CHAPTER 1

GENERALIZED ANALYSIS OF PROMOTERS (GAP): A METHOD FOR DNA SEQUENCE DESCRIPTION

R. Romero Zaliz[a], I. Zwir[b] and E. Ruspini[c]

[a] *Department of Computer Science*
Facultad de CienciasExactas y Naturales
Universidad of Buenos Aires
Buenos Aires, Argentina
E-mail: rromero@dc.uba.ar

[b] *Department of Molecular Microbiology*
Washington University School of Medicine
St. Louis, Missouri, U.S.A.
Email:zwir@borcim.wustl.edu [a]

[c] *Artificial Intelligence Center*
SRI International
Menlo Park, California, U.S.A.
E-mail: ruspini@ai.sri.com

Recent advances in the accessibility of databases containing representations of complex objects—exemplified by repositories of time-series data, information about biological macromolecules, or knowledge about metabolic pathways—have not been matched by availability of tools that facilitate the retrieval of objects of particular interest while aiding to understand their structure and relations. In applications such as the analysis of DNA sequences, on the other hand, requirements to retrieve objects on the basic of qualitative characteristics are poorly met by descriptions that emphasize precision and detail rather than structural features.

This paper presents a method for identification of interesting qualitative features in biological sequences. Our approach relies on a generalized clustering methodology, where the features being sought correspond to the solutions of a multivariable, multiobjective optimization problem and generally correspond to fuzzy subsets of the object being represented. Foremost among the optimization objectives being considered are measures of the degree by which features resemble prototypical

[a] Corresponding author.

structures deemed to be interesting by database users. Other objectives include feature distance and, in some cases, performance criteria related to domain-specific constraints.

Genetic-algorithm methods are employed to solve the multiobjective optimization problem. These optimization algorithms discover candidate features as subsets of the object being described that lie in the set of all Pareto-optimal solutions—of that problem. These candidate features are then inter-related employing domain-specific relations of interest to the end users.

We present results of the application of a method termed Generalized Analysis of Promoter (GAP) to identify one of the most important factors involved in the gene regulation problem in bacteria, which is crucial for detecting regulatory behaviors or genetic pathways as well as gene transcription: the RNA polymerase motif. The RNA polymerase or promoter motif presents vague submotifs linked by different distances, thus, making its recognition in DNA sequences difficult. Moreover, multiple promoter motifs can be present in the same regulatory regions and all of them can be potential candidates until experimental mutagenesis is performed. GAP is available for public use in http://soar-tools.wustl.edu.

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[4]. Gene expression is determined by protein-protein interactions among regulatory proteins and with RNA polymerase, and protein-DNA interactions of these trans-acting factors with cis-acting DNA sequences in the promoters of regulated genes [22,11]. Therefore, identifying these protein-DNA interactions, by means of those DNA motifs that characterize the regulatory factors that operate in the transcription of a gene[1,23], becomes crucial for determining which genes participate in a regulation process, how they behave and how are they connected to build genetic networks. The RNA polymerase or promoter is an enzyme that transcribes a gene or recruits other regulatory factors to interact with it, producing cooperative regulations [22]. Different computational methods have been applied to discover promoter motifs or patterns [5,14,16,13,1]. However, most of them failed to provide accurate predictions in prokaryotic promoters because of the variability of the pattern, which comprises more than one vague submotif and variable distances between them. Moreover, multiple occurrences of promoters in the same regulatory region of one gene can be found (e.g. different promoters can be used for gene activation and repression, or can interact with different regulatory factors from the same regulatory pathway [19,7]).

This paper presents a method termed Generalized Analysis of Promoters (GAP), which applies generalized clustering techniques [29,35] to the discovery of qualitative features in complex biological sequences, particularly multiple promoters in bacterial genomes. The motivation for the development of this methodology is provided by requirements to search and interpret databases containing representations of this type of objects in terms that are close to the needs and experience of the users of those data-based descriptions. These qualitative features include both interesting substructures and interesting relations between those structures, where the notion of interestingness is provided by domain experts by means of abstract qualitative models or learned from available databases. The GAP method represents promoter features as fuzzy logic expressions with fuzzy predicates, whose membership functions are learned from probabilistic distributions[30,21,36]. The proposed method takes adventage of a new developed Multi-Objective Scatter Search (MOSS) algorithm to identify multiple promoters occurrences within genomic regulatory regions by optimizing multiple criteria that those features that describe promoters should satisfy. This methodology formalizes previous attempts to produce exhaustive searches of promoters[1], most of which emphasize the processing of detailed system measurements rather than that of qualitative features of direct meaning to users (called *perceptions* by Zadeh) [32].

