ISSN 0026-8933, Molecular Biology, 2009, Vol. 43, No. 3, pp. 485–499. © Pleiades Publishing, Inc., 2009.
Original Russian Text © K.Yu. Gorbunov, E.V. Lyubetskaya, E.A. Asarin, V.A. Lyubetsky, 2009, published in Molekulyarnaya Biologiya, 2009, Vol. 43, No. 3, pp. 527–541.

UDC 575.852

Modeling Evolution of the Bacterial Regulatory Signals Involving Secondary Structure

K. Yu. Gorbunov, E. V. Lyubetskaya,
E. A. Asarin, and V. A. Lyubetsky

Kharkevich Institute for Information Transmission Problems,
Russian Academy of Sciences,
Moscow, 127994 Russia;
e-mail: gorbunov@iitp.ru

Received july 8, 2008
accepted for publication October 21, 2008

Abstract—An algorithm for modeling the evolution of the regulatory signals involving the interaction with RNA secondary structure is proposed. The algorithm implies that the species phylogenetic tree is known and is based on the assumption that the considered signals have a conserved secondary structure. The input data are the extant primary structure of a signal for all leaves of the phylogenetic tree; the algorithm computes the signal primary and secondary structures at all the nodes. Concurrently, the algorithm constructs a multiple alignment of the extant (in leaves) sites of a regulatory signal taking into account its secondary structure. The results of successful testing of the algorithm for three main types of attenuation regulation in bacteria—classic attenuation (threonine and leucine biosyntheses in Gammaproteobacteria), T-box (in Actinobacteria), and RFN-mediated (in Eubacteria) regulations—are described.

DOI: 10.1134/S0026893309030170

Key words: evolutionary scenario, regulatory signal, evolution along a tree, species tree, secondary structure

PROBLEM STATEMENT

Two problems. The problem to be considered (hereinafter, problem 1) is to model the ancestral states of noncoding genomic regions, namely, the regions (sites) of RNA-mediated regulation, in an analogous way as this problem is stated for the modeling of coding genomic regions (structural genes).

This problem is connected, yet does not coincide, with the known problem (hereinafter, problem 2) of constructing a multiple alignment of a specified set of nucleotide sequences taking into account an assumed (but unknown beforehand) common secondary structure in these sequences. There are algorithms for constructing multiple alignments that do not take into account the common secondary structure and, if the list of conserved helices constituting this presumable common secondary structure is known, then these algorithms or their improved versions can be used for solving problem 2. However, such set of conserved helices is usually unknown beforehand, and it is natural to connect the determination of this set (at least, this is how we do) with solving problem 1: the conserved helices searched for are the helices that are stable during the evolution of a primary structure. In this sense, the solution of problem 1 gives the desired list of conserved helices of the secondary structures in a specified set of extant RNA-mediated regulation sites, thereby leading to the solution of problem 2. This paper deals with problem 1.

A review of the modern state of the problem. The problem of constructing a phylogenetic tree for a protein family is well known and has been actively studied [1–5]. The associated issues belong to the even wider field of the modeling the evolution of species, described in a large body of literature [6–8]. The evolutionary models comprise modeling of molecular level events, such as gene duplications, originations, horizontal transfer, and losses, as well as constructing and estimating various evolutionary scenarios (for example, see [9–14]). In particular, such modeling can be performed concurrently with determining the lengths of tree branches by maximum likelihood estimation [15] or concurrently with constructing ancestral sequences provided that the initial data have been previously multiply aligned (see also http://evolution.genetics.washington.edu/phylip/software.

serv.html, where the available software for constructing phylogenetic trees and the reconstruction of ancestral sequences are listed). The corresponding studies are connected with modeling the evolutionary events themselves (for example, see [17–21]).

On the other hand, a considerable part of the bioinformatics research during the last two decades has been connected with the mechanisms involved in the regulation of gene expression, especially, in bacteria. The most important among these mechanisms are the regulation via protein–DNA interactions; classical attenuation-based regulation; T-box regulation; and the regulation involving RNA switches (riboswitches), certain special proteins, such as TRAP, or LEU elements. The main problems here are the search for regulatory elements (regulatory sites), functional and evolutionary analyses of the detected signals, and the modeling of regulatory processes (for example, see [22–27]). This problem is elucidated in hundreds of papers. The biological aspects of the evolution of regulatory structures are described by McAdams et al. [28]. As for the algorithmic aspects, ancestral sequences have been modeled until recently without taking into account the secondary structures or only partially taking them into account. Savill et al. [29] took into account the secondary structures via the modification of the standard model of nucleotide substitutions, namely, by adding the event of simultaneous substitution of two complementary nucleotides in connected positions. Note that genetic algorithms were applied to the simulation of helix evolution for assessing the rate of nucleotide substitutions and lengths of tree branches [30] as well as that the evolution of RNA secondary structures was taken into account to construct a more probable phylogenetic tree [31].

Our approach. We assume that a natural continuation of these two directions is modeling of the evolution of a regulatory signal along the phylogenetic tree of species or proteins (for example, a transcription factor) or along another tree corresponding to the evolution of a particular regulation. We have studied such evolutionary processes for the regulatory signals of amino acid metabolism when the mechanism of these signals involves alternative RNA secondary structures. We are developing two approaches to modeling such evolution. In both approaches, the phylogenetic tree and regulatory sites in its leaves (sites of one type, for example, sites of classical attenuation-based regulation) are given and the ancestral states of these sites are searched for in all the tree nodes, including the secondary structures in ancestral and extant sequences.

Our first approach is detailed in [32] and was presented at several conferences [33, 34; (see also http://lab6.iitp.ru, item 4)]. We will just relate briefly this approach for the sake of completeness. A Gibbs functional, $H(\sigma)$, is put down in an explicit form, its argument σ is a configuration, i.e., a function that ascribes a nucleotide sequence (assumed ancestral state of a regulatory signal in this node) to each inner tree node. Thus, the configuration is a set of ancestral states of the regulatory signal that is specified in the leaves. This functional $H(\sigma) = H_1(\sigma) + H_2(\sigma)$ comprises two conditions (constraints) for the desired configuration σ: (1) for each sequence $\sigma(k)$ (i.e., for the value of σ in the kth tree node) and along each edge leaving this node at each position $i = 1, \ldots, n$ of this sequence $\sigma(k)$, an independent substitution of letters occurs according to the substitution matrix R as well as insertions and deletions and (2) the sequences $\sigma(k)$ from configuration σ, when possible, preserve the secondary structure from the edge beginning to its end and along the pathway in the tree (for many generations); in this process, the longer and more numerous the pathways, the smaller is the functional. The first condition is represented in the term $H_1(\sigma)$, which describes a standard dynamics of regulatory site primary structure, and the second, in the term $H_2(\sigma)$, which describes the dynamics of site primary structure that provides for maintaining a high degree of the secondary structure conservation. The global minimum of the $H(\sigma)$ functional is searched for; according to the first approach, it corresponds to the biological evolution of the regulatory signal specified in tree leaves by only primary structure. Thus, $H(\sigma)$ represents the biologically motivated principles of a common evolution (dynamics) of primary structure and, concurrently, secondary structure conservation. This approach has been efficiently realized as a software and tested using the same examples as described below [32].

