Howard Hughes Medical Institute, Washington University School of Medicine, USA*
[c] *Departamento de Ciencias de la Computación e Inteligencia Artificial, ETS de Ingeniería Informática, Universidad de Granada, Spain*

Available online 14 November 2004

Abstract

One of the big challenges of the post-genomic era is identifying regulatory systems and integrating them into genetic networks. Gene expression is determined by protein–protein interactions among regulatory proteins and with RNA polymerase(s), and protein–DNA interactions of these *trans*-acting factors with *cis*-acting DNA sequences in the promoter regions of those regulated genes. Therefore, identifying these protein–DNA interactions, by means of the DNA motifs that characterize the regulatory factors operating in the transcription of a gene, becomes crucial for determining which genes participate in a regulation process, how they behave and how they are connected to build genetic networks.

In this paper, we propose a hybrid promoter analysis methodology (HPAM) to discover complex promoter motifs that combines: the neural network efficiency and ability of representing imprecise and incomplete patterns; the flexibility and interpretability of fuzzy models; and the multi-objective evolutionary algorithms capability to identify optimal instances of a model by searching according to multiple criteria. We test our methodology by learning and predicting the RNA polymerase motif in prokaryotic genomes. This constitutes a special challenge due to the multiplicity of the RNA polymerase targets and its connectivity with other transcription factors, which sometimes require multiple functional binding sites even in close located regulatory regions; and the uncertainty

* Corresponding author.
E-mail addresses: vcotik@dc.uba.ar (V. Cotik), rromero@dc.uba.ar (R. Romero Zaliz), zwir@borcim.wustl.edu, zwir@decsai.ugr.es (I. Zwir).

0165-0114/$ - see front matter
doi:10.1016/j.fss.2004.10.016

of its motif, which allows sites with low specificity (i.e., differing from the best alignment or consensus) to still be functional. HPAM is available for public use in http://soar-tools.wustl.edu.

Keywords: Gene regulation; Prokaryotic promoters; RNA polymerase; Pattern recognition; Time delay neural networks; Multi-objective evolutionary algorithms; Fuzzy sets

1. Introduction

One of the big challenges of the post-genomic era is determining when, where and for how long genes are turned on or off [3]. Gene expression is determined by protein–protein interactions among regulatory proteins and with RNA polymerase(s), and protein–DNA interactions of these *trans*-acting factors with *cis*-acting DNA sequences in the promoter regions of those regulated genes [10,31]. Therefore, identifying these protein–DNA interactions, by means of the DNA motifs that characterize the regulatory factors operating in the transcription of a gene [18,32], becomes crucial for determining which genes participate in a regulation process, how they behave and how they are connected to build genetic networks.

Different computational methods have been applied to discover binding site motifs [1,14,16,18,23], however the problem remains open even for the simplest prokaryotic promoters. This happens because of the variability of the DNA motifs, which comprise more than one vague submotif arranged in direct or inverted tandem repeats, the fixed or variable distance separation between submotifs, and the availability of multiple occurrences of the former motifs and/or interactions between them in the promoter region of a gene (e.g. one combination of a *cis*-transcription factor and the RNA polymerase may correspond to the activation of a gene, while another to the repression of such gene [5,28]).

To address the promoter recognition problem in bacterial genomes, we propose a hybrid promoter analysis methodology (HPAM). HPAM is a machine learning approach, consisting of the sequential application of three different methods. First, we propose a time delayed neural network (TDNN) classification method, which learns binding site motifs from non-specific DNA sequences (i.e., a motif is known to be present in a training sequence, but their submotifs are not certainly identified). Particularly, this connectionist approach decomposes a compound binding site motif into modules, where a module corresponds to a motif feature (e.g., matching with the 1st tandem repeat, distance between repeats); enables the contribution of all available training examples to learn models for each module; and integrates individual models into a unique predictive model of the promoter motif. This is in contrast to previous approaches that generate different models for the same promoter motif [18], difficulting the inference and predicting processes, which are based on too many partitions of the data. In addition, the inference approach followed by neural networks allows to perform fast predictions in large genome sequences. Second, although the TDNN approach may achieve accurate results, sometimes, the subjacent model remains hidden because of the neural network black-box representation. Thus, we learn interpretable models from the TDNN findings by using fuzzy logic expressions with fuzzy predicates, whose membership functions are learned from probabilistic distributions [29,33,42]. Fuzzy set theory offers excellent tools for representing the uncertainty associated with the modular decomposition task, providing smooth transitions between individual local submodels. It also facilitates the interpolation of various types of knowledge within a common framework (e.g., matching of DNA sequences and distances between DNA

submotifs), giving an appropriate balance between the complexity and the accuracy of the model [6]. Third, we propose a multi-objective scatter search (MOSS) pattern recognition method, which by taking advantage of the model-based representation of the promoter motifs, identifies instances of a motif using multiple criteria optimization techniques based on evolutionary algorithms. This approach, which differs from previous manual approaches [18], formalizes the search of compound promoter motifs into a mathematical optimization framework, providing an accurate as well as interpretable methodology to discover prokaryotic promoter motifs.

To test our methodology we identified one of the most important *cis*-factors by learning and predicting its DNA binding motif: the RNA polymerase. This enzyme is crucial in the regulatory process because it transcribes genes or recruits other regulatory factors to do it cooperatively [31]. The DNA compound motif corresponding to this enzyme comprises two distinct submotifs separated by variable distances.

This paper is organized as follows: Section 2 describes the biological problem of discovering RNA polymerase binding sites; Section 3 describes the HPAM methodology; Section 4 shows and explains the results obtained by applying the HPAM methodology to discover promoter motifs in *Escherichia coli* (*E. coli*); and Section 5 summarizes the concluding remarks.

2. Problem: discovering RNA polymerase binding sites in DNA sequences

Prokaryotic promoter data gathered and analyzed by many compilations [14,15,24] reveal the presence of two well conserved sequences or submotifs separated by variable distances and a less conserved sequence. The variability of the distance between submotifs and their fuzziness, in the sense that they present several mismatches, hinder the existence of a clear binding site model of prokaryotic promoters. The most representative RNA polymerase promoters (i.e., $\sigma 70$ subunits) are described by the following conserved patterns:

(1) *Transcription start site* (+1): In general, a pyrimidine (C or T) followed by a purine (A or G) compose the (+1) motif. Usually the (+1) is the second base of the sequence CA.
(2) *TATAAT*: This pattern is a hexanucleotide conserved sequence, whose middle nucleotide is located approximately 10 bp upstream of the (+1). It is often called *-10 submotif*.
(3) *TTGACA*: This pattern is also a hexanucleotide conserved sequence whose middle nucleotide is located approximately 35 bp upstream of the (+1). It is often called *-35 submotif*.
(4) *Distance(TTGACA, TATAAT)*. The distance between the TTGACA and TATAAT consensus submotifs follows a data distribution between 15 and 21 bp. This distance is critical in holding the two sites at the appropriate distance for the geometry of RNA polymerase [15].

The study of the gene regulation problem requires the identification of the RNA polymerase binding site, which allows to describe the gene condition of being activated or repressed; to delimit regulatory regions, where different transcription factors can interact; and to define classes of protein–protein interactions with transcription factors (e.g., class I or II promoters [19]). Therefore, the identification of the RNA polymerase constitutes a special challenge due to the multiplicity of the its targets and its connectivity with other transcription factors, which sometimes require multiple binding sites to be functional even in close located regulatory regions; and the uncertainty of its motif, which allows sites with low specificity (i.e., differing from the best alignment or consensus) to still be functional.

3. HPAM: a hybrid promoter analysis methodology

We propose HPAM to perform accurate and interpretable predictions of promoter motifs in prokaryotes (see Fig. 1). This methodology has been developed as a hybrid approach that combines the neural network efficiency and ability of representing imprecise and incomplete patterns [33], the flexibility and interpretability of models represented as fuzzy sets [20], and the multi-objective evolutionary algorithms capability to identify optimal instances of a model by searching according to multiple criteria [42].

3.1. First step: a time delay neural network classification method

Neural Networks have been widely used for promoter recognition tasks [17,30,40] because they can capture imprecise and incomplete patterns, such as individual promoter motifs including mismatches. However, compound promoter motifs need more flexible models to capture the variability of the distance between them [18]. Therefore, we propose a TDNN, which is a multilayer feed-forward neural network that learns patterns independently of their input location [22,38] from training DNA sequences containing motifs with unspecific length (i.e., submotifs are not certainly identified or the distance between them is variable). This TDNN harbors a modular network architecture, which uses all available training examples

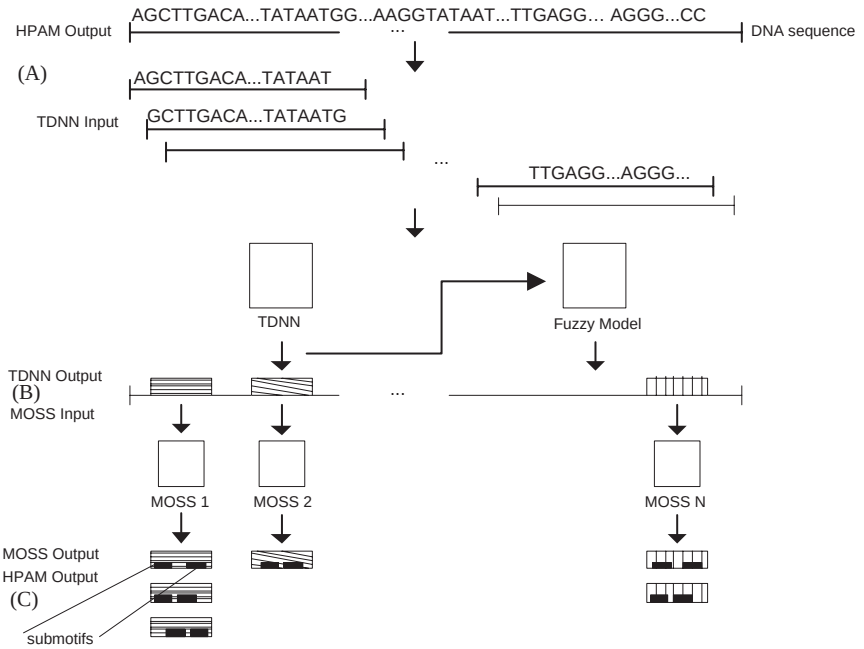

Fig. 1. HPAM hybrid methodology. (A) The TDNN receives as input the original DNA sequence, which is split into fix length windows. (B) The output of the TDNN is a set of predicted promoters, with the locations of their conserved sequences. This output is preprocessed by building probability distributions for the identified submotif, calculating histograms of the frequencies of their nucleotides and their distances. Fuzzy models are learned from the former distributions. Each neural network prediction and the fuzzy models are used as the input of the MOSS method. (C) The MOSS is used to specifically recognize all optimal motifs located in a DNA sequence.