Therefore, this chapter is organized as follows: Section 2 describes the generalized clustering framework; Section 3 explines the problem ofdiscoverying and describing bacterial promoters; Section 4 applies the GAP method to the promoter discovery problem in Escherichia coli (*E. coli*) genome; Section 5, shows the results obtained by the proposed method and its evaluation; and Section 6 summarizes the concluding remarks.

2. Generalized Clustering

The method presented in this paper belong to a family of techniques for the discovery of interesting structures in datasets by classification of its points into a finite number of fuzzy subsets, or *fuzzy clustering*. Fuzzy clustering methods were introduced by Ruspini[27] to provide a richer representation scheme, based on a flexible notion of partition, for the summarization of dataset structure, and to take advantage of the ability of continuous-analysis techniques to express and treat classification problems in a formal manner.

In Ruspini's original formulation the clustering problem was formulated as a continuous-variable optimization problem over the space of fuzzy partitions of the dataset. This original formulation of the clustering problem as an optimization problem has been largely retained in various extensions of the approach, which differ primarily on the nature of the functionals being optimized and on the constraints that the partition must satisfy[3].

The original approach proposed by Ruspini, however, focused on the determination of the clustering as a whole, i.e., a family of fuzzy subsets of the dataset providing a disjoint, exhaustive partition of the set into interesting structures. Recent developments, however, have emphasized the determination of individual clusters as fuzzy subsets having certain optimal properties. From this perspective, a fuzzy clustering is a collection of optimal fuzzy clusters—that is, each cluster is optimal in some sense and the partition satisfies certain conditions—rather than an optimal partition—that is, the partition, as a whole, is optimal in the sense that it minimizes some predefined functional defining classification quality. Redirecting the focus of the clustering process to the isolation of individual subsets having certain desirable properties provides also a better foundation for the direct characterization of interesting structure while freeing the clustering process from the requirement that clusters be disjoint and that partitions be exhaustive.

In the context of image-processing applications, for example, features may correspond to certain interesting prototypical shapes. In these applications not every image element may belong to an interesting feature while some points might belong to more than one cluster (e.g., the intersection of two linear structures). It was, indeed, n the context of image-processing applications that Krishnapuram and Keller[6] reformulated the fuzzy clustering problem so as to permit the sequential isolation of clusters. This methodology, called *possibilistic clustering*, does not rely, like previous approaches, on prior knowledge about the number of clusters while permitting to take full advantage of clustering methods based on the idea of *prototype*.

Prototype-based classification methods[3] are based on the idea that a dataset could be represented, in a compact manner, by a number of prototypical points. The well-known *fuzzy c-means* method of Bezdek—the earliest fuzzy-clustering approach exploiting this idea—seeks to describe a dataset by a number of prototypical points lying in the same domain as the members of that dataset. Extensions of this basic idea based on generalization of the notion of prototypical structure in a variety of ways (e.g., as line or curve segments in some euclidean space) are the basis for methods that seek to represent datasets in terms of structures that have been predefined as being of particular interest to those seeking to understand the underlying physical systems being studied. Generally speaking, however, these methods require that prototypical structures belong to certain restricted families of objects so as to exploit their structural properties (e.g., the linear structure of line segments or hyperplane patches).

The generalized clustering methodology presented in this paper belongs to this type of approaches, extending them by consideration of arbitrary definitions of interesting structures provided by users by users by means of a family of parameterized models $M = [M_\alpha]$ and a set of relations between them [28,35]. In addition to a variety of geometric structures, these models may also be described by means of structures (e.g., neural networks) learned from significant examples of the features being defined or in terms of very general constraints that features might satisfy to some degree (*soft* or *fuzzy* constraints). As is the case with possibilistic clustering methods, our approach is based on the formulation of the qualitative-feature identification problem in terms of the optimization of a continuous functional $Q(F, M_\alpha)$ that measures the degree of matching between a fuzzy subset F of the dataset and some instantiation M_α of the family of interesting models[29].

Our approach recognizes, however, that simple reliance on optimization of a *single* performance index Q would typically result in the generation of a large number of features with small extent and poor generalization as it is usually easier to match smaller subsets of the dataset than significant portions of it. For this reason, it is also necessary to consider, in addition to measures Q of representation quality, additional criteria S gauging the size of the structure being represented. In addition, it may also be necessary to consider also application-specific criteria introduced to assure that the resulting features are valid and meaningful (e.g., constraints preventing selective picking of sample points so that they lie, for example, close to a line in sample space).