In this paper, we describe our second approach. It is based on two other biologically motivated requirements: (1) the secondary structure of a site must be as much conserved as possible along the pathways in the tree (from leaves to the root, as in the first approach; see below step 1 of the algorithm) and (2) the primary structure of a regulatory site must permit the smallest number of evolutionary events as possible (parsimony principle; see below step 2 of the algorithm).

Thus, the first approach is based on a conventional model of nucleotide substitutions and the requirement of conservation, while the second approach, on the parsimony principle and the same requirement of conservation. These requirements reflect the generally accepted concept of evolution. Correspondingly, these two approaches conceptually differ in their first requirements. However, they fundamentally differ in the implementation of these requirements—in the first approach, this is a minimization by annealing of the functional (exponent of the substitution matrix, indels, and some complex mathematical expressions describing the degree of conservation) and in the second, it is alternate alignments and minimizations of quite a different functional.

These two requirements of the second approach can be implemented in different ways, for example, it is possible to determine an integral (for the overall algorithm) functional with its minimum corresponding to the final configuration of ancestral sites or to reach this final configuration by iterations, as described in the algorithm below.

The purpose of several approaches to problem 1 is in comparing the results they give. This is a method for testing their adequacy, because experimental data on the ancestral states of bacterial regulatory signals are even more difficult to obtain than the data on ancestral states of genes. Note that these two approaches gave amazingly similar computational results for biological and artificial data.

We have implemented the model and the algorithm as a software and tested the algorithm using many examples of classical attenuation-mediated regulation, T-box regulation, riboswitches, LEU regulation, and so on. Note that our model is similarly applicable to the evolutionary analysis of these quite different regulations.

Multiple alignment of the initial sequences in leaves is, in general, unknown and in no way assumed given in our model. When testing our algorithm, we noticed that the output alignments and secondary structure were close to the known or postulated (according to independent studies) alignments for the sequences in leaves input to the algorithm. In addition, the algorithm outputs the multiple alignment of all the found sites at all the tree nodes taking into account a coevolution of their secondary structures. Actually, the algorithm also outputs as a side product the set of evolutionarily conserved helices, which is a fundamental step in solving problem 2.

As mentioned above, we detail here our second approach to solving problem 1; i.e., constructing the evolution of regulatory signals (with their secondary structure) in bacteria using an iteration procedure with configurations of the next new type. Such configuration ascribes a sequence of distributions rather than a nucleotide sequence (as in our first approach) to each tree node; the distribution specifies the frequencies of all nucleotides and, possibly, the symbol of a gap.

The sections Model Description, Algorithm, and Examples of Applying the Algorithm to Biological Data Analysis detail our model of evolution of the regulatory signal with secondary structure, describe the algorithm used for constructing the ancestral states of the signal specified in the leaves without specifying the secondary structure, and demonstrate the application of this algorithm to biological examples of various regulation types.

MODEL DESCRIPTION

As mentioned above, the model is based on two natural requirements: (1) the secondary structure of a site must be as much conserved as possible along the pathways within the tree (from leaves to root); see below step 1; and (2) the primary structure of a regulatory site must contain as low number of evolutionary events as possible; see below step 2.

A specific feature of the proposed model is in the following. A sequence is ascribed to each tree node; each position of this sequence is the frequencies of five symbols—four letters (A, C, T, and G) and a gap symbol (*d*). Thus, ascribed to the nodes are the sequences of various lengths composed of the vectors (distributions) with a length of 5; each vector consists of the numbers from zero to one totally giving unity; and these numbers correspond to the probabilities of A, C, T, G, and *d* (hereinafter, the numbers correspond to these letters in the indicated order). Let this sequence of vectors be named the sequence of distributions. Then the sequence ascribed to each node of the same tree is naturally transformed into a common nucleotide sequence with gaps (step 3, see below). Each sequence characterizes or directly represents the assumed site at a given tree node for one fixed regulation, the mechanism of which requires that RNA secondary regulatory structure is maintained during the evolution. In addition to the sequences of distributions, nucleotide sequences with gaps are ascribed to the tree leaves. The algorithm operation commences with the initial regulatory sites, which in the further process will be filled with gaps.

Designate as σ a sequence of distributions with a varying length n, where σ_i is its ith term ($1 \le i \le n$) and name i the ith position in σ. Designate $X(i, \sigma)$ the fraction of letter X at the ith position in sequence σ. Hereinafter, we do not distinguish between the sequence of distributions that has a distribution with one unity and the remaining zeros at each position and the corresponding nucleotide sequence with gaps. Note that for each leaf, the algorithm gives two variants of nucleotide sequence: one of them is designated with leaf number in the figures with even numbers and the other is designated with species name. Recollect that the initial sequence without gaps is given for each leaf at the beginning of algorithm operation.

In this model, we consider alignments of the primary structures of the sequences of distributions, i.e., the sequences themselves as words in an infinite alphabet, and their secondary structures as the words composed of "helices" in the sequences of distributions. The alignment is regarded as an insert of gaps into two words in a certain alphabet. For this purpose, one of the standard alignment algorithms utilizing the determined weights is applied to the words. The new feature is that the secondary structures in the sequences of distributions are taken into account. It is necessary to define the term helix, given above in quotation marks.

Let two positions, i and j, in the sequence of distributions σ be named complementary, if the following sum

$$[\min(\mathrm{A}(i,\sigma),\mathrm{T}(j,\sigma)) + \min(\mathrm{T}(i,\sigma),\mathrm{A}(j,\sigma))]$$

$$+ [\min(\mathrm{C}(i,\sigma),\mathrm{G}(j,\sigma)) + \min(\mathrm{G}(i,\sigma),\mathrm{C}(j,\sigma))]$$

$$+ 0.5\ [\min(\mathrm{G}(i,\sigma)\text{-}\min(\mathrm{G}(i,\sigma),\mathrm{C}(j,\sigma)),\ \min(\mathrm{T}(j,\sigma)\text{-}$$

$$\min(\mathrm{T}(j,\sigma),\mathrm{A}(i,\sigma))) + \min(\mathrm{T}(i,\sigma)\text{-}\min(\mathrm{T}(i,\sigma),\mathrm{A}(j,\sigma)),$$

$$\min(\mathrm{G}(j,\sigma)\text{-}\min(\mathrm{G}(j,\sigma),\mathrm{C}(i,\sigma)))]$$

$$+ 0.25\ [\min(d(i,\sigma),d(j,\sigma)]$$

is larger than a certain threshold, termed the complementarity threshold. It amounts to 0.5 in examples 1–4. The definition of complementarity reflects the interconnection of two positions where the corresponding distributions are located. The definitions of helix and loop in the sequence of distributions are naturally derived from the definition of complementarity (for example, see [26, 35]). For example, a helix is formed by the pairing of the maximally long segments of complementary positions in the sequence of distributions. As usual, we name these segments arms. Thus, the definitions of complementarity and helix for the sequence of distributions is a natural generalization of the common definitions for a nucleotide sequence.