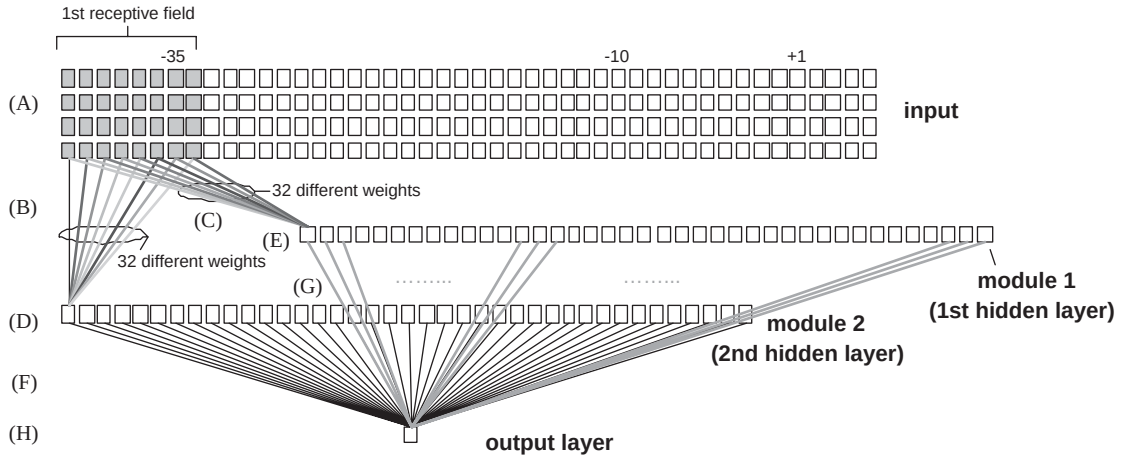

Fig. 2. TDNN Architecture. (A) The input layer, where each row represents a nucleotide (i.e., A,T,C,G) and each column a position in a DNA sequence. Serial receptive fields associate subsets of 8 positions. (B and C) Connections between each receptive field and the hidden layers, where each unit is connected with one receptive field and the 32 resulting weight values are repeated in each connection. (D and E) The hidden layers representing motif modules (e.g., conserved submotifs such as TATAAT and TTGACA in the RNA polymerase binding site). (F and G) The connections between all hidden units of both hidden layers and the output neuron. (H) The output neuron.

to build a unique—as opposed to several [18]—model for a transcription factor binding site. Particularly, to identify the RNA polymerase compound motif, we propose the use of two modules, each one representing a submotif and implemented as one hidden layer, which are linked by another connecting module representing their distances and implemented as the output layer (see Fig. 2).

3.1.1. Network architecture

We use an orthogonal input representation, which uses four binary input units per nucleotide (i.e. A: (1, 0, 0, 0); C: (0, 1, 0, 0); G: (0, 0, 1, 0); T: (0, 0, 0, 1)) to preserve the same Hamming distance between every pair of vectors. This representation avoids possible learning biases such as the correlation among different nucleotides codification vectors [2,9,13,25]. As we already stated, prokaryotic composed promoter motifs always have two or more conserved sequences (usually between 5 and 8 bp) separated by fixed (usually between 0 and 8 bp) or variable distances (usually between 15 and 21 bp) [34]. Particularly, the input layer considered for the RNA polymerase motif includes sequences of $6 + 6 + 21 + 12 = 46$ nucleotides. Since each nucleotide is represented by 4 input units, the TDNN has an input layer of $46 \times 4 = 184$ input units. For each module of conserved submotifs, a hidden layer is added to the neural network. These modules are linked by a new module representing their distance relationship. The proposed architecture is illustrated in Fig. 2, where each unit in a hidden layer is connected to 8 consecutive nucleotides of the input layer. Hence, each hidden layer has $46 - 8 + 1 = 39$ units. The 32 input units connected to each hidden unit are called the receptive fields of the unit. The output layer consists of one unit connected to each of the 39 units of both hidden layers with its value being a real number.

We use the *correlation coefficient* (CC) [26], which calculates the correlation between predictions and observations independently of the amount of positives or negatives examples, to determine the best

threshold to discriminate between promoters and non-promoters:

$$CC = \frac{(TP' \times TN') - (FN' \times FP')}{\sqrt{(TP' + FN') \times (TN' + FP') \times (TP' + FP') \times (TN' + FN')}}, \quad (1)$$

where

$$TP' = \frac{TP \times 100}{p},\ TN' = \frac{TN \times 100}{r},\ FP' = \frac{FP \times 100}{r},\ FN' = \frac{FN \times 100}{p}, \quad (2)$$

p is the promoter quantity and r the non-promoter quantity in the test set, TP is the number of true positive results, TN the number of true negative results, FP the number of false positive results and FN the number of false negative results.

3.1.2. Learning algorithm

We use a modification of the back-propagation algorithm, where the first step consists of a forward and a backward pass of the traditional algorithm. The weights are organized as blocks or receptive fields of 32 connections, which are repeated for each unit of the hidden layers (see Fig. 2). Each individual weight is updated by averaging all weight variations obtained from a forward pass (see Fig. 3, Step 6). The complete algorithm is illustrated in Fig. 3.

In order to calculate the number of training epochs and to prevent the overfitting of the model, a small portion of the training set was used as validation data [39]. The training phase is stopped when the validation error achieved its lower value. This procedure is performed five times and the epochs are averaged. Finally, the net is retrained using the whole training data, including the validation set, up to the number of epochs previously determined.

3.2. Second Step: a model-based representation of promoter motifs based on fuzzy sets

The TDNN performs accurate predictions of promoter regions in DNA sequences. However, because neural network models are black boxes, the resulting weights are not always interpretable [27]. Therefore, we analyze the output of the TDNN network to extract knowledge for building an interpretable promoter model based on fuzzy sets. We inspect the submotifs recognized by each module of the network to obtain the frequency of nucleotides, as well as the frequency distribution of the distances between them. Then, we perform histograms and learn membership functions of fuzzy sets representing each one of the network modules. Therefore, a modular neural network topology can be transformed into modular fuzzy logic expressions with fuzzy predicates, whose membership functions are learned from probabilistic distributions [20].

The membership functions corresponding to the fuzzy models of the RNA polymerase binding site motif are calculated by using the information of their nucleotide consensus frequency as discrete fuzzy sets [20]:

$$\mu_{\text{tataat}}(X) = \mu_1^1(x_1) \cup \cdots \cup \mu_6^1(x_6), \quad (3)$$

where $X = \{x_1, \ldots, x_n\}$ is a sequence of n nucleotides, the fuzzy discrete set corresponding to the first nucleotide of the submotif $T_{0.77}A_{0.76}T_{0.60}A_{0.61}A_{0.56}T_{0.82}$ is defined as $\mu_1^1(x_1) = A/0.08 + T/0.77 + G/0.12 + C/0.05$, the other fuzzy sets corresponding to 2–6 positions are calculated in a similar way according to data distributions, and the union corresponds to fuzzy set operations [20,29].

$$\mu_{\text{ttgaca}}(Y) = \mu_1^2(y_1) \cup \cdots \cup \mu_6^2(y_6), \quad (4)$$

Input:
η: learning rate,
Set of training patterns. $\mu = (\xi^\mu, \zeta^\mu)$ is a training set pattern, ξ is the input and ζ the desired output.
Notation:
V_i^m: output of unit i in layer m,
w_{ij}^{nm}: synaptic weight from neuron j of layer n to neuron i of layer m.

1: **Weight and threshold initialization.**
2: **repeat** {Training patterns presentation randomly ordered. An epoch is presented to the network. Take a pattern μ and present it to the input, in this way $V_k^0 = \xi_k^\mu$.}
3: **while** a non-processed pattern exists **do**
4: **Forward step** Calculate:
$V_i^m = g(h_i^m) = g(\sum_{j=0}^{32} w_{ij}^{0,m} \cdot V_j^0)$, for $m \in 1, 2$, $i \in \{1, 39\}$
$V_1^3 = g(h_1^3) = g((\sum_{j=1}^{39}(w_{1j}^{1,3} \cdot V_j^1 + w_{1j}^{2,3} \cdot V_j^2)) - w_{10}^3)$, for the output layer,
where g is the activation function
5: **Backward step.** Local gradient calculation (δ) of the network
$\delta_1^3 = g'(h_1^3)[\zeta_1^\mu - V_1^3]$ (output layer)
$\delta_i^m = g'(h_i^m) w_{1i}^{m,3} \delta_1^3$, where $m \in 1, 2$ (hidden layers)
6: **Weight updates**
$\Delta w_{1j}^{n,3} \leftarrow \eta \cdot \delta_1^3 \cdot V_j^n$, where $n \in 1, 2$
$\Delta w_{ij}^{0,m} \leftarrow \eta \cdot \delta_i^m \cdot V_j^0$, where $m \in 1, 2$
$w_{1j}^{n,3} \leftarrow w_{1j}^{n,3} + \Delta w_{1j}^{n,3}$, where $n \in 1, 2$
$average_j \leftarrow \frac{\sum_{i=1}^{39} \Delta w_{ij}^{0,m}}{39}$, $j \in \{0, 32\}$, $m \in \{1, 2\}$
$w_{ij}^{0,m} \leftarrow w_{ij}^{0,m} + average_j^m$, where $m \in 1, 2$, $i \in \{1, 39\}$, $j \in \{0, 32\}$ {The average calculation makes it possible to maintain the equality between corresponding weights (weights, which from different hidden units to different receptive fields have the same value)}
7: **end while**
8: **until** satisfaction of stopping criteria

Fig. 3. TDNN Learning Algorithm.

where $Y = \{y_1, \ldots, y_m\}$ is a sequence of m nucleotides, the fuzzy crisp set corresponding to the first nucleotide of the submotif $T_{0.69}T_{0.79}G_{0.61}A_{0.56}C_{0.54}A_{0.54}$ is defined as $\mu_1^2(x) = A/0.12 + T/0.69 + G/0.13 + C/0.06$, the fuzzy sets corresponding to 2–6 positions are calculated in a similar way according to data distributions, and the union also corresponds to fuzzy set operations [20,29]. The distances between submotifs are accumulated into histograms and used for learning the triangular membership function $\mu_{\text{distance}}(X, Y)$ (see Fig. 4).

3.3. Third step: a multi-objective scatter search pattern recognition method

The above model-based representation is used in a MOSS pattern recognition method [11,21]. The MOSS considers the matching of a DNA sequence with each promoter submotif and their distance as *multiple objectives* to be optimized. Moreover, the pattern recognition process is also deemed by MOSS as a *multi-modal* problem, since, as was stated in Section 2, more than one solution can be found in each promoter region. The evolutionary algorithm used in MOSS is an extension of the original *scatter search*

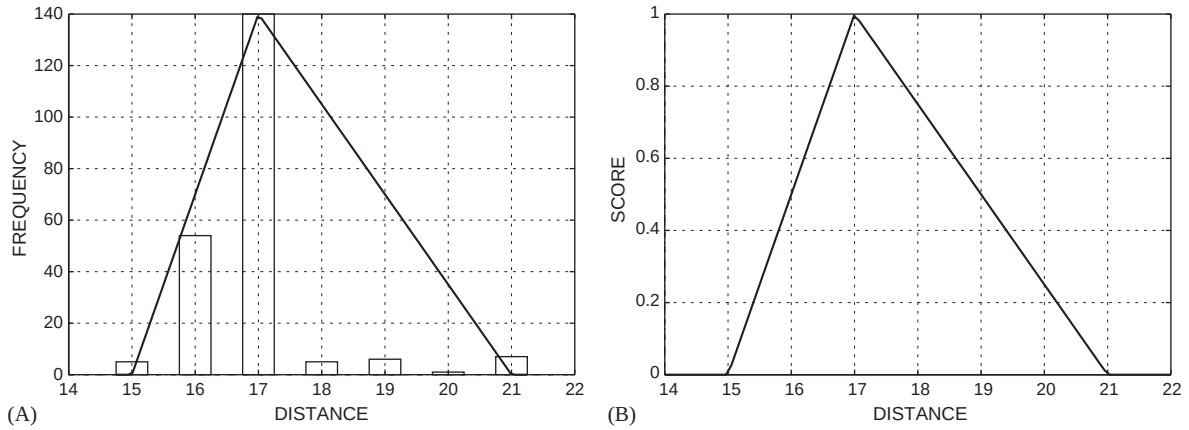

Fig. 4. Graphical representation of the fuzzy model (μ_{distance}). (A) The histogram corresponding to the frequency of the distances between submotifs. (B) The fuzzy membership function learned from (A).