This multiobjective problem might be treated by aggregation of the multiple measures of feature desirability into a global measure of cluster quality [28]. A problem with this type of approach, which is close in spirit to minimum description length methods[26], is the requirement to provide a-priori relative weights to each one of the objectives being aggregated. It should be clear that assignment of larger weight to measures Q of quality representation would lead to small features with higher degrees of matching to models in the prototype families while, conversely, assigning higher weights to measures S of cluster extent would tend to produce larger clusters albeit with poor modeling ability. Ideally, a family of optimization problems, each similar in character to the others but with different weights assigned to each of the aggregated objectives, should be solved so as to produce a full spectrum of candidate clusters.

Rather than following such a path—involving the solution of multiple problems—our approach relies, instead, on a reformulation of the generalized clustering problem as a multiobjective optimization problem involving several measures of cluster desirability[29]. In this formulation, subsets of the dataset of potential interest are *locally optimal* in the *Pareto sense*, i.e., they are *locally nondominated* solutions of the optimization problem.[b]. Locally nondominated solutions of a multiobjective optimization problem are those points in feature space such that their neighbors do not have better objective values for all objectives while being strictly superior in at least one of them. (i.e., a better value, for a neighbor, of some objective implies a lower value of another). The set of these solutions is called the *local Pareto-optimal* or *local effective frontier*. We employ a multiobjective genetic algorithm (MGA)[29] based on an extension of methods originally proposed by Marti and Laguna [18,12] to solve this problem. This method is particularly an attractive tools to solve such complex optimization problems because of their generality and their ability, stemming from application of *multimodal optimization* procedures, to isolate local optima.

3. Problem: Discovering Promoters in DNA Sequences

Biological sequences, such as DNA or protein sequences, are a good example of the type of complex objects that maybe described in terms of meaningful structural patterns. Availability of tools to discover these structures and to annotate the sequences on the basis of those discoveries would greatly improve the usefulness of these repositories that currently rely on methods developed on the basis of computational efficiency and representation accuracy rather than on terms of structural and functional properties deemed to be important by molecular biologists.

An important example of biological sequences are prokaryotic promoter data gathered and analyzed by many compilations [8,5,17] that reveal the presence of two well conserved sequences or submotifs separated by variable distances and a less conserved sequence. The variability of the distance

[b] The notions of proximity and neighborhood in feature space is application dependent

between submotifs and their fuzziness, in the sense that they present several mismatches, hinder the existence of a clear model of prokaryotic core-promoters. The most representative promoters in *E. coli* (i.e. σ^{70}subunits) are described by the following conserved patterns:

(1) *TTGACA:* This pattern is an hexanucleotide conserved sequence whose middle nucleotide is located approximately 35 pair of bases upstream of the transcription start site. The consensus sequence for this pattern is `TTGACA` and the nucleotides reported in [17] reveal the following nucleotide distribution: $T_{69}T_{79}G_{61}A_{56}C_{54}A_{54}$, where for instance the first `T` is the most seen nucleotide in the first position of the pattern and is present in 69 % of the cases. This pattern is often called *-35 region*.
(2) *TATAAT:* This pattern is also an hexanucleotide conserved sequence, whose middle nucleotide is located approximately 10 pair of bases upstream of the transcription start site. The consensus sequence is `TATAAT` and the nucleotide distribution in this pattern is $T_{77}A_{76}T_{60}A_{61}A_{56}T_{82}$, which is often called *-10 region*[17].
(3) *CAP Signal:* In general, a pyrimidine (`C` or `T`) followed by a purine (`A` or `G`) compose the CAP Signal. This signal constitutes the transcription start site (TSS) of a gene.
(4) *Distance(TTGACA, TATAAT).* The distance between the TTGACA and TATAAT consensus submotifs follows a data distribution between 15 and 21 pair of bases. This distance is critical in holding the two sites at the appropriate distance for the geometry of RNA polymerase [8].

The identification of the former RNA polymerase or promoters sites becomes crucial to detect gene activation or repression, by the way in which such promoters interact with different regulatory proteins (e.g. overlapping suggest repression and distances of approximately 40 base pairs suggest typical activation). Moreover, combining the promoter sites with other regulatory sites [37] can reveal different types of regulation, harboring RNA polymerase alone, RNA polymerase recruiting other regulatory protein, or cooperative regulations among more than one regulator[22]. Different methods have been used to identify promoters [9,16,13,5], but several failed to perform accurate predictions because of their lack of flexibility, by using crisp instead of fuzzy models for the submotifs (e.g., TATAAT or TTGACA [24]), or restricting distances between submotifs to fixed values (e.g., 17 base pairs[1]). The vagueness of the compound promoter motifs and the uncertainty of identifying which of those predicted sites correspond to a functional promoter can be completely solved only by performing mutagenesis experiments[22]. Thus more accurate and interpretable predictions would be useful in order to reduce the experiment costs and ease the researchers work.