To define the energy of helix, we generalize a standard definition (for example, see [26]), where the energy is summed for quadruplets of nucleotides (pairs of neighboring paired nucleotides). In our case, the summing is performed for analogous quadruplets of distributions. In the case of such a fixed quadruplet F of distributions, we designate as $X(i)$ the frequency of nucleotide X in the ith component of the quadruplet. Then the energy components corresponding to F is $\sum_{f = \langle X, Y, Z, V\rangle} E_f \frac{X(1)Y(2)Z(3)V(4)}{S}$, where the summing is performed over all the possible quadruplets f of nucleotides, E_f is the contribution of quadruplet f to energy calculated in a standard manner, and normalizing divisor $S = \sum_{f = \langle X, Y, Z, V\rangle} X(1)Y(2)Z(3)V(4)$. As usual, the energy thus calculated is supplemented with the terms that take into account the inner and outer loops. The loop length is defined as the sum of all nucleotide frequencies in it. The additional terms (entropy) are calculated for all rational arguments by a linear interpolation of the known values from [26] for the integer arguments. For example, if $f(x, y)$ is a standard function determining the entropy from an inner loop with the side lengths of $x = 2.75$ and $y = 3.25$, then

$$f(2.75, 3.25) = \frac{1}{4}f(2, 3.25) + \frac{3}{4}f(3, 3.25)$$

$$= \frac{1}{4}\left(\frac{3}{4}f(2, 3) + \frac{1}{4}(2, 4)\right) + \frac{3}{4}\biggl(\frac{3}{4}f(3, 3)$$

$$+ \frac{1}{4}f(3, 4)) = \frac{3}{16}f(2, 3) + \frac{1}{16}f(2, 4)$$

$$+ \frac{9}{16}f(3, 3) + \frac{3}{16}f(3, 4).$$

The cost of matching of two distributions or a distribution and a gap in a pairwise alignment of two sequences of distributions is determined as follows. Let $X(1)$ and $X(2)$ be the frequencies of symbol X in the first and second distributions, respectively; gap is conventionally designated d. Then this cost is

$$RC - P_1|d(1) - d(2)| - P_2(1 - C - \max(d(1), d(2))),$$

where $C = \sum_{X = \{A, C, T, G\}} \min(X(1), X(2))$, R is a standard bonus for a match of two nucleotides, P_1 is a standard penalty for matching a nucleotide and a gap, and P_2 is a standard penalty for the mismatch of two nucleotides. In certain more complex situations, we took smaller penalties for matching the nucleotides A and G or C and T, respectively, than for matching another pair of nucleotides.

ALGORITHM

Let one initial nucleotide sequence be ascribed to each tree leaf. The iterations consist in a successive alternation of steps 1 and 2. The proposed algorithm ascribes the sequences of distributions to all the tree nodes from leaves to root by the methods described below and concurrently performs the multiple alignment of all these sequences. In this process, it is actually necessary to align only two words, which is a simple task.

Step 1. One sequence of distributions produced in the course of computations in the previous iteration, step 2, and named end sequence of distributions is ascribed to each tree leaf. At the beginning of algorithm operation, the end sequence of distributions in a leaf is specified as a mere copy of the initial nucleotide sequence in this leaf, where each distribution is composed of unity and zeros. We also need the concept of altered nucleotide sequence; in general, this is the result of step 1 and, at the beginning, a copy of the initial nucleotide sequence. Then, we align these end and altered nucleotide sequences for each leaf without taking into account their secondary structures. In this process, the sequences can acquire a certain number of gaps. At the very beginning, these sequences in each leaf coincide and, correspondingly, their alignment is trivial. The result of step 1, which is conveyed to step 2, is an altered nucleotide sequence in each leaf. The result of step 2, which is conveyed to step 1, is the end sequence of distributions in each leaf. During the overall algorithm operation, the altered nucleotide sequence in a leaf differs from the initial one specified in this leaf by only the addition of gaps.

Let two sons ν_1 and ν_2 of the node ν to already ascribe sequences of distributions σ_1 and σ_2. The algorithm must determine the sequence of distributions σ in ν. For this purpose, it first aligns the secondary structures of these sequences in the following way. According to σ_1 and σ_2, we determine two sets Ω_1 and Ω_2, respectively, which consist of the helices (in the first and second sequences, respectively) with the energy exceeding a certain threshold, which in examples 1–4 is assumed as 10 kcal/mol. We name it the energy threshold.

From the sets Ω_1 and Ω_2, the algorithm proceeds to the linear orderings of arms of the helices belonging to these sets. More precisely, the algorithm linearly orders the arms: roughly speaking, it arranges the arms in the same order as they are located in the sequence, namely, at the middle of the arm and, if the middle points coincide, at the beginning of the arms. Each arm is included into this ordering as a letter from a certain new alphabet, which reflects the information about the beginning and end of the arm together with its neighborhood of a certain size, nucleotide composition of the arm and its neighborhood, the number of helix to which the arm belongs, and the information about whether it is right or left. We name these orderings words in the nodes ν_1 and ν_2, respectively, and designate also Ω_1 and Ω_2 (it is also possible to take trees as Ω_1 and Ω_2; the obtained result is close to the variant of words). These words are aligned. Then the sequences σ_1 and σ_2 themselves are aligned taking into account the obtained alignment of secondary structures in σ_1 and σ_2. This procedure is described in detail in [36]. We determine the weights for such alignments: for the matching of two distributions or two arms and for the matching of a distribution or an arm to a gap, which comply with a common model of nucleotide substitutions. Here we do not show the weights; we note only the case of two arms. In this situation, the weight is determined as follows: the quality obtained when aligning the arms with their neighborhoods as nucleotide sequences is summed to the bonus for each matching in alignment of the left and right arms of the same helix; here, only the matching of left to left or right to right arms are allowed.

Then for each position i, we take as the distribution σ_i the weighted mean of the distributions σ_{1i} and σ_{2i} with the weights that are determined by the ratio of lengths of two edges from ν to ν_1 and ν_2, i.e.,

$$\sigma_i = \frac{l(\nu, \nu_2)}{l(\nu, \nu_1) + l(\nu, \nu_2)}\sigma_{1i} + \frac{l(\nu, \nu_1)}{l(\nu, \nu_1) + l(\nu, \nu_2)}\sigma_{2i},$$

where $l(\nu,\nu_i)$ is the length of the edge coming from node ν to node ν_i.