(SS) heuristic [21] that uses the promoter regions detected by the TDNN, avoiding the intractability of applying evolutionary algorithms in searching spaces consisting of hundreds of bp, and the learned fuzzy models identified for each promoter module as inputs, and finds all optimal instances that satisfy the model constraints. Therefore, to extend the original SS algorithm to a multi-objective environment we need to introduce some concepts [4,8]:

Definition 1. A multi-objective optimization problem is defined as:

$$\begin{aligned} \text{Minimize/Maximize } & f_m(x), && m = 1, 2, \ldots, M, \\ \text{subject to } & g_j(x) \geqslant 0, && j_g = 1, 2, \ldots, J, \\ & h_k(x) = 0, && k = 1, 2, \ldots, K, \\ & x_i^{(L)} \leqslant x_i \leqslant x_i^{(U)}, && i = 1, 2, \ldots, n. \end{aligned}$$

where M corresponds to the number of problem objectives, J to the number of inequality constraints, K to the number of equality constraints and n is the number of decision variables. The last set of constraints restrict each decision variable x_i to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound.

Specifically, in the identification of the RNA polymerase binding site problem, we consider the following instantiations:

- $M = 3$. We have three objectives consisting of maximizing the degree of matching between the fuzzy models (fuzzy membership) and an instance corresponding to a DNA sequence: $f_1 = \mu_{\text{tataat}}(X)$ and $f_2 = \mu_{\text{ttgaca}}(Y)$ are the objective functions for each submotif, and $\mu_{\text{distance}}(X, Y)$ corresponds to the distance between them (recall Eqs. (3) and (4), and Fig. 4 respectively).
- $J = 1$. We have just one constraint g_1 corresponding to the distance between submotifs, which cannot be less than 15 and no more than 21 bp.
- $K = 0$. No equality restrictions are needed.
- Only valid solutions are kept in each generation.

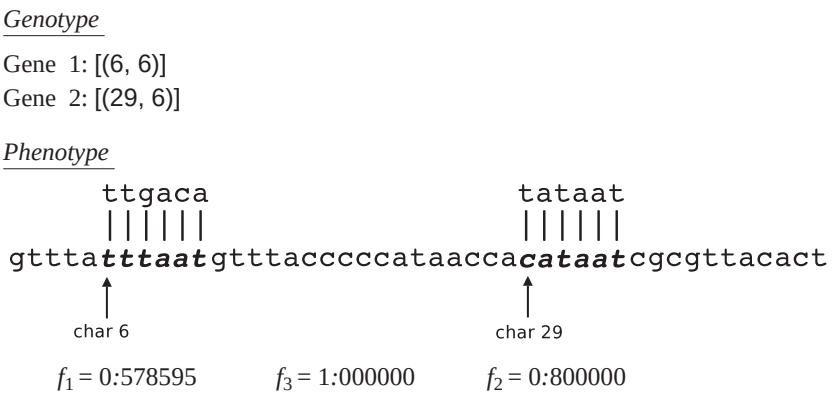

Fig. 5. The representation of an individual in MOSS. Each chromosome is composed of two genes, one for each submotif. Each gene has two integer numbers representing the starting position of the pattern and its size, respectively. The distance between submotifs is not saved in the chromosome, but inferred from the distance between the position of the genes.

- The submotifs cannot be located outside the sequence searched, that is, it cannot start at negative positions or greater than the length of the query sequence.

Definition 2. A solution x is said to dominate solution y ($x \prec y$), if both conditions 1 and 2 are true: (1) The solution x is no worse than y in all objectives: $f_i(x) \not\triangleright f_i(y)$ for all $i = 1, 2, \ldots, M$; (2) The solution x is strictly better than y in at least one objective: $f_j(x) \triangleleft f_j(y)$ for at least one $i \in \{1, 2, \ldots, M\}$. If x dominates the solution y it is also customary to write that x is *non-dominated* by y.

3.3.1. Combination operator and local search

We use a block representation to code each individual, where each block corresponds to one of the promoter submotifs (e.g., TATAAT or TTGACA submotifs). Particularly, each block is represented by two integers, where the first number corresponds to the starting point of the submotif, and the second one represents its size (see Fig. 5).

The combination process is implemented as a one-point combine operator, where the point is always located between both blocks [42]. For example, given chromosomes with two blocks A and B, and parents $P = A_1B_1$ and $P' = A_2B_2$, the corresponding siblings would be $S = A_1B_2$ and $S' = A_2B_1$. The *local search* is implemented as a search for non-dominated solutions in a certain neighborhood. For example, a local search performed on the chromosome space includes a specified number of nucleotides located on the left or right sides of the blocks composing the chromosome. The selection process considers that a new mutated chromosome that dominates one of its parents will replace it, but if it becomes dominated by its ancestors no modification is performed. Otherwise, if the new individual is not dominated by the non-dominated population found so far, it replaces its father only if it is located in a less crowded region (see Fig. 6).

3.3.2. Algorithm

We modified the original SS algorithm to allow multiple-objective solutions by adding the *non-dominance* criterion to the solution ranking [8]. Thus, non-dominated solutions were added to the set

1: Randomly select which block g in the representation of the individual c to apply local search.
2: Randomly select a number n in $[-neighbor, neighbor]$ and move the block g, n nucleotides. Notice that it can be moved upstream or downstream. Resulting block will be g' and resulting individual will be called c'.
3: **if** c' meets the restrictions **then**
4: **if** c' dominates c **then**
5: Replace c with c'
6: **end if**
7: **if** c' does not dominate c and c' is not dominated by c and c' is not dominated by any solution in the Non-Dominated set **then**
8: Replace c with c' if $crowd(c') < crowd(c)$.
9: **end if**
10: **end if**

Fig. 6. Local search process used by the MOSS method.

1: Start with $P = \emptyset$. Use the generation method to build a solution and the local search method to improve it. If $x \notin P$ then add x to P, else, reject x. Repeat until P has the user specified size.
2: Create a reference set $RefSet$ with $b/2$ non-dominated solutions of P and $b/2$ solutions of P more diverse from the other $b/2$. If there are not enough non-dominated solutions to fill the $b/2$, complete the set with dominated solutions.
3: $NewSolution \leftarrow$ `true`
4: **while** Exists a Solution not yet explored (NewSolution = true) **do**
5: $NewSolution \leftarrow$ `false`
6: Generate subsets of $RefSet$ where there is at least one non-dominated solution in each one.
7: Generate an empty subset N to store non-dominated solutions.
8: **while** subset to examine **do**
9: Select a subset and mark it as examined.
10: Apply combination operators to the solutions in the set.
11: Apply local search to each new solution x found after the combination process as explained in Fig. 6 and name it x^b.
12: **if** x^b is non-dominated by any $x \in N$ and $x^b \notin N$ **then**
13: Add x^b to N.
14: **end if**
15: **end while**
16: Add solutions $y \in N$ to P if there is no solution $z \in P$ that dominates y.
17: $NewSolution \leftarrow$ `true`.
18: **end while**

Fig. 7. MOSS algorithm.

in any order, but dominated solutions were only added if no more non-dominated solutions could be found. In addition to maintaining a good set of non-dominated solutions, and to avoid one of the most common problems of multi-objective algorithms such as multi-modality [8], we also keep track of the diversity of the available solutions through all generations. Finally, the initial populations are created randomly and unfeasible solutions corresponding to out of distance ranges between promoter submotifs (g_1) are checked at each generation. Fig. 7 clearly illustrates the MOSS algorithm.

4. Experiments and results

The three-step HPAM methodology was applied to a set of sequences known to contain RNA polymerase binding sites with more than one alternative motif in the same regulatory region [14]. In this work 272 candidate promoter regions were identified by means of their -35 and -10 submotifs. We randomly selected 60% of the promoters as a training set, while another 40% was used as a test set. We considered DNA sequences of 46 bp, where shorter sequences reported in [14] have been completed from the sequences of Genbank. The negative data was represented by randomly generated sequences with equal probability for each nucleotide. The proportion of positives versus negatives examples for the training set was 1:10, and 1:25 for the test set. The threshold corresponding to the best CC value was 0.11 [7]. The TDNN was implemented with the Stuttgart neural network simulator (SNNS) [1] and the execution parameters can be seen in Table 1.

The fuzzy models, by means of their membership functions, were learned from probability distributions corresponding to the frequencies of the nucleotides composing the submotifs and the distances between them. We inspected the submotifs and distances recognized by the TDNN in the training set, aligned the extracted subsequences for each submotif using Clustal_X [37], built histograms for submotifs and distance values (see Fig. 4), and learned fuzzy membership functions by projecting the former distributions into triangular functions [36]. Unsurprisingly, and in agreement with the literature [14,24], the two modules of the TDNN recognize the TATAAT and TTGACA consensus motifs, respectively, as well as the distance distribution between them centered in 17 bp.

The MOSS method was executed 20 times with different seeds for each input sequence with the parameters listed in Table 2. Both methods, TDNN and MOSS, were individually compared with other methods. On the one hand, the TDNN performance was compared with a string-based method, [2] which considers mismatches from a consensus as crisp probabilities in a (0,1) representation. In addition, we also compared our approach with the Consensus-Patser [16] method, which represents motifs as probabilistic

Table 1
The TDNN Parameters: η specifies the step width of the gradient descent, $d_{\max}$ is the maximum d_j where $d_j = t_j - o_j$ between a teaching value t_j and an output value o_j which is propagated back as $d_j = 0$

Parameter	Value
Activation function of hidden and output neurons:	Act_Logistic
Output function of hidden: neurons:	Out_Identity
Shuffle:	On
Update function:	TimeDelay_Order
Learning function:	TimeDelayBackprop with η=0.2 and $d_{\max}$=0
Weights and bias Initialization:	Randomize Weights with values in $[-0.01, 0.01]$
Average Epochs:	47

[1] SNNS is (c) (Copyright) 1990–95 SNNS Group, Institute for Parallel and Distributed High-Performance Systems (IPVR), University of Stuttgart, Breitwiesenstrasse 20–22, 70565 Stuttgart, Fed. Rep. of Germany.

[2] http://rsat.ulb.ac.be/rsat/, option: dna-pattern.

Table 2
Parameters of the MOSS method.

Parameter	Value
Number of generations	200
RefSet	16
Non-Dominated population size	300

Table 3
Results obtained by the TDNN method with training and test sets.