4. Biological Sequence Description Methods

In this paper we present results of the application of GAP to the discovery of interesting qualitative features in DNA sequences based inthose ideas discussed in Section 2. The notion of interesting feature is formally defined by means of a family of parameterized models $M = \{M_\alpha\}$ specified by domain experts[29] who are interested in finding patterns such as epoch descriptors of individual or multiple DNA sequences. These idealized versions of prototypical models are the basis for a characterization of clusters as cohesive sets that is more general than their customary interpretation as "subsets of close points." To address the promoter prediction problem we take advantage of the ability of representing imprecise and incomplete motifs, the fuzzy sets representations flexibility and interpretability, and the multi-objective genetic algorithms ability to obtain optimal solutions using different criteria.

Our proposed method GAP represents each promoter submotif (i.e., -10 and -35 regions and the distance that separates them) as fuzzy models, whose membership functions are learned from data distributions[15,21]. In addition, as a generalized clustering method, GAP considers the quality of matching with each promoter submotif model (Q), as well as the size of the promoter extend (S), by means of the distance between submotifs, as the *multiple objectives* to be optimized. To do so, we used a Multi-objective Scatter Search (MOSS) optimitation algorithm [18,12], which obtains a set of multiple and optimal promoter descriptions for each promoter region. Moreover, the former matching is also considered by MOSS as a *multimodal* problem, since there is more than one solution for each region. GAP, by using MOSS, overcomes other methods used for DNA motif discovery, such as Consensus/Patser based on weight probabilistic matrices (see Section 5), and provides the desired trade-off between accurate and interpretable solutions, which becomes particurary desirable for the end users. The extension of the original *Scatter Search (SS)* heuristic [18] uses the DNA regions where promoters should be detected as inputs and finds all optimal relationships among promoter submotifs and distance models. In order to extend the original SS algorithm to a multi-objective environment we need to introduce some concepts[10,25]:

A multi-objective optimization problem is defined as:

$$\left.\begin{array}{rll} \mathit{Maximize}\ Q_m(x, M_\alpha), & & m = 1, 2, \ldots, |M|; \\ \mathit{subject\ to}\ g_j(x) \geq 0, & & j_g = 1, 2, \ldots, J; \\ h_k(x) = 0, & & k = 1, 2, \ldots, K; \\ x_i^{(L)} \leq x_i \leq x_i^{(U)}, & & i = 1, 2, \ldots, n. \end{array}\right\}$$

where M_α is a generalized clustering model, $|M|$ corresponds to the number of models and Q_m the objectives to optimize, J to the number of inequality constraints, K to the number of equality constraints and finally 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, we consider the following instantiations:

- $|M| = 3$. We have three models: M_α^1 and M_α^2 are the models for each of the boxes,TTGACA-box and TATAAT-box, respectively, and M_α^3 corresponds to the distance between these two boxes (recall Equations 1 and 2, and Figure 1).
- $|Q| = 3$. We have three objectives consisting of maximizing the degree of matching to the fuzzy models (fuzzy membership): $Q_1(x, M_\alpha^1)$, $Q_2(x, M_\alpha^2)$ and $Q_3(x, M_\alpha^3)$
- $J = 1$. We have just one constraint g_1: the distance between boxes can not be less than 15 and no more than 21 pair of bases.
- $K = 0$. No equality constraints needed.
- Only valid solutions are kept in each generation.
- The boxes can not be located outside the sequence searched, that is, it can not start at negative positions or grater than the length of the query sequence.

Definition 1: 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: $f_i(x) \ntriangleright f_i(y)$ 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 *nondominated* by y.

In order to code the algorithm, three different models were developed. Both submotif models were implemented by using their nucleotide consensus frequency as discrete fuzzy sets, whose membership function has been learned from distributions[15] The first model corresponding to the TATAAT-box was formulated as:

$$M_{\alpha}^{1} = \mu_{tataat}(x) = \mu_{1}^{1}(x_1) \cup ... \cup \mu_{6}^{1}(x) \qquad (1)$$

where the fuzzy discrete set corresponding to the first nucleotide of the submotif $T_{0.77}A_{0.76}T_{0.60}A_{0.61}A_{0.56}T_{0.82}$ was defined as $\mu_{1}^{1}(x_1) = A/0.08 + T/0.77 + G/0.12 + C/0.05$, and the other fuzzy sets corresponding to positions 2-6 were calculated in a similar way accordingly to data distributions from[17]. The second model corresponding to the TTGACA-box was described as:

$$M_{\alpha}^{2} = \mu_{ttgaca}(x) = \mu_{1}^{2}(x_1) \cup ... \cup \mu_{6}^{2}(x) \qquad (2)$$

where the fuzzy crisp set corresponding to the first nucleotide of the submotif $T_{0.69}T_{0.79}G_{0.61}A_{0.56}C_{0.54}A_{0.54}$ was defined as $\mu_{1}^{2}(x) = A/0.12 + T/0.69 + G/0.13 + C/0.06$ and the other fuzzy sets corresponding to positions 2-6 were calculated in a similar way accordingly to data distributions from[17]. The union operation corresponds to fuzzy set operations[21,15]. The third model, i.e., the distance between the previous submotifs, was built as a fuzzy set, whose triangular membership function M_{α}^{3} (see Figure 1) was learned from data distributions[5] centered in 17, where the best value (one) is achieved. Therefore, the objective functions Q_m correspond to the membership to the former fuzzy models M_{α}.

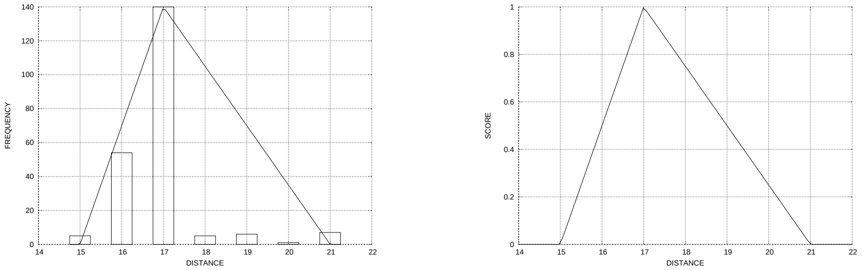

Fig. 1. Graphical representation of M_{α}^{3}

Combination Operator and Local Search. We used a block representation to code each individual, where each block corresponds to one of the promoter submotifs (i.e., TATAAT-box or TTGACA-box). Particularly, each block was represented by two integers, where the first number corresponds to the starting point of the submotif, and the second one represents the size of the box (see Figure 2). The combination process was implemented

Phenotype

```
     ttgaca                 tataat
gtttatttaatgtttacccccataaccacataatcgcgttacact
     ↑                      ↑
char 6                 char 29
```

Genotype

Gen 0	Gen 1	
[(6,6)]	[(29,6)]	
$f_1 = 0.578595$	$f_2 = 0.800000$	$f_3 = 1.000000$

Fig. 2. Example of the representation of an individual

as a one-point combine operator, where the point is always located between both blocks. 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* was implemented as a search for nondominated solutions in a certain neighborhood. For example, a local search performed over the chromosome space involves 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 parent 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 nondominated population found so far, it replaces its father only if it is located in a less crowded region (see Figure 3).

Algorithm. We modified the original SS algorithm to allow multiple-objective solutions by adding the *nondominance* criterion to the solution ranking[10]. Thus, nondominated solutions were added to the set in any order, but dominated solutions were only added if no more nondominated solutions could be found. In addition to maintaining a good set of nondominated solutions, and to avoid one of the most common problems of multi-objective algorithms such as multi-modality[10], we also kept track of the diversity of the available solutions through all generations. Finally, the initial populations were created randomly and unfeasible solutions corresponding to out of distance ranges between promoter submotifs (g_1) were checked at each generation. Figure 4 clearly illustrates the MOSS algorithm proposed in GAP.

```
Randomly select which block $g$ in the representation of the individual $c$ to apply local search.
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 $c'$.
if $c'$ meets the restrictions then
  if $c'$ dominates $c$ then
    Replace $c$ with $c'$
  end if
  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
    Replace $c$ with $c'$ if $crowd(c') < crowd(c)$.
  end if
end if
```

Fig. 3. Local search

```
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.
Create a reference set $RefSet$ with $b/2$ nondominated solutions of $P$ and $b/2$ solutions of $P$ more diverse from the other $b/2$. If there are not enough nondominated solutions to fill the $b/2$, complete the set with dominated solutions.
$NewSolution \leftarrow$ true
while Exists a Solution not yet explored (NewSolution = true) do
  $NewSolution \leftarrow$ false
  Generate subsets of $RefSet$ where there is at least one nondominated solution in each one.
  Generate an empty subset $N$ to store nondominated solutions.
  while subset to examine do
    Select a subset and mark it as examined.
    Apply combination operators to the solutions in the set.
    Apply local search to each new solution $x$ found after the combination process as explained in Figure 3 and name it $x^b$.
    if $x^b$ is nondominated by any $x \in N$ and $x^b \notin N$ then
      Add $x^b$ to $N$.
    end if
  end while
  Add solutions $y \in N$ to $P$ if there are no solution $z \in P$ that dominates $y$.
  $NewSolution \leftarrow$ true.
end while
```

Fig. 4. MOSS algorithm

5. Experimental Algorithm Evaluation

The GAP method was applied to a set of known promoter sequences reported in[5]. In this work 261 promoter regions and 68 the alternative solutions (multiple promoters) defined in[5] for the corresponding sequences (totalizing 329 regions) constituted the input of the method.