This gives the sequence of distributions σ in the node ν; then the former σ_1 and σ_2 in ν_1 and ν_2 are replaced with the sequences obtained by alignment; the latter are also designated as σ_1 and σ_2. Evidently, the new σ_1 and σ_2 are obtained from the former sequences of distributions by a coordinated addition of a certain number of gaps into the former sequences. The addition of gaps into all the sequences of distributions ascribed below to σ, including the altered and initial nucleotide sequences ascribed to the tree leaves, is continued. Now all these descendants of the sequence σ have an equal length.

When step 1 reaches the root, the altered nucleotide sequences are formed in the leaves; these sequences are the only result of step 1, and it is conveyed to step 2. All the sequences (of distributions and nucleotides) ascribed to the nodes at the end of step 1 have the same lengths; we name it the length of the zone. Note that both the sequence of distributions and the altered nucleotide sequence with gaps are ascribed to each leaf. During the algorithm operation, the altered nucleotide sequence in any leaf differs from the initial input sequence in this leaf only by the added gaps. Now proceed to step 2.

Step 2. The following procedure is performed for each position of the altered nucleotide sequences generated during step 1. Each node ν of the evolutionary tree is ascribed with the set δ(ν) of the frequencies of five possible symbols at the considered position. At step 2, all these values are considered variables; and the below described functional *F* is minimized by these variables (search for the nearest local minimum).

Let ρ be a certain measure of closeness of vectors (in the simplest case, the sum of squares of the differences between components); for each leaf ν, let σ(ν) denote the constant vector where unity corresponds to the symbol of the altered nucleotide sequence ascribed to this leaf, let zero denote the remaining elements, and let e_b and e_e be the ends of edge *e*. The functional *F* is defined by the following equation (the first summing is performed over all the tree edges and the second, over all its leaves):

$$F = \sum_e \rho(\delta(e_n), \delta(e_\kappa)) \cdot w(e) + \sum_\nu \rho(\delta(\nu), \sigma(\nu)) \cdot w(\nu).$$

Here, *w*(*e*) and *w*(ν) are weight coefficients. The weight *w*(*e*) is the larger, the shorter is the edge *e*; the weight *w*(ν) is equal to the weight *w*(*e*) of the edge *e* coming from the leaf ν to its parent multiplied by a special parameter regulating the closeness of the distributions in leaves to the initial data relative to the conservation of distributions in the inner tree nodes. In examples 1–4, this parameter is 1.5.

Minimization (in turn for each position) is performed over all variables, i.e., over the fractions of the five symbols in all tree nodes. The natural constraints are imposed: all the variables are non-negative and the sum of the corresponding five variables at each node is unity. In the mentioned simplest case, the functional is quadratic with only one minimum; thus, this minimum point is easily found by quadratic programming. The end (in leaves) sequences of distributions obtained by this minimization is the only result of step 2, which is conveyed to step 1.

We considered a more complex variant of this algorithm with the additional step aimed to take into account the mutations of letters according to one of the substitution models as well as insertions and deletions in the primary structure. However, this does not lead to a considerable difference for the classic attenuation regulation. For other regulation types, the situation is more intricate, and is beyond our consideration here.

The algorithm terminates the alternation of steps 1 and 2 (at step 1), if the length of the zone stops growing. Such a choice of the criterion for terminating the alteration of steps 1 and 2 results from a heuristic observation that the algorithm cycled and the length of the zone stopped growing when a good multiple alignment of the nucleotide sequences in leaves, taking into account the secondary structure, was found. We do not state that this will be true for any input data; however, this situation was observed for the considered examples. Then the algorithm proceeds to step 3, where it uses only the final multiple alignment of the sequences of distributions in all the tree nodes obtained after steps 1 and 2.

Step 3. In the final multiple alignment, we proceed from the sequences of distributions to new nucleotide sequences with gaps, which are ascribed to all the tree nodes including leaves. All these new sequences have the same constant lengths as in the final multiple alignment of the sequences of distributions. These particular sequences are shown in the figures with even numbers, where they are denoted by the numbers of the corresponding tree nodes; the initial nucleotide sequences with the gaps added during the algorithm operation are designated with the name of the corresponding species. It was performed in the following way: if a position is not contained in a helix, we place there the symbol (nucleotide or gap) that has the highest frequency in the distribution corresponding to these position and tree node. If this position is contained in one of several helices, we place there the letter that paired with the highest energy within the maximal number of helices and yet retains a sufficient frequency. Then we examine which of the four letters displays the best value of the following characteristic—the sum of the energies of all helices containing this position, where each term is multiplied by the weight equal to the fraction of the frequency of this letter that is involved in this helix; if none of the four letters reaches a certain threshold value of this characteristic, then a gap is put down. The figures with even numbers show the final multiple alignment of new nucleotide sequences obtained after step 3 together with the altered nucleotide sequences with gaps generated at the last step 1 pro-

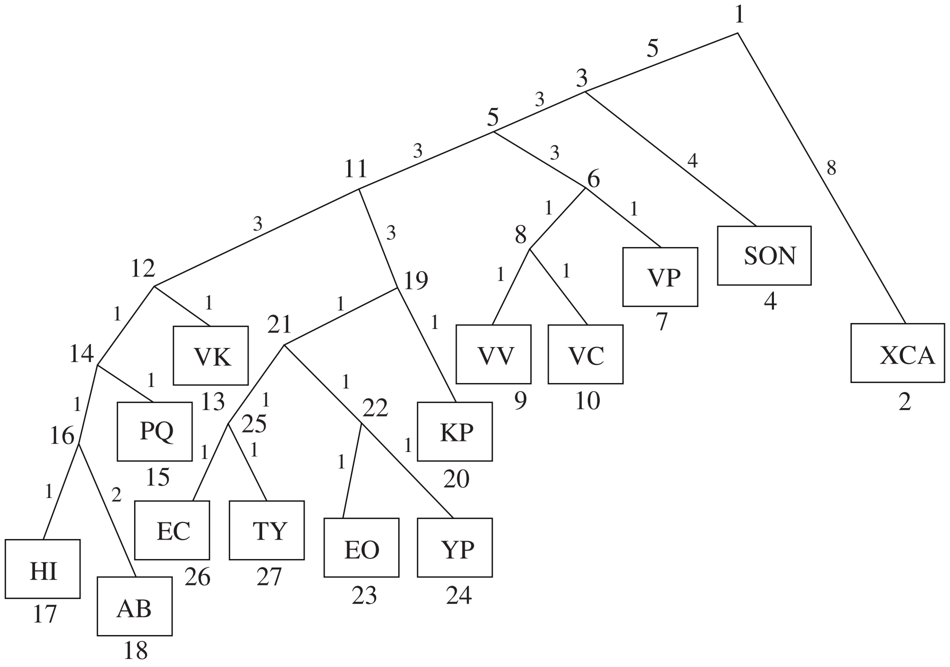


Fig. 1. Phylogenetic tree for the classic attenuation regulation of threonine biosynthesis in Gammaproteobacteria.

cedure, the same sequences as the initial ones with the gaps added during the algorithm operation. In figures, the former sequences are indicated with the numbers of nodes, and the latter are indicated with the name of species.