	0%FP		1% FP		5% FP		100%TP		80%TP		Greatest CC		
Method	%TP	CC	%TP	CC	%TP	CC	%FP	CC	%FP	CC	CC	%TP	%FP
Training set	41.45	0.51	83.55	0.84	93.42	0.89	17.83	0.82	0.72	0.81	0.90	91.45	1.91
Test set	13.33	0.27	76.67	0.78	93.33	0.88	23	0.79	1.4	0.8	0.90	95.83	5.83

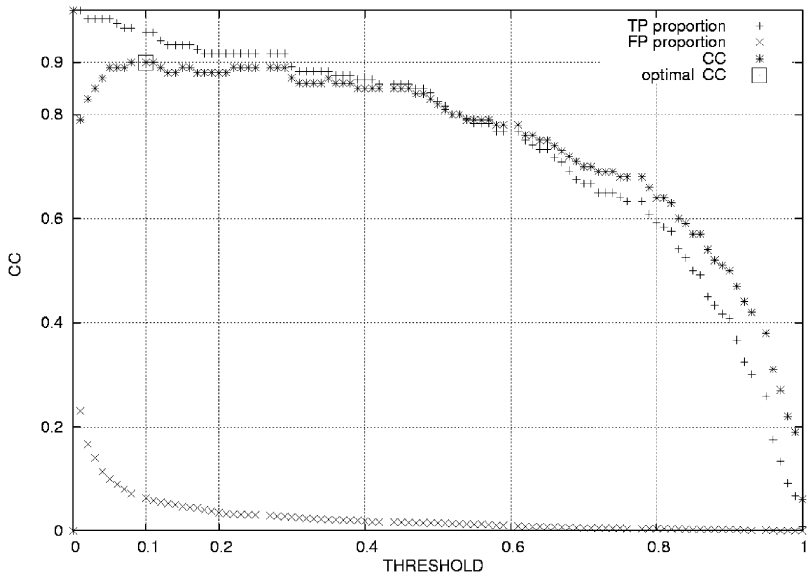

Fig. 8. Selection of the best CC as a trade-off between TP and FP examples in the test set. The optimal CC determines a threshold value used to discriminate between positive and negative promoter motifs, which is measured in a [0,1] scale.

matrices in a [0,1] scale. On the other hand, the MOSS method was compared with other evolutionary approaches: the Strength Pareto Evolutionary Algorithm (SPEA) [41] and the $(\mu + \lambda)$ GA [35].

We show in Table 3 the results obtained by the TDNN in training and test sets. The performance of our connectionist approach was tested under five different rates of TP and FP constraints and the highest CC values, which selection is illustrated in Fig. 8.

We compared the results obtained by the TDNN method with the crisp and probabilistic methods in Table 4. The TDNN achieved the best CC value, which represents the best trade-off solution between

Table 4
Comparisons among the TDNN, the crisp and the probabilistic methods with the test set

Method	Highest CC	% TP	% FP
TDNN	0.90	95.83	5.83
Crisp Method	0.46	43.33	3.90
Consensus-Patser	0.68	74	7

Fig. 9. Different solutions for the *Ada* sequence—Three different alternative locations for the preserved sequences were included in the final set of the HPAM method matching with the three alternatives reported in the literature.

Table 5
Comparisons among MOSS and other multi-objective evolutionary algorithms

	Original	Alternative	%originals	%alternatives	Total	%total
MOSS	243	59	93.10%	86.76%	302	91.79%
SPEA	217	43	83.14%	63.24%	260	79.03%
$(\mu + \lambda)$ GA	223	52	85.44%	76.47%	275	83.59%

The *Original* and *Alternative* columns indicate the number of RNA polymerase binding sites reported in the literature [14].

TP and FP examples (0.90 of the TDNN vs. 0.46 and 0.68, corresponding to the crisp and probabilistic methods, respectively).

It is important to notice that there is more than one possible description for each promoter region, as it is illustrated in Fig. 9 for the gene *Ada* reported in [14]. We test the ability of the MOSS method by considering the 261 promoter regions recognized by the TDNN method and the 68 alternative solutions (i.e., multiple promoters) defined in [14] for the corresponding sequences totalizing 329 regions. The MOSS method, by using a fuzzy model-based approach, recognizes 93.10% and 86.76% of the original and alternative solutions, respectively. The comparisons with other evolutionary algorithms, such as SPEA and $(\mu + \lambda)$, are illustrated in Table 5 and clearly reveal that the proposed MOSS overcomes the other methods. The complete set of results is listed in the Appendix A.

5. Concluding remarks

We have proposed a three-step promoter motif recognition methodology termed HPAM, which combines the advantages of different machine learning techniques, including neural networks, fuzzy model-based representations and multi-objective evolutionary algorithms. Particularly, we applied HPAM to the identification of the RNA polymerase motif, which constitutes a special challenge due to its multiple regulatory roles and its incidence in different regulatory pathways.

The modular representation of compound binding site motifs provided by the TDNN architecture allows us to obtain unique models even from training examples with variable length. Moreover, the

conservation of these modular representation in the fuzzy models and their combination with the MOSS pattern recognition method produces multiple, optimal and interpretable solutions, providing exhaustive descriptions of the promoter regions of the prokaryotic genomes by means of the RNA polymerase occurrences.

Finally, as David Goldberg stated, the integration of single methods into hybrid intelligent systems goes beyond simple combinations. For him, the future of Computational Intelligence lies in the careful integration of the best constituent technologies and subtle integration of the abstraction power of fuzzy systems and the innovating power of evolutionary systems requires a design sophistication that goes further than putting everything together [12]. The present implementation of HPAM is available in http://soar-tools.wustl.edu.

Appendix A.

Tables 6 and 7 illustrate the set of solutions found by HPAM by considering the set of promoter examples published in [14]. The last column of the tables indicates whether the neural network recognized the promoter or not by a ✓ or □. Only those sequences that were recognized by the TDNN constitute the input of the MOSS method. The first column corresponds to the name of the sequence, second column shows the beginning character position of the TTGACA motif, while the third column shows the beginning character position of the TATAAT motif. This positions were the ones detected by the MOSS. Only one result for each sequence is shown due to space limitations. The fourth column corresponds to the sequence itself with each of the submotifs clearly depicted.

Table 6
Results for the training sequences

Sequence	ttgaca	tataat	Promoter	found
aceEF	13	36	ACGTAGACCTGT CTTATT GAGCTTTC CGGCGAGAG TTCAAT GGGACAGGTCCAG	✓
ada	–	–	AGCGGCTAAAGGTG TTGACG TGCGAGAA ATGTTTAGC TAAACT TCTCTCATGTG	□
alaS	15	39	AACGCATACGGTAT TTTACC TTCCCAGTC AAGAAAACT TATCTT ATTCCCACTTTTCAGT	✓
ampC	15	37	TGCTATCCTGACAG TTGTCA CGCTGATT GGTGTCGT TACAAT CTAACGCATCGCCAATG	✓
ampC/C16	7	30	GCTATC TTGACA GTTGTCAC GCTGATTGG TATCGT TACAATCTAACGTATCG	✓
araBAD	15	37	TTAGCGGATCCTAC CTGACG CTTTTTAT CGCAACTC TCTACT GTTTCTCCATACCCGTT	✓
araC	15	38	GCAAATAATCAATG TGGACT TTTCTGCC GTGATTATA GACACT TTTGTTACGCGTTTTTG	✓
araE	12	37	CTGTTTCCGAC CTGACA CCTGCGTGA GTTGTTCACG TATTTT TTCACTATGTCTTACTC	✓
araI(c)	13	35	AGCGGATCCTAC CTGGCG CTTTTTAT CGCAACTC TCTACT GTTTCTCCATACCCGTT	✓
araI(c)X(c)	13	37	AGCGGATCCTAC CTGGCG CTTTTTATC GCAACTCTC TACTAT TTCTCCATACCCGTTTT	✓
argCBH	15	39	TTTGTTTTTCATTG TTGACA CACCTCTGG TCATGATAG TATCAA TATTCATGCAGTATT	✓
argCBH-P1/6-	15	36	TTTGTTTTTCATTG TTGACA CACCTCT GGTCATAA TATTAT CAATATTCATGCAGTAT	✓
argCBH-P1/LL	15	36	TTTGTTTTTCATTG TTGACA CACCTCT GGTCATGA TATTAT CAATATTCATGCAGTAT	✓
argE-P1	15	38	TTACGGCTGGTGGG TTTTAT TACGCTCA ACGTTAGTG TATTTT TATTCATAAATACTGCA	✓
argE-P2	15	38	CCGCATCATTGCTT TGCGCT GAAACAGT CAAAGCGGT TATGTT CATATGCGGATGGCG	✓
argE/LL13	15	38	CCGCATCATTGCTT TGCGCT GAAACAGT CAAAGCGGT TATATT CATATGCGGATGGCG	✓
argF	15	38	ATTGTGAAATGGGG TTGCAA ATGAATAA TTACACATA TAAAGT GAATTTTAATTCAATAA	✓
argI	7	30	TTAGAC TTGCAA ATGAATAA TCATCCATA TAAATT GAATTTTAATTCATTGA	✓
argR	12	35	TCGTCGCCGCG TTGCAG GAGCAAGG CTTTGACAA TATTAA TCAGTCTAAAGTCTCGG	✓
aroF	15	37	TACGAAAATATGGA TTGAAA ACTTTACT TTATGTGT TATCGT TACGTCATCCTCGCTG	✓
aroG	15	38	AGTGTAAAACCCCG TTTACA CATTCTGA CGGAAGATA TAGATT GGAAGTATTGCATTCA	✓
aroH	15	37	GTACTAGAGAACTA GTGCAT TAGCTTAT TTTTTTGT TATCAT GCTAACCACCCGGCGAG	✓
bioA	15	39	GCCTTCTCCAAAAC GTGTTT TTTGTTGTT AATTCGGTG TAGACT TGTAAACCTAAATCT	✓
bioB	15	38	TTGTCATAATCGAC TTGTAA ACCAAATT GAAAAGATT TAGGTT TACAAGTCTACACCGAA	✓

Table 6 (*continued*)