To evaluate the performance of GAP, we first compare the obtained results with the ones retrived by a typical DNA sequence analysis method, the Consensus/Patser [14]. Then, we compare the ability of MOSS with the other two Multiobjective Evolutionary Algorithms (MOEAs), i.e., the Strength Pareto Evolutionary Algorithm (SPEA)[33] and the $(\mu + \lambda)$ Multi-Objective Evolutionary Algorithm (MuLambda)[20].

All of the former MOEA algorithms share the same following properties:

- They store optimal solutions found during the search in an external set.
- They work with the concept of Pareto dominance to assign fitness values to the individuals of the population.

Particularly, SPEA is a well known algorithm that have some special features [33], including:

- The combination of above techniques in a single algorithm.
- The determination of the fitness value of an individual by using the solutions stored in the external population, where dominance from the current population becomes irrelevant.
- All individuals of the external set participate in the selection procedure.
- A niching method is given to preserve diversity in the population. This method is based on Pareto optimality and does not require a distance parameter (e.g., the niche ratio in a *sharing* function[10]).

MuLambda is a relative new algorithm with a very different design from other Pareto approaches. This algorithm has the following characteristics[20]:

- It does not use any information from the dominated individuals of the population. Only nondominated individuals are kept from generation to generation.
- The population size is variable.
- It makes clustering to reduce the number of nondominated solutions stored without destroying the features of the optimal Pareto front.

As we explained earlier, the MOSS approach has the following properties:

- The local search is used to improve those solutions found during the execution of the algorithm.
- The diversity of the solutions is kept by including in every generation a set of diverse solutions into the current population.

To compare the results obtained from the former three algorithms, we use the same objective functions described in Section 4 and execute these algorithms 20 times with different seeds for each input sequence. A promoter is said to be found if it appears in, at least, one of the execution result sets. The parameters used in the experiments are listed in Table 1.

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

Table 1. Parameters for algorithms

Our method overcomes Consensus/Patser[14] by detecting te 93.1 % of the available promoters, while this method, based on weight matrices, identify the 74 %. Moreover, GAP, by using MOSS also overcomes the other MOEA algorithms as it is illustrated in Table 2.

	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%

Table 2. Results with different Multi-Objective Genetic Algorithms for all sequences. The *Original* column indicates the number of conserved promoter locations reported in the literature. The *Alternative* column indicates alternative locations also reported in the literature

We should note that there exist more than one possible description for each promoter region, as it is illustrated in Figure 5 for the *Ada* gene reported in Harley & Reynolds compilation[5]. These alternative descriptions were also found by MOSS in a higher percentage than the other methods (86.76 %). The complete set of results is illustrated in the Appendix.

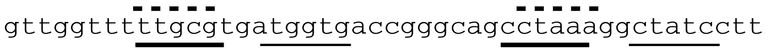

Fig. 5. Different solutions for the *Ada* sequence - Three different alternative locations for the preserved sequences were included in the final set of the MOSS method matching with the three alternatives reported in the literature

In addition to the number of promoters detected by using different MOEA algorithms, we use two other functions C^{34} and D (see Equations 3 and 4) to have a better understanding of each algorithm performance.

Definition 2: Let $X', X'' \subseteq X$ two set of decision vectors. The function C maps the ordered pairs (X', X'') to the $[0, 1]$ interval:

$$C(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X' : a' \preceq a''\}|}{|X''|} \tag{3}$$

$$D(X', X'') = |\{a' \in X'; a'' \in X'' : a'' \npreceq a' \wedge a' \neq a''\}| \tag{4}$$

The value $C(X', X'') = 1$ in the former definitions means that all solutions in X'' are equal to or dominated by the solutions in X'. Its opposite value, $C(X', X'') = 0$, represents the situation where no solutions in X'' are covered by any solutions in X'. Both $C(X', X'')$ and $C(X'', X')$ must be considered since $C(X', X'')$ it is not necessary equal to $1 - C(X'', X')$. Function $D(X', X'')$ counts the number of individuals in X' that do not dominate X'' and are not found in X''.