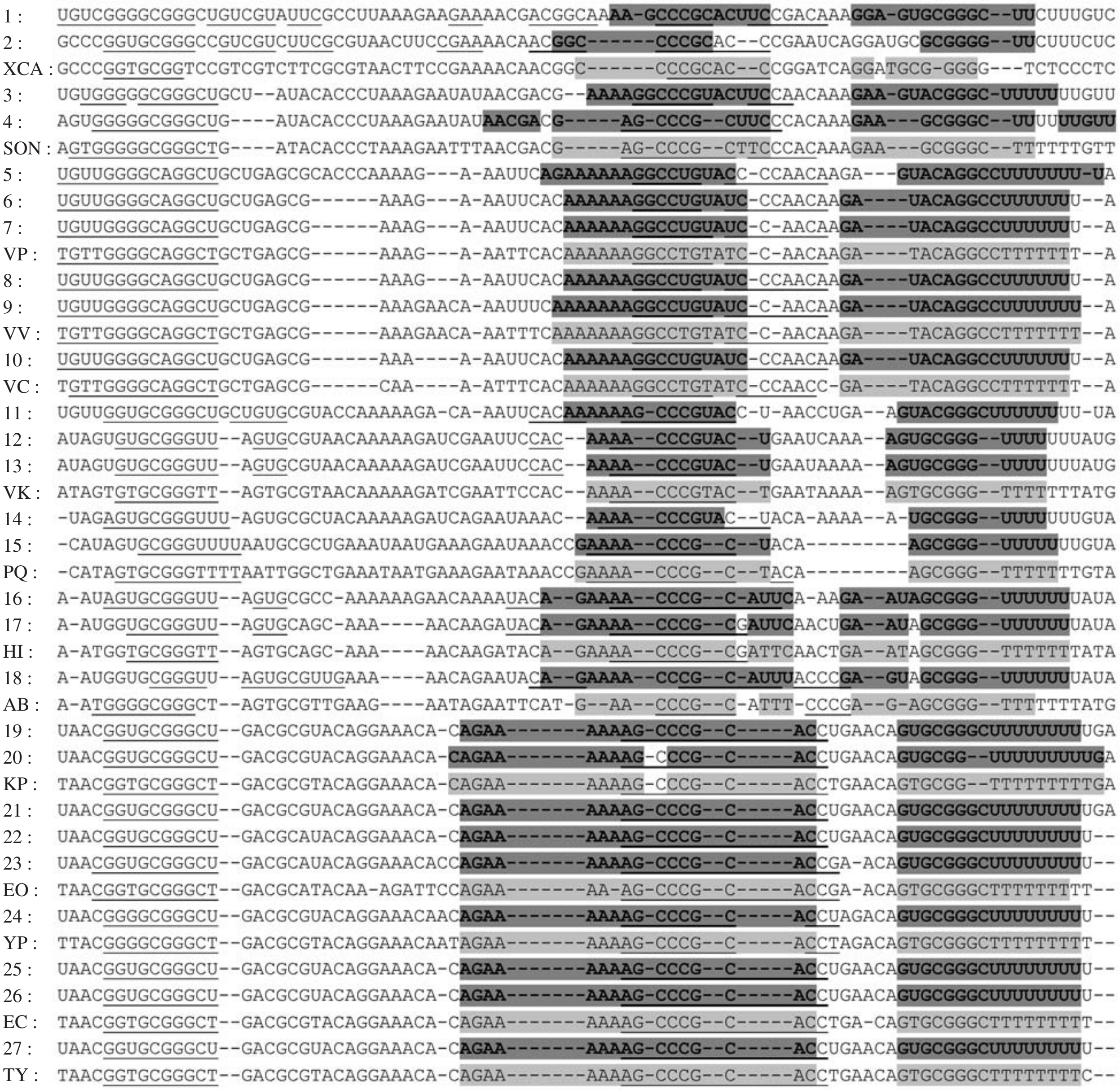
Step 4. According to a certain threshold, we select the most conserved helices from the final multiple alignment of the new nucleotide sequences in leaves; according to the alignment, we indicate the corresponding sequences in the altered nucleotide sequences (see the figures mentioned above); we output the helices induced by these in the initial nucleotide sequences. Thus, the evolutionarily induced secondary structure, which was beforehand unknown to the algorithm, is indicated in the initial data. This provides for an independent multiple alignment of the initial nucleotide sequences taking into account this secondary structure. The result is close to that after step 3.

EXAMPLES OF APPLYING THE ALGORITHM TO BIOLOGICAL DATA ANALYSIS

Due to the preliminary character of this paper and the insufficient volume, we confined ourselves to four examples (one–two for each main type of regulation based on mRNA secondary structure). For these examples, the known PAML and PAUP programs (see Discussion) gave considerably worse results as compared with our algorithm (data not shown). More comprehensive testing results are available on the site of our Laboratory (http://lab6.iitp.ru, item 9).

Example 1. Consider the classic attenuation regulation of threonine biosynthesis in Gammaproteobacteria. The initial sites in leaves are taken from [22] (the algorithm uses neither the information about secondary structure known beforehand nor multiple alignment). A standard species tree (Fig. 1) is taken; in this tree, the nodes are numbered from 1 to 27 (the numbers are under rectangles), and each edge is ascribed with its phylogenetic length in arbitrary units, a smaller number. The following abbreviations are used: EC for *Escherichia coli*, TY for *Salmonella typhi*, KP for *Klebsiella pneumoniae*, EO for *Erwinia carotovora*, YP for *Yersinia pestis*, HI for *Haemophilus influenzae*, VK for *Pasterella multocida*, AB for *Actinobacillus actinomycetemcomitans*, PQ for *Mannheimia haemolytica*, VC for *Vibrio cholerae*, VV for *Vibrio vulnificus*,, VP for *Vibrio parahaemolyticus*, SON for *Shewanella oneidensis*, and XCA for *Xanthomonas campestris.*