sequence	ttgaca	tataat	promoter	found
bioP98	15	38	TTGTTAATTCGGTG TAGACT TGTAAACC TAAATCTTT TAAATT TGGTTTACAAGTCGAT	✓
C62.5-P1	–	–	CACCTGCTCTCGC TTGAAA TTATTCTC CCTTGTCCC CATCTC TCCCACATCCTGTTTT	□
carAB-P1	15	38	ATCCCGCCATTAAG TTGACT TTTAGCGC CCATATCTC CAGAAT GCCGCCGTTTGCCAGA	✓
carAB-P2	15	39	TAAGCAGATTTGCA TTGATT TACGTCATC ATTGTGAAT TAATAT GCAAATAAAGTGAG	✓
cat	13	36	ACGTTGATCGGC ACGTAA GAGGTTCC AACTTTCAC CATAAT GAAATAAGATCACTACC	✓
cit.util-379	–	–	AAACAGGCGGGG GTCTCA GGCGACTAA CCCGCAAAC TCTTAC CTCTATACATAATTCTG	□
cit.util-431	14	38	GACAGGCACAGCA TTGTAC GATCAACTG ATTTGTGCC AATAAT TAAATGAAATCAC	✓
CloDFcloacin	15	37	TCATATATTGACAC CTGAAA ACTGGAGG AGTAAGGT AATAAT CATACTGTGTATATAT	✓
CloDFnaI	15	39	ACACGCGGTTGCTC TTGAAG TGTGCGCCA AAGTCCGGC TACACT GGAAGGACAGATTTGG	✓
colE1-B	15	36	TTATAAAATCCTCT TTGACT TTTAAAA CAATAAGT TAAAAA TAAATACTGTAA	✓
colE1-C	15	37	TTATAAAATCCTCT TTGACT TTTAAAAC AATAAGTT AAAAAT AAATACTGTACATATAA	✓
colE1-P1	15	38	GGAAGTCCACAGTC TTGACA GGGAAAAT GCAGCGGCG TAGCTT TTATGCTGTATATAAAA	✓
colE1-P2	15	37	TTTTTAACTTATTG TTTTAA AAGTCAAA GAGGATTT TATAAT GGAAACCGCGGTAGCGT	✓
colE110.13	13	37	GCTACAGAGTTC TTGAAG TAGTGGCCC GACTACGGC TACACT AGAAGGACAGTATTTGG	✓
colicinE1 P3	15	37	TTTTTAACTTATTG TTTTAA AAGTCAAA GAGGATTT TATAAT GGAAACCGCGGTAGCGT	✓
crp	15	38	AAGCGAGACACCAG GAGACA CAAAGCGA AAGCTATGC TAAAAC AGTCAGGATGCTACAG	✓
cya	15	38	GTAGCGCATCTTTC TTTACG GTCAATCA GCAAGGTGT TAAATT GATCACGTTTTAGACC	✓
dapD	–	–	AAGTGCATCAGCGG TTGACA GAGGCCCTC AATCCAAAC GATAAA GGGTGATGTGTTTACTG	□
deo-P1	14	39	CAGAAACGTTTTA TTCGAA CATCGATCT CGTCTTGTGT TAGAAT TCTAACATACGGTTGC	✓
deo-P2	10	35	TGATGTGTA TCGAAG TGTGTTGCG GAGTAGATGT TAGAAT ACTAACAAACTCGCAA	✓
deo-P3	15	37	ACACCAACTGTCTA TCGCCG TATCAGCG AATAACGG TATACT GATCTGATCATTTAAA	✓
divE	15	38	AAACAAATTAGGGG TTTACA CGCCGCAT CGGGATGTT TATAGT GCGCGTCATTCCGGAAG	✓
dnaA-1p	15	39	TGCGGCGTAAATCG TGCCCG CCTCGCGGC AGGATCGTT TACACT TAGCGAGTTCTGGAAA	✓
dnaA-2p	15	38	TCTGTGAGAAACAG AAGATC TCTTGCGC AGTTTAGGC TATGAT CCGCGGTCCCGATCG	✓
dnaK-P1	15	39	TTTGCATCTCCCCC TTGATG ACGTGGTTT ACGACCCCA TTTAGT AGTCAACCGCAGTG	✓
dnaK-P2	15	37	ATGAAATTGGGCAG TTGAAA CCAGACGT TTCGCCCC TATTAC AGACTCACAACCACA	✓
dnaQ-P1	15	37	GCCAGCGCTAAAGG TTTTCT CGCGTCCG CGATAGCG TAAAAT AGCGCCGTAACCCC	✓
Fpla-oriTpX	15	38	GAACCACCAACCTG TTGAGC CTTTTTGT GGAGTGGGT TAAATT ATTTACGGATAAAG	✓
Fplas-traM	15	38	ATTAGGGGTGCTGC TAGCGG CGCGGTGT GTTTTTTTA TAGGAT ACCGCTAGGGGCGCTG	✓
Fplas-traY/Z	14	37	GCGTTAATAAGGT GTTAAT AAAATATA GACTTTCCG TCTATT TACCTTTTCTGATTATT	✓
frdABCD	12	34	GATCTCGTCAA ATTTCA GACTTATC GATCAGAC TATACT GTTGTACCTATAAAGGA	✓
fumA	15	38	GTACTAGTCTCAGT TTTTGT TAAAAAAG TGTGTAGGA TATTGT TACTCGCTTTTAACAGG	✓
γ-δ-tnpA	15	38	ACACATTAACAGCA CTGTTT TTATGTGT GCGATAATT TATAAT ATTTCGGACGGTTGCA	✓
γ-δ-tnpR	14	36	ATTCATTAACAAT TTTGCA ACCGTCCG AAATATTA TAAATT ATCGCACACATAAAAAC	✓
gal-P1	15	38	TCCATGTCACACTT TTCGCA TCTTTGTT ATGCTATGG TTATTT CATACCATAAG	✓
gal-P2	15	37	CTAATTTATTCCAT GTCACA CTTTTCGC ATCTTTGT TATGCT ATGGTTATTTCATACC	✓
gal-P2/mut-1	14	36	TAATTTATTCCAT GTCACA CTTTTCGC ATCTTTGT TATACT ATGGTTATTTCATAC	✓
gal-P2/mut-2	14	36	TAATTTATTCCAT GTCACA CTTTTCGC ATTTTTGT TATGCT ATGGTTATTTCATAC	✓
glnL	15	40	CAATTCTCTGATGC TTCGCG CTTTTTATC CGTAAAAAGC TATAAT GCACTAAATGGTGC	✓
gln	15	38	TAAAAAACTAACAG TTGTCA GCCTGTCC CGCTTATAA GATCAT ACGCCGTTATACGTT	✓
gltA-P1	15	37	ATTCATTCGGGACA GTTATT AGTGGTAG ACAAGTTT AATAAT TCGGATTGCTAAGTA	✓
gltA-P2	15	39	AGTTGTTACAAACA TTACCA GGAAAAGCA TATAATGCG TAAAAG TTATGAAGTCGGT	✓
glyA	15	38	TCCTTTGTCAAGAC CTGTTA TCGCACAA TGATTCGGT TATACT GTTCGCCGTTGTCC	✓
glyA/geneX	15	39	ACACCAAAGAACCA TTTACA TTGCAGGGC TATTTTTTA TAAGAT GCATTTGAGATACAT	✓
gnd	15	38	GCATGGATAAGCTA TTTATA CTTTAATA AGTACTTTG TATACT TATTTGCGAACATTCCA	✓
groE	–	–	TTTTTCCCCC TTGAAG GGGCGAAG CCATCCCCA TTTCTC TGGTCACCAGCCGGGAA	□
gyrB	11	38	CGGACGAAAA TTCGAA GATGTTTACCGT GGAAAAGGG TAAAAT AACGGATTAACCCAAGT	✓
his	14	38	ATATAAAAAGTTC TTGCTT TCTAACGTG AAAGTGGTT TAGGTT AAAAGACATCAGTTGAA	✓
hisA	15	38	GATCTACAAACTAA TTAATA AATAGTTA ATTAACGCT CATCAT TGTACAATGAACTGTAC	✓
hisBp	15	38	CCTCCAGTGCGGTG TTTAAA TCTTTGTG GGATCAGGG CATTAT CTTACGTGATCAG	✓
hisJ(St)	15	37	TAGAATGCTTTGCC TTGTCG GCCTGATT AATGGCAC GATAGT CGCATCGGATCTG	✓
hisS	15	38	AAATAATAACGTGA TGGGAA GCGCCTCG CTTCCCGTG TATGAT TGAACCCGCATGGCTC	✓
htpR-P1	15	38	ACATTACGCCACTT ACGCCT GAATAATA AAAGCGTGT TATACT CTTTCCTGCAATGGTT	✓
htpR-P2	15	39	TTCACAAGCTTGCA TTGAAC TTGTGGATA AAATCACGG TCTGAT AAAACAGTGAATG	✓
htpR-P3	15	38	AGCTTGCATTGAAC TTGTGG ATAAAATC ACGGTCTGA TAAAAC AGTGAATGATAACCTCGT	✓
ilvGEDA	15	38	GCCAAAAAATATCT TGTACT ATTTACAA AACCTATGG TAACTC TTTAGGCATTCCTTCGA	✓
ilvIH-P1	14	37	CTCTGGCTGCCAA TTGCTT AAGCAAGA TCGGACGGT TAATGT GTTTTACACATTTTTTC	✓
ilvIH-P2	15	38	GAGGATTTTATCGT TTCTTT TCACCTTT CCTCCTGTT TATTCT TATTACCCCGTGT	✓
ilvIH-P3	14	37	ATTTTAGGATTAA TTAAAA AAATAGAG AAATTGCTG TAAGTT GTGGGATTCAGCCGATT	✓
ilvIH-P4	15	38	TGTAGAATTTTATT CTGAAT GTCTGGGC TCTCTATTT TAGGAT TAATTAAAAAAATAGAG	✓
ISlins-PL	15	37	CGAGGCCGGTGATG CTGCCA ACTTACTG ATTTAGTG TATGAT GGTGTTTTTGAGGTGCT	✓
ISlins-PR	13	36	ATATATACCTTA TGGTAA TGACTCCA ACTTATTGA TAGTGT TTTATGTTCAGATAAT	✓
IS2I-II	7	30	GATGTC TGGAAA TATAGGGG CAAATCCAC TAGTAT TAAGACTATCACTTATT	✓
lacI	15	38	GACACCATCGAATG GCGCAA AACCTTTC GCGGTATGG CATGAT AGCGCCCGGAAGAGAGT	✓

Table 6 (*continued*)