We show in Table 3 the average results obtained for the comparissons among the MOEA algorithms. The first Table measures the $C(X', X'')$,and the other measures the $D(X', X'')$. This numbers were obtained by executing the algorithms 20 times with different seeds and calculating the average value for both functions and sequences.

$C(X', X'')$	MOSS	SPEA	$\mu + \lambda$
MOSS	-	0.538	0.360
SPEA	0.013	-	0.054
$\mu + \lambda$	0.029	0.349	-

$D(X', X'')$	MOSS	SPEA	$\mu + \lambda$
MOSS	-	14.204	12.977
SPEA	0.170	-	0.876
$\mu + \lambda$	1.066	2.284	-

Table 3. Sequence results

As we previously suggested, function D counts the number of nondominated individuals of an algorithm that were not found in the other two

MOEAs. The MOSS algorithm achieves the best value of D in all experiments, while SPEA and MuLambda present lower values. Moreover those results obtained by MOSS do not present much fluctuation between different sequences. MOSS leads the rankings followed by MuLambda and SPEA in the last position of the table. In addition, the diversity of solutions found by MOSS is considerably better than the other two algorithms (aproximately seven times better according to the D value). Finally, MOSS becomes the most robust algorithm by finding, in average, a specific promoter 16.81 times of the 20 runs. In contrast, SPEA obtains a promoter 6.48 times of the total 20 runs and and MuLambda 9.33 of the times.

6. Concluding Remarks

Generalized-clustering algorithms—solving multivariable, multiobjective, optimization problems—provide effective tools to identify interesting features that help to understand complex objects such as DNA sequences. We have proposed GAP, a promoter recognition method that was tested by predicting *E.coli* promoters. This method combines the advantages of feature representation based on fuzzy sets and the searching abilities of multiobjective genetic algorithms to obtain accurate as well as interpretable solutions. Particularly, these kinds of solutions are the most useful ones for the end users. That is, allows to detect multiple occurrences of promoters, sheding light on different putative transcription start sites. The ability of finding multiple promoters becomes more useful when the whole intergenic regions are considered, allowing to predict distinct regulatory activities, harboring activation or repression. The present approach can be extended to identify other DNA motifs, which are also conected by variable distances, such as binding sites of transcriptional regulators (e.g., direct or inverted repeats). Therefore, by combining multiple and heterogeneous DNA motifs (e.g., promoters, binding sites, etc.), we can obtain different descriptions of the cis-acting regions and, thus, different regulatory environments. The present implementation of GAP is available for academic use in the SOAR-TOOLS web site (http://soar-tools.wustl.edu) and will be updated soon with a new dataset from RegulonDB database[31] (in process).

Appendix

Tables 4 through 7 illustrate the set of solutions found by GAP by considering the set of promoter examples published in [5]. The last column of the tables indicates whether the GAP recognized the promoter or not by the simbols ✓ and □, respectively. The first column corresponds to the name of the sequence, the second column shows the beginning character position of the TTGACA-box, and the third column shows the character position where the TATAAT-box begins. These positions are those ones recognized by GAP. Only one result for each sequence is shown due to space limitations. The fourth column corresponds to the sequence itself with each of the boxes clearly depicted.

References

1. A. M. Huerta and J. Collado-Vides. Sigma70 promoters in escherichia coli: specific transcription in dense regions of overlapping promoter-like signals. *J Mol Biol.*, 333(2):261–278, 2003.
2. T. Bäck, D. Fogel, and Z. Michalewicz, Eds. 1997. *Handbook of Evolutionary Computation.* Institute of Physics Publishing and Oxford University Press.
3. J. C. Bezdek. 1998. Fuzzy clustering. In *Handbook of Fuzzy Computation.* E. H. Ruspini, P. P. Bonissone, & W. Pedrycz, Eds.: F6.2. Institute of Physics Press.
4. S. Brenner. Genomics. the end of the beginning. *Science*, 287(5461):2173–2179, 2000.
5. C. B. Harley and R. P. Reynolds. Analysis of e.coli promoter sequences. *Nucleic Acids Research*, 15(5):2343–2361, 1987.
6. R. Krishnapuram and J. Keller. 1993. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems,* 98–110.
7. B. Magasanik J. Collado-Vides, and J. D. Gralla. Control site location and transcriptional regulation in escherichia coli. *Microbiol Rev*, 55(3):371–394, 1991.
8. D. K. Hawley and W. R. McClure. Compilation and analysis of escherichia coli promoter dna sequences. *Nucleic Acids Research*, 11(8):2237–2255, 1983.
9. A. Ulyanov and G. Stormo. Multi-alphabet consensus algorithm for identification of low specificity protein-dna interactions. *Nucleic Acids Research*, 23(8):1434–1440, 1995.
10. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons, 2001.
11. M. Gibson and E. Mjolsness. *Computational Modeling of Genetic and Biochemical Networks*, chapter Modeling the Activity of Single Genes. The MIT Press, 2001.
12. D. E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, 1989.
13. L. Bailey and C. Elkan T. The value of prior knowledge in discovering motifs with meme. In *Proc Int Conf Intell Syst Mol Biol*, volume 3, pages 21–29, 1995.
14. G. Z. Hertz and G. D. Stormo. Identifiying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7/8):563–577, 1999.