The algorithm outputs the final multiple alignment of nucleotide sequences—the assumed regulatory sites and altered nucleotide sequences (see Fig. 2). The terminator is shown dark gray, and the antitermi-



```
1 :   UGUCGGGGCGGCUGUCGUAUUCGCCUUAAAGAAGAAAACGACGGCAAAA-GCCCGCACUUCCGACAAAGGA-GUGCGGGC--UUCUUUGUC
2 :   GCCCGGUGCGGGCCGUCGUCUUCGCGUAACUUCCGAAAACAACGGC------CCCGCAC--CCGAAUCAGGAUGCGCGGGG--UUCUUUCUC
XCA : GCCCGGTGCGGTCCGTCGTCTTCGCGTAACTTCCGAAAACAACGGC------CCCGCAC--CCGGATCAGGATGCG-GGGG---TCTCCCTC
3 :   UGUGGGGGCGGGCUGCU--AUACACCCUAAAGAAUAUAACGACG--AAAAGGCCCGUACUUCCAACAAAGAA-GUACGGGC-UUUUUUUGUU
4 :   AGUGGGGGCGGGCUG----AUACACCCUAAAGAAUAUAACGACG-----AG-CCCG--CUUCCCACAAAGAA---GCGGGC--UUUUUUGUU
SON : AGTGGGGGCGGGCTG----ATACACCCTAAAGAATTTAACGACG-----AG-CCCG--CTTCCCACAAAGAA---GCGGGC--TTTTTTGTT
5 :   UGUUGGGGCAGGCUGCUGAGCGCACCCAAAAG---A-AAUUCAGAAAAAAGGCCUGUACC-CCAACAAGA---GUACAGGCCUUUUUUUU-UA
6 :   UGUUGGGGCAGGCUGCUGAGCG------AAAG---A-AAUUCACAAAAAAGGCCUGUAUC-CCAACAAGA----UACAGGCCUUUUUUUU--A
7 :   UGUUGGGGCAGGCUGCUGAGCG------AAAG---A-AAUUCACAAAAAAGGCCUGUAUC-C-AACAAGA----UACAGGCCUUUUUUUU--A
VP :  TGTTGGGGCAGGCTGCTGAGCG------AAAG---A-AATTCACAAAAAAGGCCTGTATC-C-AACAAGA----TACAGGCCTTTTTTTT--A
8 :   UGUUGGGGCAGGCUGCUGAGCG------AAAG---A-AAUUCACAAAAAAGGCCUGUAUC-CCAACAAGA----UACAGGCCUUUUUUUU--A
9 :   UGUUGGGGCAGGCUGCUGAGCG------AAAGAACA-AAUUUCAAAAAAAGGCCUGUAUC-C-AACAAGA----UACAGGCCUUUUUUUUU--A
VV :  TGTTGGGGCAGGCTGCTGAGCG------AAAGAACA-AATTTCAAAAAAAGGCCTGTATC-C-AACAAGA----TACAGGCCTTTTTTTTT--A
10 :  UGUUGGGGCAGGCUGCUGAGCG------AAA----A-AAUUCACAAAAAAGGCCUGUAUC-CCAACAAGA----UACAGGCCUUUUUUUU--A
VC :  TGTTGGGGCAGGCTGCTGAGCG------CAA----A-ATTTCACAAAAAAGGCCTGTATC-CCAACC-GA----TACAGGCCTTTTTTTT--A
11 :  UGUUGGUGCGGGCUGCUGUGCGUACCAAAAAGA-CA-AAUUCACAAAAAAG-CCCGUACC-U-AACCUGA--AGUACGGGCUUUUUUUU-UA
12 :  AUAGUGUGCGGGUU--AGUGCGUAACAAAAAGAUCGAAUUCCAC--AAAA--CCCGUAC--UGAAUCAAA--AGUGCGGG--UUUUUUUAUG
13 :  AUAGUGUGCGGGUU--AGUGCGUAACAAAAAGAUCGAAUUCCAC--AAAA--CCCGUAC--UGAAUAAAA--AGUGCGGG--UUUUUUUAUG
VK :  ATAGTGTGCGGGTT--AGTGCGTAACAAAAAGATCGAATTCCAC--AAAA--CCCGTAC--TGAATAAAA--AGTGCGGG--TTTTTTTATG
14 :  -UAGAGUGCGGGUUU-AGUGCGCUACAAAAAGAUCAGAAUAAAC--AAAA--CCCGUAC--UACA-AAAA--A-UGCGGG--UUUUUUUGUA
15 :  -CAUAGUGCGGGUUUUAAUGCGCUGAAAUAAUGAAAGAAUAAACCGAAAA--CCCG--C--UACA---------AGCGGG--UUUUUUUGUA
PQ :  -CATAGTGCGGGTTTTAATTGGCTGAAATAATGAAAGAATAAACCGAAAA--CCCG--C--TACA---------AGCGGG--TTTTTTTGTA
16 :  A-AUAGUGCGGGUU--AGUGCGCC-AAAAAGAACAAAAUACA--GAAAA--CCCG--C-AUUCA-AAGA--AUAGCGGG--UUUUUUUAUA
17 :  A-AUGGUGCGGGUU--AGUGCAGC-AAA----AACAAGAUACA--GAAAA--CCCG--CGAUUCAACUGA--AUAGCGGG--UUUUUUUAUA
HI :  A-ATGGTGCGGGTT--AGTGCAGC-AAA----AACAAGATACA--GAAAA--CCCG--CGATTCAACTGA--ATAGCGGG--TTTTTTTATA
18 :  A-AUGGUGCGGGUU--AGUGCGUUGAAA----AACAGAAUACA--GAAAA--CCCG--C-AUUUACCCGA--GUAGCGGG--UUUUUUUAUA
AB :  A-ATGGGGCGGGCT--AGTGCGTTGAAG----AATAGAATTCAT-G--AA--CCCG--C-ATTT-CCCGA--G-AGCGGG--TTTTTTTATG
19 :  UAACGGUGCGGGCU--GACGCGUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGGGCUUUUUUUUUGA
20 :  UAACGGUGCGGGCU--GACGCGUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGG--UUUUUUUUUGA
KP :  TAACGGTGCGGGCT--GACGCGTACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCTGAACAGTGCGG--TTTTTTTTTGA
21 :  UAACGGUGCGGGCU--GACGCGUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGGGCUUUUUUUUUGA
22 :  UAACGGUGCGGGCU--GACGCAUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGGGCUUUUUUUUU--
23 :  UAACGGUGCGGGCU--GACGCAUACAGGAAACACCAGAA-------AAAAG-CCCG--C-----ACCGA-ACAGUGCGGGCUUUUUUUUU--
EO :  TAACGGTGCGGGCT--GACGCATACAA-AGATTCCAGAA-------AA-AG-CCCG--C-----ACCGA-ACAGTGCGGGCTTTTTTTTT--
24 :  UAACGGGGCGGGCU--GACGCGUACAGGAAACAACAGAA-------AAAAG-CCCG--C-----ACCUAGACAGUGCGGGCUUUUUUUUU--
YP :  TTACGGGGCGGGCT--GACGCGTACAGGAAACAATAGAA-------AAAAG-CCCG--C-----ACCTAGACAGTGCGGGCTTTTTTTTT--
25 :  UAACGGUGCGGGCU--GACGCGUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGGGCUUUUUUUUU--
26 :  UAACGGUGCGGGCU--GACGCGUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGGGCUUUUUUUUU--
EC :  TAACGGTGCGGGCT--GACGCGTACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCTGA-CAGTGCGGGCTTTTTTTTT--
27 :  UAACGGUGCGGGCU--GACGCGUACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCUGAACAGUGCGGGCUUUUUUUUU--
TY :  TAACGGTGCGGGCT--GACGCGTACAGGAAACA-CAGAA-------AAAAG-CCCG--C-----ACCTGAACAGTGCGGGCTTTTTTTTC--
```

Fig. 2. The multiple alignment (with secondary structure) of assumed regulatory sites generated by the proposed algorithm for the classic attenuation regulation of threonine biosynthesis in Gammaproteobacteria.

nator is underlined. The terminator in altered nucleotide sequences is shown in light gray. The secondary structures obtained in leaves (Fig. 2) only slightly differ from the secondary structures either predicted by other bioinformatics methods or determined experimentally [26].