sequence	ttgaca	tataat	promoter	found
lacP1	15	39	TAGGCACCCCAGGC TTTACA CTTTATGCT TCCGGCTCG TATGTT GTGTGGAATTGTGAGC	✓
lacP115	14	37	TTTACACTTTATG CTTCCG GCTCGTAT GTTGTGTGG TATTGT GAGCGGATAACAATTT	✓
lacP2	15	38	AATGTGAGTTAGCT CACTCA TTAGGCAC CCCAGGCTT TACACT TTATGCTTCCGGCTCG	✓
lep	15	37	TCCTCGCCTCAATG TTGTAG TGTAGAAT GCGGCGTT TCTATT AATACAGACGTTAAT	✓
leu	2	25	G TTGACA TCCGTTTT TGTATCCAG TAACTC TAAAAGCATATCGCATT	✓
leultRNA	15	37	TCGATAATTAACTA TTGACG AAAAGCTG AAAACCAC TAGAAT GCGCCTCCGTGGTAGCA	✓
lex	15	38	TGTGCAGTTTATGG TTCCAA AATCGCCT TTTGCTGTA TATACT CACAGCATAACTGTAT	✓
livJ	15	38	TGTCAAAATAGCTA TTCCAA TATCATAA AAATCGGGA TATGTT TTAGCAGAGTATGCT	✓
lpd	7	30	TTGTTG TTTAAA AATTGTTA ACAATTTTG TAAAAT ACCGACGGATAGAACGA	✓
lpp	15	38	CCATCAAAAAAATA TTCTCA ACATAAAA AACTTTGTG TAATAC TTGTAACGCTACATGGA	✓
lppP1	13	37	ATCAAAAAAATA TTCTCA ACATAAAAA ACTTTGTGT TATACT TGTAACGCTACATGGA	✓
lppP2	13	37	ATCAAAAAAATA TTCTCA ACATAAAAA ACTTTGTGT TATAAT TGTAACGCTACATGGA	✓
lppR1	13	36	ATCAAAAAAATA TTCACA ACATAAAA A ACTTTGT GTAATA CTTGTAACGCTACATGGA	✓
Mlrna	15	38	ATGCGCAACGCGGG GTGACA AGGGCGCG CAAACCCTC TATACT GCGCGCCGAAGCTGACC	✓
mac11	14	38	CCCCCGCAGGGAT GAGGAA GGTGGTCGA CCGGGCTCG TATGTT GTGTGGAATTGTGAGC	✓
mac12	14	38	CCCCCGCAGGGAT GAGGAA GGTCGGTCG ACCGGCTCG TATGTT GTGTGGAATTGTGAGC	✓
mac21	14	38	CCCCCGCAGGGAT GAGGAA GGTCGACCT TCCGGCTCG TATGTT GTGTGGAATTGTGAGC	✓
mac3	14	37	CCCCCGCAGGGAT GAGGAA GGTCGGTC GACCGCTCG TATGTT GTGTGGAATTGTGAGCG	✓
mac31	14	37	CCCCCGCAGGGAT GAGGAA GGTCGGTC GACCGCTCG TATATT GTGTGGAATTGTGAGCG	✓
malEFG	15	37	AGGGGCAAGGAGGA TGGAAA GAGGTTGC CGTATAAA GAAACT AGAGTCCGTTTAGGTGT	✓
malK	15	37	CAGGGGGTGGAGGA TTTAAG CCATCTCC TGATGACG CATAGT CAGCCCATCATGAATG	✓
malPQ	15	38	ATCCCCGCAGGATG AGGAAG GTCAACAT CGAGCCTGG CAAACT AGCGATAACGTTGTGT	✓
malPQ/A516P1	12	34	ATCCCCGCAGG ATGAGG AGCCTGGC AAACTAGC GATGAT AACGTTGTGTTGAA	✓
malPQ/A516P2	15	39	ATCCCCGCAGGAGG ATGAGG AGCCTGGCA AACTAGCGA TAACGT TGTGTTGAAAA	✓
malPQ/A517/A	15	37	CCCCGCAGGATGAG GTCGAG CCTGGCAA ACTAGCGA TAACGT TGTGTTGAAAA	✓
malPQ/Pp12	–	–	ATCCCCGCAGGAT GAGGAA GGTCAACA TCGAGCCTG GAAAAC TAGCGATAACGTTGTGT	□
malPQ/Pp13	14	38	ATCCCCGCAGGAT TAGGAA GGTCAACAT CGAGCCTGG CAAACT AGCGATAACGTTGTGT	✓
malPQ/Pp14	14	37	ATCCCCGCAGGAT GAGGAA GGTCAACA TCGAGCCTG GAAACT AGCGATAACGTTGTGT	✓
malPQ/Pp15	14	38	ATCCCCGCAGGAT GAGAAA GGTCAACAT CGAGCCTGG CAAACT AGCGATAACGTTGTGT	✓
malPQ/Pp16	15	38	ATCCCCGCAGGATA AGGAAG GTCAACAT CGAGCCTGG CAAACT AGCGATAACGTTGTGT	✓
malPQ/Pp18	15	38	ATCCCCGCAGGATG GGGAAG GTCAACAT CGAGCCTGG CAAACT AGCGATAACGTTGTGT	✓
malT	15	37	GTCATCGCTTGCAT TAGAAA GGTTTCTG GCCGACCT TATAAC CATTAATTACG	✓
manA	15	38	CGGCTCCAGGTTAC TTCCCG TAGGATTC TTGCTTTAA TAGTGG GATTAATTTCCACATTA	✓
metA-P1	15	38	TTCAACATGCAGGC TCGACA TTGGCAAA TTTTCTGGT TATCTT CAGCTATCTGGATGT	✓
metA-P2	15	38	AAGACTAATTACCA TTTTCT CTCCTTTT AGTCATTCT TATATT CTAACGTAGTCTTTTCC	✓
metBL	12	35	TTACCGTGACA TCGTGT AATGCACC TGTCGGCGT GATAAT GCATATAATTTTAACGG	✓
metF	8	31	TTTTCGG TTGACG CCCTTCGG CTTTTCCTT CATCTT TACATCTGGACG	✓
micF	15	37	GCGGAATGGCGAAA TAAGCA CCTAACAT CAAGCAAT AATAAT TCAAGGTTAAAATCAAT	✓
motA	15	39	GCCCCAATCGCGCG TTAACG CCTGACGAC TGAACATCC TGTCAT GGTCAACAGTGGA	✓
MuPc-1	6	33	AAATT TTGAAA AGTAACTTTATAGAAAAGAAT AATACT GAAAAGTCAATTTGGTG	✓
MuPc-2	9	32	GGAACACA TTTAAA AACCCTCC TAAGTTTTG TAATCT ATAAAGTTAGCAATTTA	✓
MuPe	15	38	TACCAAAAAGCACC TTTACA TTAAGCTT TTCAGTAAT TATCTT TTTAGTAAGCTAGCTA	✓
NR1rnaC	15	39	GTCACAATTCTCAA GTCGCT GATTTCAAA AAACTGTAG TATCCT CTGCGAAACGATCCCT	✓
NR1rnaC/m	15	38	TCACAATTCTCAAG TTGCTG ATTTCAAA AAACTGTAG TATCCT CTGCGAAACGATCCCT	✓
NTP1rna100	11	35	GGAGTTTGTC TTGAAG TTATGCACC TGTTAAGGC TAAACT GAAAGAACAGATTTTGT	✓
nusA	7	30	CAGTAT TTGCAT TTTTTACC CAAAACGAG TAGAAT TTGCCACGTTTCAGGCG	✓
ompA	12	34	GCCTGACGGAG TTCACA CTTGTAAG TTTTCAAC TACGTT GTAGACTTTAC	✓
ompC	15	38	GTATCATATTCGTG TTGGAT TATTCTGC ATTTTTGGG GAGAAT GGACTTGCCGACTG	✓
ompF	7	30	GGTAGG TAGCGA AACGTTAG TTTGAATGG AAAGAT GCCTGCAGACACATAAA	✓
ompF/pKI217	3	26	GG TAGCGA AACGTTAG TTTGCAAGC TTTAAT GCGGTAGTTTATCAC	✓
ompR	15	36	TTTCGCCGAATAAA TTGTAT ACTTAAG CTGCTGTT TAATAT GCTTTGTAACAATTT	✓
p15primer	15	38	ATAAGATGATCTTC TTGAGA TCGTTTTG GTCTGCGCG TAATCT CTTGCTTGAAAACGAAA	✓
p15rnaI	15	39	TAGAGGAGTTAGTC TTGAAG TCATGCGCC GGTTAAGGC TAAACT GAAAGGACAAGTTTTG	✓
P22ant	15	38	TCCAAGTTAGTGTA TTGACA TGATAGAA GCACTCTAC TATATT CTCAATAGGTCCACGG	✓
P22mnt	15	38	CCACCGTGGACCTA TTGAGA ATATAGTA GAGTGCTTC TATCAT GTCAATACACTAACTT	✓
P22PR	15	37	CATCTTAAATAAAC TTGACT AAAGATTC CTTTAGTA GATAAT TTAAGTGTTCTTTAAT	✓
P22PRM	9	32	AAATTATC TACTAA AGGAATCT TTAGTCAAG TTTATT TAAGATGACTTAACTAT	✓
pBR313Htet	12	35	AATTCTCATGT TTGACA GCTTATCA TCGATAAGC TAGCTT TAATGCGGTAGTTTAT	✓
pColViron-P1	15	38	TCACAATTCTCAAG TTGATA ATGAGAAT CATTATTGA CATAAT TGTTATTATTTTAC	✓
pColViron-P2	13	35	TGTTTCAACACC ATGTAT TAATTGTG TTTATTTG TAAAAT TAATTTTCTGACAATAA	✓
pEG3503	6	30	CTGGC TGGACT TCGAATTCA TTAATGCGG TAGTTT ATCACAGTTAA	✓
phiXA	15	38	AATAACCGTCAGGA TTGACA CCCTCCCA ATTGTATGT TTTCAT GCCTCCAAATCTTGGA	✓
phiXB	15	39	GCCAGTTAAATAGC TTGCAA AATACGTGG CCTTATGGT TACAGT ATGCCCATCGCAGTT	✓
phiXD	15	39	TAGAGATTCTCTTG TTGACA TTTTAAAAG AGCGTGGAT TACTAT CTGAGTCCGATGCTGTT	✓

Table 7
Results for the test sequences.