15. G. J. Klir and T. A. Folger. *Fuzzy sets, uncertainty, and information.* Prentice Hall International, 1988.
16. Lawrence CE et. al. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.
17. S. Lisser and H. Margalit. Compilation of e.coli mrna promoter sequences. *Nucleic Acids Research*, 21(7):1507–1516, 1993.
18. R. Marti M. Laguna. *Scatter Search: Methodology and Implementations in C.* Kluwer Academic Publishers, Boston, 2003.
19. C. Mouslim, T. Latifi and 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):50588–95, 2003.
20. C. Newton R. Sarker, K. Liang. A new multiobjective evolutionary algorithm. *European Journal of Operational Research*, 140:12–23, 2002.
21. W. Pedrycz, P. P. Bonissone and E. H. Ruspini. *Handbook of fuzzy computation.* Institute of Physics, 1998.
22. M. Ptashne and A. Gann. *Genes and signals.* Cold Spring Harbor Laboratory Press, 2002.
23. M. G. Reese. Application of a time-delay neural network to promoter annotation in the drosophila melanogaster genome. *Computers & Chemistry*, 26(1):51–56, 2002.
24. J. van Helden, B. André and J. Collado-Vides. A web site for the computational analysis of yeast regulatory sequences. *Yeast*, 16(2):177–187, 2000.
25. D. Van Veldhuizen, C. Coello Coello, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, New York, May 2002.
26. J. Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry.* World Scientific.
27. E. H. Ruspini. 1969. A new approach to clustering. *Information and Control* 15:1, 22–32.
28. E. H. Ruspini and I. Zwir. 1999. Automated qualitative description of measurements. In *Proc. 16th IEEE Instrumentation and Measurement Technology Conf.*
29. E. H. Ruspini and I. Zwir. 2001. Automated Generation of Qualitative Representations of Complex Object by Hybrid Soft-computing Methods. In *Pattern Recognition: From Classical to Modern Approaches.* S. K. Pal & A. Pal, Eds. World Scientific Company, Singapore.
30. E. H. Ruspini and I. Zwir. Automated generation of qualitative representations of complex object by hybrid soft-computing methods. In S. K. Pal and A. Pal, editors, *Lecture Notes in Pattern Recognition.* World Scientific Company, 2001.
31. H. Salgado et al. Regulondb (version 3.2): transcriptional regulation and operon organisation in escherichia coli k-12. *Nucleic Acids Research*, 29:72–74, 2001.
32. L. A. Zadeh. 2000. Outline of a Computational Theory of Perceptions Based on Computing with Words. In *Soft Computing and Intelligent Systems: Theory and Applications.* N .K. Sinha, M. M. Gupta & L. A. Zadeh, Eds.: 3–22.

Academic Press, San Diego.

33. E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998.
34. E. Zitzler, L. Thiele, and K. Deb. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8:2, 173-195.
35. I.Zwir and E.H.Ruspini. 1999. Qualitative Object Description: Initial Reports of the Exploration of the Frontier. In *Proc. of the EUROFUSE-SIC'99*. Budapest, Hungary, 485-490.
36. I. Zwir, R. Romero Zaliz and E. H. Ruspini. Automated biological sequence description by genetic multiobjective generalized clustering. *Ann N Y Acad Sci*, 980:65–82, 2002.
37. I. Zwir, P. Traverso, and E. A. Groisman. Semantic-oriented analysis of regulation: the phop regulon as a model network. In *Proceedings of the 3rd International Conference on Systems Biology (ICSB)*, 2003.

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	✓
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	✓

Table 4. Results for the training sequences

sequence	ttgaca	tataat	promoter	found
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 GATGTTTACCGTGGAAAAGGG 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	✓
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	✓

Table 5. Results for the training sequences

sequence	ttgaca	tataat	promoter	found
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	✓
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	✓

Table 6. Results for the training sequences

sequence	ttgaca	tataat	promoter	found
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	✓
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	ACGAAAAACAGGTA 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	✓

Table 7. Results for the training sequences