Example 2. Consider the classic attenuation regulation of leucine biosynthesis in Gammaproteobacteria. The initial sites in leaves are taken from [22]. A standard species tree (Fig. 3) is taken.

The algorithm outputs the final multiple alignment of nucleotide sequences—the assumed regulatory

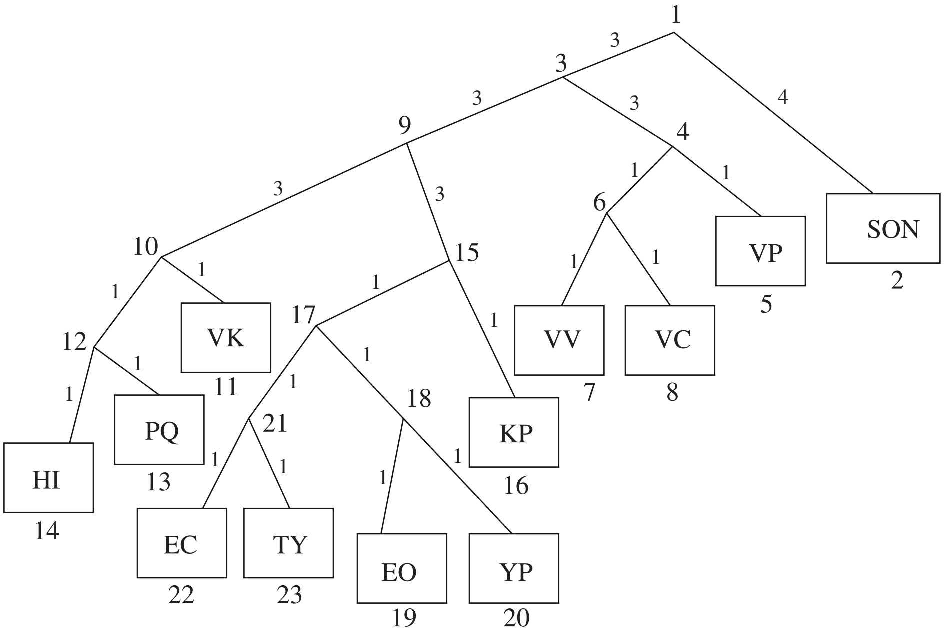
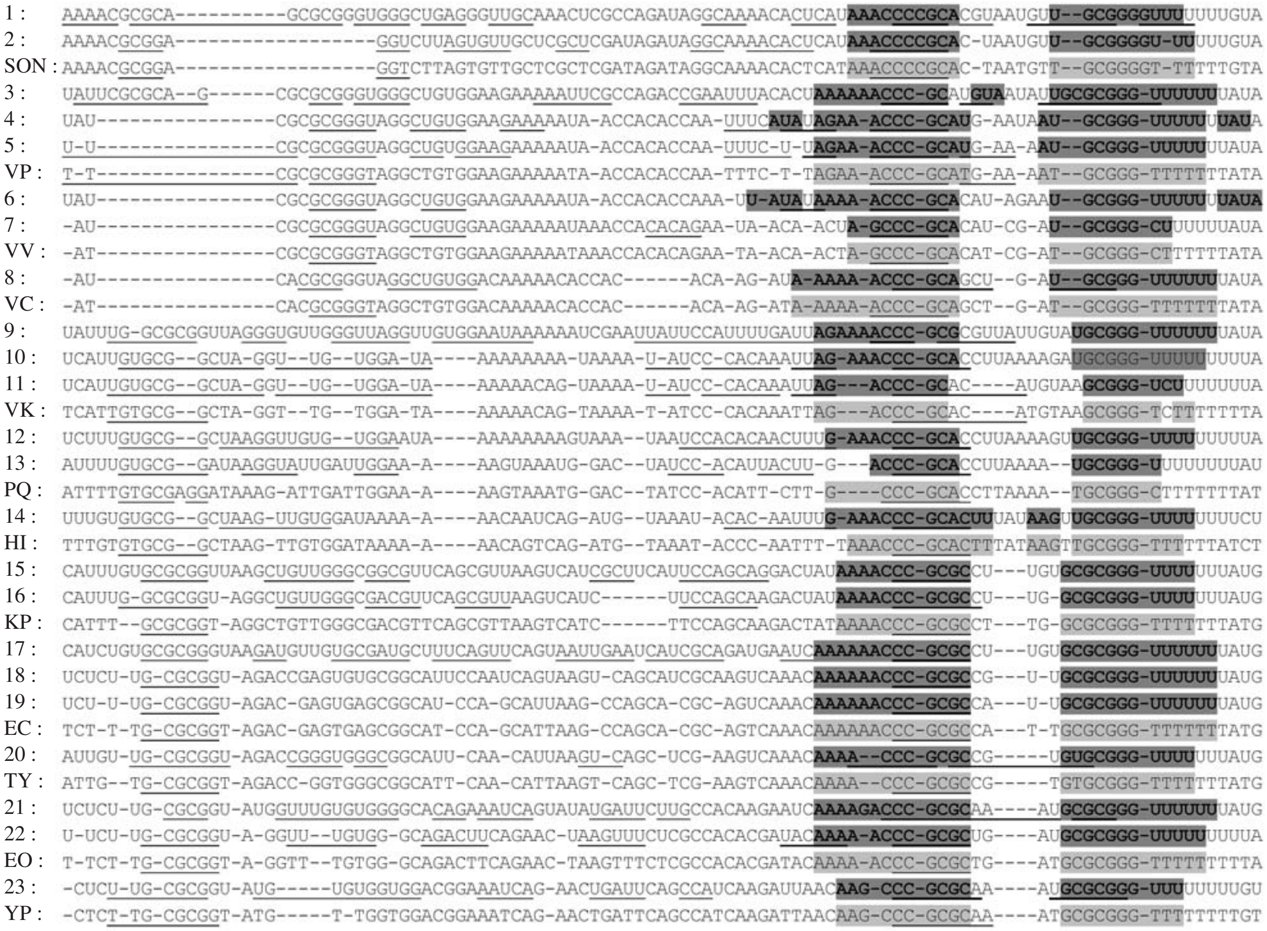