sequence	ttgaca	tataat	promoter	found
lambdac17	15	38	GGTGTATGCATTTA TTTGCA TACATTCA ATCAATTGT TATAAT TGTTATCTAAGGAAAT	✓
lambdacin	15	38	TAGATAACAATTGA TTGAAT GTATGCAA ATAAATGCA TACACT ATAGGTGTGGTTTAAT	✓
lambdaL57	14	37	TGATAAGCAATGC TTTTTT ATAATGCC AACTTAGTA TAAAAT AGCCAACCTGTTCGACA	✓
lambdaPI	15	38	CGGTTTTTTCTTGC GTGTAA TTGCGGAG ACTTTGCGA TGTACT TGACACTTCAGGAGTG	✓
lambdaPL	15	38	TATCTCTGGCGGTG TTGACA TAAATACC ACTGGCGGT GATACT GAGCACATCAGCAGGA	✓
lambdaPo	15	38	TACCTCTGCCGAAG TTGAGT ATTTTTGC TGTATTTGT CATAAT GACTCCTGTTGATAGAT	✓
lambdaPR	15	38	TAACACCGTGCGTG TTGACT ATTTTACC TCTGGCGGT GATAAT GGTTGCATGTACTAAG	✓
lambdaPR'	15	38	TTAACGGCATGATA TTGACT TATTGAAT AAAATTGGG TAAATT TGACTCAACGATGGGTT	✓
lambdaPRE	15	39	GAGCCTCGTTGCGT TTGTTT GCACGAACC ATATGTAAG TATTTC CTTAGATAACAAT	✓
lambdaPRM	15	38	AACACGCACGGTGT TAGATA TTTATCCC TTGCGGTGA TAGATT TAACGTATGAGCACAA	✓
pBR322bla	15	38	TTTTTCTAAATACA TTCAAA TATGTATC CGCTCATGA GACAAT AACCCTGATAAATGCT	✓
pBR322P4	15	42	CATCTGTGCGGTAT TTCACA CCGCATATGGTGCACTCTCAG TACAAT CTGCTCTGATGCCGCAT	✓
pBR322primer	15	38	ATCAAAGGATCTTC TTGAGA TCCTTTTT TTCTGCGCG TAATCT GCTGCTTGCAAACAAAA	✓
pBR322tet	15	38	AAGAATTCTCATGT TTGACA GCTTATCA TCGATAAGC TTTAAT GCGGTAGTTTATCACA	✓
pBRH4-25	4	27	TCG TTTTCA AGAATTCA TTAATGCGG TAGTTT ATCACAGTTAA	✓
pBRP1	15	42	TTCATACACGGTGC CTGACT GCGTTAGCAATTTAACTGTGA TAAACT ACCGCATTAAAGCTTA	✓
pBRRNAI	15	39	GTGCTACAGAGTTC TTGAAG TGGTGGCCT AACTACGGC TACACT AGAAGGACAGTATTTG	✓
pBRtet-10	15	38	AAGAATTCTCATGT TTGACA GCTTATCA TCGATGCGG TAGTTT ATCACAGTTAA	✓
pBRtet-15	15	38	AAGAATTCTCATGT TTGACA GCTTATCA TCGGTAGTT TATCAC AGTTAAATTGC	✓
pBRtet-22	15	39	AAGAATTCTCATGT TTGACA GCTTATCAT CGATCACAG TTAAAT TGCTAACGCAG	✓
pBRtet/TA22	10	33	TTCTCATGT TTGACA GCTTATCA TCGATAAGC TAAATT TTATATAAAATTTAGCT	✓
pBRtet/TA33	10	33	TTCTCATGT TTGACA GCTTATCA TCGATAAGC TAAATT TATATAAAATTTTATAT	✓
pori-I	15	38	CTGTTGTTCAGTTT TTGAGT TGTGTATA ACCCCTCAT TCTGAT CCCAGCTTATACGGT	✓
pori-r	–	–	GATCGCACGATCTG TATACT TATTTGAGT AAATTAACC CACGAT CCCAGCCATTCTTCTGC	□
ppc	–	–	CGATTTCGCAGCAT TTGACG TCACCGCT TTTACGTGG CTTTAT AAAAGACGACGAAAA	□
pSC101oriP1	3	30	TT TTGTAG AGGAGCAAACAGCGTTTGCGA CATCCT TTTGTAATACTGCGGAA	✓
pSC101oriP2	8	30	ATTATCA TTGACT AGCCCATC TCAATTGG TATAGT GATTAAAATCACCTAGA	✓
pSC101oriP3	15	38	ATACGCTCAGATGA TGAACA TCAGTAGG GAAAATGCT TATGGT GTATTAGCTAAAGC	✓
pyrB1-P1	15	37	CTTTCACACTCCGC CCTATA AGTCGGAT GAATGGAA TAAAAT GCATATCTGATTGCGTG	✓
pyrB1-P2	13	36	TTGCATCAAATG CTTGCG CCGCTTCT GACGATGAG TATAAT GCCGGACAATTTGCCGG	✓
pyrD	15	38	TTGCCGCAGGTCAA TTCCCT TTTGGTCC GAACTCGCA CATAAT ACGCCCCCGGTTTG	✓
pyrE-P1	15	38	ATGCCTTGTAAGGA TAGGAA TAACCGCC GGAAGTCCG TATAAT GCGCAGCCACATTTG	✓
pyrE-P2	14	38	GTAGGCGGTCATA CTGCGG ATCATAGAC GTTCCTGTT TATAAA AGGAGAGGTGGAAGG	✓
R100rna3	15	39	GTACCGGCTTACGC CGGGCT TCGGCGGTT TTACTCCTG TATCAT ATGAAACAACAGAG	✓
R100RNAI	15	38	CACAGAAAGAAGTC TTGAAC TTTTCCGG GCATATAAC TATACT CCCCGCATAGCTGAAT	✓
R100RNAII	15	38	ATGGGCTTACATTC TTGAGT GTTCAGAA GATTAGTGC TAGATT ACTGATCGTTTAAGGAA	✓
R1RNAII	15	37	ACTAAAGTAAAGAC TTTACT TTGTGGCG TAGCATGC TAGATT ACTGATCGTTTAAGGAA	✓
recA	15	37	TTTCTACAAAACAC TTGATA CTGTATGA GCATACAG TATAAT TGCTTCAACAGAACAT	✓
rnh	15	38	GTAAGCGGTCATTT ATGTCA GACTTGTC GTTTTACAG TTCGAT TCAATTACAGGA	✓
rn(pRNaseP)	15	38	ATGCGCAACGCGGG GTGACA AGGGCGCG CAAACCCTC TATACT GCGCGCCGAAGCTGACC	✓
rp1J	15	38	TGTAAACTAATGCC TTTACG TGGGCGGT GATTTTGTC TACAAT CTTACCCCCACGTATA	✓
rpmH1p	15	38	GATCCAGGACGATC CTTGCG CTTTACCC ATCAGCCCG TATAAT CCTCCACCCGGCGCG	✓
rpmH2p	15	38	ATAAGGAAAGAGAA TTGACT CCGGAGTG TACAATTAT TACAAT CCGGCCTCTTTAATC	✓
rpmH3p	15	38	AAATTTAATGACCA TAGACA AAAATTGG CTTAATCGA TCTAAT AAAGATCCCAGGACG	✓
rpoA	15	38	TTCGCATATTTTTC TTGCAA AGTTGGGT TGAGCTGGC TAGATT AGCCAGCCAATCTTT	✓
rpoB	15	37	CGACTTAATATACT GCGACA GGACGTCC GTTCTGTG TAAATC GCAATGAAATGGTTTAA	✓
rpoD-Pa	13	36	CGCCCTGTTCCG CAGCTA AAACGCAC GACCATGCG TATACT TATAGGGTTGC	✓
rpoD-Pb	9	33	AGCCAGGT CTGACC ACCGGGCAA CTTTTAGAG CACTAT CGTGGTACAAAT	✓
rpoD-Phs	13	36	ATGCTGCCACCC TTGAAA AACTGTCG ATGTGGGAC GATATA GCAGATAAGAA	✓
rpoD-Phs/min	–	–	CCC TTGAAA AACTGTCGATGTGGGACGATA TAGCAG ATAAGAATATTGCT	□
rrn4.5S	14	37	GGCACGCGATGGG TTGCAA TTAGCCGG GGCAGCAGT GATAAT GCGCCTGCGCGTTGGTT	✓
rrnABP1	15	37	TTTTAAATTTCCTC TTGTCA GGCCGGAA TAACTCCC TATAAT GCGCCACCACTGACACG	✓
rrnABP2	15	37	GCAAAAATAAATGC TTGACT CTGTAGCG GGAAGGCG TATTAT GCACACCCCGCGCCGC	✓
rrnB-P3	14	40	CTATGATAAGGAT TACTCA TCTTATCCTT ATCAAACCGT TAAAAT GGGCGGTGTGAGCTTG	✓
rrnB-P4	15	36	GCGTATCCGGTCAC CTCTCA CCTGACA GTTCGTGG TAAAAT AGCCAACCTGTTCGACA	✓
rrnDEXP2	15	37	CCTGAAATTCAGGG TTGACT CTGAAAGA GGAAAGCG TAATAT ACGCCACCTCGCGACAG	✓
rrnD-P1	15	37	GATCAAAAAAATAC TTGTGC AAAAAATT GGGATCCC TATAAT GCGCCTCCGTTGAGACG	✓
rrnE-P1	15	37	CTGCAATTTTTCTA TTGCGG CCTGCGGA GAACTCCC TATAAT GCGCCTCCATCGACACG	✓
rrnG-P1	15	37	TTTATATTTTTCGC TTGTCA GGCCGGAA TAACTCCC TATAAT GCGCCACCACTGACACG	✓
rrnG-P2	15	37	AAGCAAAGAAATGC TTGACT CTGTAGCG GGAAGGCG TATTAT GCACACCGCCGCGCCG	✓
rrnX1	15	37	ATGCATTTTTCCGC TTGTCT TCCTGAGC CGACTCCC TATAAT GCGCCTCCATCGACACG	✓
RSFprimer	15	38	GGAATAGCTGTTCG TTGACT TGATAGAC CGATTGATT CATCAT CTCATAAATAAAGAA	✓
RSFrnaI	15	39	TAGAGGAGTTTGTC TTGAAG TTATGCACC TGTTAAGGC TAAACT GAAAGAACAGATTTTG	✓
S10	15	37	TACTAGCAATACGC TTGCGT TCGGTGGT TAAGTATG TATAAT GCGCGGGCTTGTCGT	✓

Table 7 (*continued*)

sequence	ttgaca	tataat	promoter	found
sdh-P1	14	37	ATATGTAGGTTAA TTGTAA TGATTTTG TGAACAGCC TATACT GCCGCCAGTCTCCGGAA	✓
sdh-P2	15	37	AGCTTCCGCGATTA TGGGCA GCTTCTTC GTCAAATT TATCAT GTGGGGCATCCTTACCG	✓
spc	15	38	CCGTTTATTTTTTC TACCCA TATCCTTG AAGCGGTGT TATAAT GCCGCGCCCTCGATA	✓
spot42r	15	37	TTACAAAAAGTGCT TTCTGA ACTGAACA AAAAAGAG TAAAGT TAGTCGCGTAGGGTACA	✓
ssb	15	39	TAGTAAAAGCGCTA TTGGTA ATGGTACAA TCGCGCGTT TACACT TATTCAGAACGATTTT	✓
str	15	38	TCGTTGTATATTTC TTGACA CCTTTTCG GCATCGCCC TAAAAT TCGGCGTCCTCATAT	✓
sucAB	15	39	AAATGCAGGAAATC TTTAAA AACTGCCCC TGACACTAA GACAGT TTTAAAAGGTTCCTT	✓
supB-E	15	38	CCTTGAAAAAGAGG TTGACG CTGCAAGG CTCTATACG CATAAT GCGCCCCGCAACGCCGA	✓
T7-A1	15	38	TATCAAAAAGAGTA TTGACT TAAAGTCT AACCTATAG GATACT TACAGCCATCGAGAGGG	✓
T7-A3	15	38	GTGAAACAAAACGG TTGACA ACATGAAG TAAACACGG TACGAT GTACCACATGAAACGAC	✓
T7-C	15	38	CATTGATAAGCAAC TTGACG CAATGTTA ATGGGCTGA TAGTCT TATCTTACAGGTCATC	✓
T7-D	15	38	CTTTAAGATAGGCG TTGACT TGATGGGT CTTTAGGTG TAGGCT TTAGGTGTTGGCTTTA	✓
T7A2	15	39	ACGAAAAAACAGGTA TTGACA ACATGAAGT AACATGCAG TAAGAT ACAAATCGCTAGGTAAC	✓
T7E	11	34	CTTACGGATG ATGATA TTTACACA TTACAGTGA TATACT CAAGGCCACTACAGATA	✓
TAC16	10	32	AATGAGCTG TTGACA ATTAATCA TCGGCTCG TATAAT GTGTGGAATTGTG	✓
Tn10Pin	9	33	TCATTAAG TTAAGG TGGATACAC ATCTTGTCA TATGAT CAAATGGTTTCGCGAAA	✓
Tn10Pout	15	38	AGTGTAATTCGGGG CAGAAT TGGTAAAG AGAGTCGTG TAAAAT ATCGAGTTCGCACATC	✓
Tn10tetA	15	39	ATTCCTAATTTTTG TTGACA CTCTATCAT TGATAGAGT TATTTT ACCACTCCCTATCAGT	✓
Tn10tetR	15	39	TATTCATTTCACTT TTCTCT ATCACTGAT AGGGAGTGG TAAAAT AACTCTATCAATGATA	✓
Tn10tetR*	11	34	TGATAGGGAG TGGTAA AATAACTC TATCAATGA TAGAGT GTCAACAAAAATTAGG	✓
Tn10xxxP1	15	37	TTAAAATTTTCTTG TTGATG ATTTTTAT TTCCATGA TAGATT TAAAATAACATACC	✓
Tn10xxxP2	15	38	AAATGTTCTTAAGA TTGTCA CGACCACA TCATCATGA TACCAT AAACATACTGACGG	✓
Tn10xxxP3	11	38	CCATGATAGA TTTAAA ATAACATACCGTCAGTATGTT TATGGT ATCATGATGATGTGGTC	✓
Tn2660bla-P3	15	38	TTTTTCTAAATACA TTCAAA TATGTATC CGCTCATGA GACAAT AACCCTGATAAATGCT	✓
Tn2661bla-Pa	15	38	GGTTTATAAAATTC TTGAAG ACGAAAGG GCCTCGTGA TACGCT TATTTTTATAGGTTAA	✓
Tn2661bla-Pb	5	28	CCTC GTGATA CGCTTATT TTTATAGGT TAATGT CATGATAATAATGGTTT	✓
Tn501mer	14	39	TTTTCCATATCGC TTGACT CCGTACATG AGTACGGAAG TAAGGT TACGCTATCCAATTTC	✓
Tn501merR	15	37	CATGCGCTTGTCCT TTCGAA TTGAAATT GGATAGCG TAACCT TACTTCCGTACTCA	✓
Tn5TR	15	38	TCCAGGATCTGATC TTCCAT GTGACCTC CTAACATGG TAACGT TCATGATAACTTCTGCT	✓
Tn5neo	15	38	CAAGCGAACCGGAA TTGCCA GCTGGGGC GCCCTCTGG TAAGGT TGGGAAGCCCTGCAA	✓
Tn7-PLE	15	38	ACTAGACAGAATAG TTGTAA ACTGAAAT CAGTCCAGT TATGCT GTGAAAAAGCAT	✓
tnaA	15	37	AAACAATTTCAGAA TAGACA AAAACTCT GAGTGTAA TAATGT AGCCTCGTGTCTTGCG	✓
tonB	15	39	ATCGTCTTGCCTTA TTGAAT ATGATTGCT ATTTGCATT TAAAAT CGAGACCTGGTTT	✓
trfA	15	39	AGCCGCTAAAGTTC TTGACA GCGGAACCA ATGTTTAGC TAAACT AGAGTCTCCTT	✓
trfB	15	38	AGCGGCTAAAGGTG TTGACG TGCGAGAA ATGTTTAGC TAAACT TCTCTCATGTG	✓
trp	15	38	TCTGAAATGAGCTG TTGACA ATTAATCA TCGAACTAG TTAACT AGTACGCAAGTTCACGT	✓
trpP2	15	38	ACCGGAAGAAAACC GTGACA TTTTAACA CGTTTGTTA CAAGGT AAAGGCGACGCCGCCC	✓
trpR	15	39	TGGGGACGTCGTTA CTGATC CGCACGTTT ATGATATGC TATCGT ACTCTTTAGCGAGTACA	✓
trpS	15	38	CGGCGAGGCTATCG ATCTCA GCCAGCCT GATGTAATT TATCAG TCTATAAATGACC	✓
trxA	15	39	CAGCTTACTATTGC TTTACG AAAGCGTAT CCGGTGAAA TAAAGT CAACTAGTTGGTTAA	✓
tufB	15	38	ATGCAATTTTTTAG TTGCAT GAACTCGC ATGTCTCCA TAGAAT GCGCGCTACTTGATGCC	✓
tyrT	15	37	TCTCAACGTAACAC TTTACA GCGGCGCG TCATTTGA TATGAT GCGCCCCGCTTCCCGAT	✓
tyrT/109	15	39	ACAGCGCGTCTTTG TTTACG GTAATCGAA CGATTATTC TTTAAT CGCCAGCAAAAATAA	✓
tyrT/140	–	–	TTAAGTCGTCACTA TACAAA GTACTGGCA CAGCGGGTC TTTGTT TACGGTAATCG	□
tyrT/178	13	34	TGCGCGCAGGTC GTGACG TCGAGAA AAACGTCT TAAGTC GTGCACTATACA	✓
tyrT/212	2	24	C ATGTCG ATCATACC TACACAGC TGAAGA TATGATGCGCGCAGGTCGTGACG	✓
tyrT/6	–	–	ATTTTTCTCAAC GTAACA CTTTACAG GCGCGTCA TTTGAT ATGATGCGCCCCGCTTC	□
tyrT/77	13	38	ATTATTCTTTAA TCGCCA GCAAAAATA ACTGGTTACC TTTAAT CCGTTACGGATGAAAAT	✓
uncI	15	37	TGGCTACTTATTGT TTGAAA TCACGGGG GCGCACCG TATAAT TTGACCGCTTTTTGAT	✓
uvrB-P1	15	38	TCCAGTATAATTTG TTGGCA TAATTAAG TACGACGAG TAAAAT TACATACCTGCCCGC	✓
uvrB-P2	15	39	TCAGAAATATTATG GTGATG AACTGTTTT TTTATCCAG TATAAT TTGTTGGCATAATTAA	✓
uvrB-P3	15	38	ACAGTTATCCACTA TTCCTG TGGATAAC CATGTGTAT TAGAGT TAGAAAACACGAGGCA	✓
uvrC	15	38	GCCCATTTGCCAGT TTGTCT GAACGTGA ATTGCAGAT TATGCT GATGATCACCAAGG	✓
uvrD	15	37	TGGAAATTTCCCGC TTGGCA TCTCTGAC CTCGCTGA TATAAT CAGCAAATCTGTATAT	✓
434PR	15	38	AAGAAAAACTGTAT TTGACA AACAAGAT ACATTGTAT GAAAAT ACAAGAAAGTTTGTTGA	✓
434PRM	15	38	ACAATGTATCTTGT TTGTCA AATACAGT TTTTCTTGT GAAGAT TGGGGGTAAATAACAGA	✓

Acknowledgements

This work was partially supported by the Spanish Ministerio de Ciencia y Tecnologia under project TIC2003-00877 (including FEDER fundings).

References

[1] T.L. Bailey, C. Elkan, The value of prior knowledge in discovering motifs with MEME, in: Proc. Internat. Conf. on Intelligent Systems & Molecular Biology, Vol. 3, 1995, pp. 21–29.
[2] P. Baldi, S. Brunak, Bioinformatics: The Machine Learning Approach, MIT Press, Cambridge, MA, 1998.
[3] S. Brenner, Genomics: the end of the beginning, Science 287 (5461) (2000) 2173–2179.
[4] C. Coello Coello, D. Van Veldhuizen, G. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York, 2002.
[5] J. Collado-Vides, B. Magasanik, J.D. Gralla, Control site location and transcriptional regulation in *Escherichia coli*, Microbiol. Rev. 55 (3) (1991) 371–394.
[6] O. Cordón, F. Herrera, I. Zwir, A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology, Fuzzy Sets and Systems 138 (2) (2003) 307–341.
[7] V. Cotik, Una propuesta conexionista para el reconocimiento y predicción de promotores en secuencias de ADN de procariotas, Master's Thesis, Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2004.
[8] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, New York, 2001.
[9] B. Demeler, G. Zhou, Neural network optimization for *E. coli* promoter prediction, Nucl. Acids Res. 19 (1991) 1593–1599.
[10] M. Gibson, E. Mjolsness, Computational Modeling of Genetic and Biochemical Networks, Ch. Modeling the Activity of Single Genes, The MIT Press, Cambridge, MA, 2001.
[11] D.E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989, URL: citeseer.nj.nec.com/goldberg89genetic.html
[12] D.E. Goldberg, A meditation on the computational intelligence and its future, Technical Report 2000019, Department of General Engineering, University of Illinois at Urbana-Champaign, 2000.
[13] A. Gorm Pedersen, J. Engelbrecht, Investigations of *Escherichia coli* promoter sequences with artificial neural networks: new signals discovered upstream of the transcriptional startpoint, in: Proc. Third Internat. Conf. on Intelligent Systems for Molecular Biology, Vol. 3, 1995, pp. 292–299.
[14] C.B. Harley, R.P. Reynolds, Analysis of *E. coli* promoter sequences, Nucl. Acids Res. 15 (5) (1987) 2343–2361.
[15] D.K. Hawley, R. McClure, Compilation and analysis of *Escherichia coli* promoter DNA sequences, Nucl. Acids Res. 11 (8) (1983) 2237–2255.
[16] G.Z. Hertz, G.D. Stormo, Identifying DNA and protein patterns with statistically significant alignments of multiple sequences, Bioinformatics 15(7/8) (1999) 563–577, URL: http://bioinformatics.oupjournals.org/cgi/reprint/15/7/563.pdf
[17] J.D. Hirst, M.J.E. Sternberg, Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks, Biochemistry 31 (32) (1992) 7211–7218.
[18] A.M. Huerta, J. Collado-Vides, Sigma70 promoters in *Escherichia coli*: specific transcription in dense regions of overlapping promoter-like signals, J. Mol. Biol. 333 (2) (2003) 261–278.
[19] A. Ishihama, Protein–protein communication within the transcription apparatus, J. Bacteriol. 175 (1993) 2483–2489.
[20] G.J. Klir, T.A. Folger, Fuzzy sets, uncertainty, and information, Prentice-Hall International, Englewood Cliffs, NJ, 1988.
[21] M. Laguna, R. Marti, Scatter Search: Methodology and Implementations in C, Kluwer Academic Publishers, Boston, 2003.
[22] K.J. Lang, A.H. Waibel, A time-delay neural network architecture for isolated word recognition, Neural Networks 3 (1990) 23–43.
[23] C.E. Lawrence, et al., Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, Science 262 (5131) (1993) 208–214.
[24] S. Lisser, H. Margalit, Compilation of *E. coli* mRNA promoter sequences, Nucl. Acids Res. 21 (7) (1993) 1507–1516.
[25] N. Mache, M. Reczko, A. Hatzigeorgiou, Multistate Time-Delay Neural Networks for the recognition of Pol II promoter sequences, unpublished, URL: http://www.informatik.unistuttgart.de/ipvr/bv/personen/mache/ismb/ismb.html
[26] B.W. Matthews, Comparison of the predicted and observed secondary structure of T4 Phage Lysozyme, Biochim. Biophys. Acta 405 (1975) 442–451.
[27] T.M. Mitchell, Machine learning, McGraw-Hill, New York, 1997.
[28] C. Mouslim, T. Latifi, E.A. Groisman, Signal-dependent requirement for the co-activator protein RcsA in transcription of the RcsB-regulated ugd gene, J. Biol. Chem. 278 (50) (2003) 50588–50595.
[29] W. Pedrycz, P.P. Bonissone, E.H. Ruspini, Handbook of Fuzzy Computation, Institute of Physics, 1998.

[30] S.R. Presnell, F.E. Cohen, Artificial Neural Networks for pattern recognition in biochemical sequences, Annu. Rev. Biophys. Biomol. Struct. 22 (1993) 283–298.
[31] M. Ptashne, A. Gann, Genes and signals, Cold Spring Harbor Laboratory Press, 2002.
[32] M.G. Reese, Application of a time-delay neural network to promoter annotation in the Drosophila Melanogaster genome, Comput. Chem. 26 (1) (2002) 51–56.
[33] E.H. Ruspini, I. Zwir, Automated generation of qualitative representations of complex object by hybrid soft-computing methods, in: S.K. Pal, A. Pal (Eds.), Lecture Notes in Pattern Recognition, World Scientific Company, Singapore, 2001.
[34] H.E.A. Salgado, Regulondb (version 3.2): transcriptional regulation and operon organization in *Escherichia coli* K-12, Nucl. Acids Res. 29 (2001) 72–74.
[35] R. Sarker, K. Liang, C. Newton, A new multiobjective evolutionary algorithm, Eur. J. Oper. Res. 140 (2002) 12–23.
[36] M. Sugeno, A fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems 1 (1) (1993) 7–31.
[37] J. Thompson, T. Gibson, F. Plewniak, F. Jeanmougin, D. Higgins, The CLUSTAL_*X* windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools, Nucl. Acids Res. 25 (24) (1997) 4876–4882.
[38] A. Waibel, et al., Phoneme recognition using time-delay neural networks, IEEE Trans. Acoust. Speech Signal Process. 37 (3) (1989) 328–339.
[39] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, Los Altos, CA, 1999.
[40] C.H. Wu, J.W. McLarthy, Neural Networks and Genome Informatics, Elsevier, Amsterdam, 2000.
[41] E. Zitzler, L. Thiele, An evolutionary algorithm for multiobjective optimization: the strength Pareto approach, Technical Report 43, Computer Engineering and Communication Networks Lab TIK, Swiss Federal Institute of Technology ETH, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998, URL: citeseer.nj.nec.com/article/zitzler98evolutionary.html
[42] I. Zwir, R. Romero Zaliz, E.H. Ruspini, Automated biological sequence description by genetic multiobjective generalized clustering, Ann. New York Acad. Sci. 980 (2002) 65–82.