Fig. 3. Phylogenetic tree for the classic attenuation regulation of leucine biosynthesis in Gammaproteobacteria.



```
1 :   AAAACGCGCA----------GCGCGGGUGGGCUGAGGGUUGCAAACUCGCCAGAUAGGCAAAACACUCAUAAACCCCGCACGUAAUGUU--GCGGGGUUUUUUUGUA
2 :   AAAACGCGGA------------------GGUCUUAGUGUUGCUCGCUCGAUAGAUAGGCAAAACACUCAUAAACCCCGCAC-UAAUGUU--GCGGGGU-UUUUUGUA
SON : AAAACGCGGA------------------GGTCTTAGTGTTGCTCGCTCGATAGATAGGCAAAACACTCATAAACCCCGCAC-TAATGTT--GCGGGGT-TTTTTGTA
3 :   UAUUCGCGCA--G------CGCGCGGGUGGGCUGUGGAAGAAAAAUUCGCCAGACCGAAUUUACACUAAAAAACCC-GCAUGUAAUAUUGCGCGGG-UUUUUUUAUA
4 :   UAU----------------CGCGCGGGUAGGCUGUGGAAGAAAAAUA-ACCACACCAA-UUUCAUAUAGAA-ACCC-GCAUG-AAUAAU--GCGGG-UUUUUUUAUA
5 :   U-U----------------CGCGCGGGUAGGCUGUGGAAGAAAAAUA-ACCACACCAA-UUUC-U-UAGAA-ACCC-GCAUG-AA-AAU--GCGGG-UUUUUUUAUA
VP :  T-T----------------CGCGCGGGTAGGCTGTGGAAGAAAAATA-ACCACACCAA-TTTC-T-TAGAA-ACCC-GCATG-AA-AAT--GCGGG-TTTTTTTATA
6 :   UAU----------------CGCGCGGGUAGGCUGUGGAAGAAAAAUA-ACCACACCAAA-UU-AUAUAAAA-ACCC-GCACAU-AGAAU--GCGGG-UUUUUUUAUA
7 :   -AU----------------CGCGCGGGUAGGCUGUGGAAGAAAAAUAAACCACACAGAA-UA-ACA-ACUA-GCCC-GCACAU-CG-AU--GCGGG-CUUUUUUAUA
VV :  -AT----------------CGCGCGGGTAGGCTGTGGAAGAAAAATAAACCACACAGAA-TA-ACA-ACTA-GCCC-GCACAT-CG-AT--GCGGG-CTTTTTTATA
8 :   -AU----------------CACGCGGGUAGGCUGUGGACAAAAACACCAC------ACA-AG-AUA-AAAA-ACCC-GCAGCU--G-AU--GCGGG-UUUUUUUAUA
VC :  -AT----------------CACGCGGGTAGGCTGTGGACAAAAACACCAC------ACA-AG-ATA-AAAA-ACCC-GCAGCT--G-AT--GCGGG-TTTTTTTATA
9 :   UAUUUG-GCGCGGUUAGGGUGUUGGGUUAGGUUGUGGAAUAAAAAAUCGAAUUAUUCCAUUUUGAUUAGAAAACCC-GCGCGUUAUUGUAUGCGGG-UUUUUUUAUA
10 :  UCAUUGUGCG--GCUA-GGU--UG--UGGA-UA----AAAAAAAA-UAAAA-U-AUCC-CACAAAUUAG-AAACCC-GCACCUUAAAAGAUGCGGG-UUUUUUUUUA
11 :  UCAUUGUGCG--GCUA-GGU--UG--UGGA-UA----AAAAACAG-UAAAA-U-AUCC-CACAAAUUAG---ACCC-GCAC----AUGUAAGCGGG-UCUUUUUUUA
VK :  TCATTGTGCG--GCTA-GGT--TG--TGGA-TA----AAAAACAG-TAAAA-T-ATCC-CACAAATTAG---ACCC-GCAC----ATGTAAGCGGG-TCTTTTTTTA
12 :  UCUUUGUGCG--GCUAAGGUUGUG--UGGAAUA----AAAAAAAAGUAAA--UAAUCCACACAACUUUG-AAACCC-GCACCUUAAAAGUUGCGGG-UUUUUUUUUA
13 :  AUUUUGUGCG--GAUAAGGUAUUGAUUGGAA-A----AAGUAAAUG-GAC--UAUCC-ACAUUACUU-G---ACCC-GCACCUUAAAA--UGCGGG-UUUUUUUUAU
PQ :  ATTTTGTGCGAGGATAAAG-ATTGATTGGAA-A----AAGTAAATG-GAC--TATCC-ACATT-CTT-G----CCC-GCACCTTAAAA--TGCGGG-CTTTTTTTAT
14 :  UUUGUGUGCG--GCUAAG-UUGUGGAUAAAA-A----AACAAUCAG-AUG--UAAAU-ACAC-AAUUUG-AAACCC-GCACUUUAUAAGUUGCGGG-UUUUUUUUCU
HI :  TTTGTGTGCG--GCTAAG-TTGTGGATAAAA-A----AACAGTCAG-ATG--TAAAT-ACCC-AATTT-TAAACCC-GCACTTTATAAGTTGCGGG-TTTTTTATCT
15 :  CAUUUGUGCGCGGUUAAGCUGUUGGGCGGCGUUCAGCGUUAAGUCAUCGCUUCAUUCCAGCAGGACUAUAAAACCC-GCGCCU---UGUGCGCGGG-UUUUUUUAUG
16 :  CAUUUG-GCGCGGU-AGGCUGUUGGGCGACGUUCAGCGUUAAGUCAUC------UUCCAGCAAGACUAUAAAACCC-GCGCCU---UG-GCGCGGG-UUUUUUUAUG
KP :  CATTT--GCGCGGT-AGGCTGTTGGGCGACGTTCAGCGTTAAGTCATC------TTCCAGCAAGACTATAAAACCC-GCGCCT---TG-GCGCGGG-TTTTTTTATG
17 :  CAUCUGUGCGCGGGUAAGAUGUUGUGCGAUGCUUUCAGUUCAGUAAUUGAAUCAUCGCAGAUGAAUCAAAAAACCC-GCGCCU---UGUGCGCGGG-UUUUUUUAUG
18 :  UCUCU-UG-CGCGGU-AGACCGAGUGUGCGGCAUUCCAAUCAGUAAGU-CAGCAUCGCAAGUCAAACAAAAAACCC-GCGCCG---U-UGCGCGGG-UUUUUUUAUG
19 :  UCU-U-UG-CGCGGU-AGAC-GAGUGAGCGGCAU-CCA-GCAUUAAG-CCAGCA-CGC-AGUCAAACAAAAAACCC-GCGCCA---U-UGCGCGGG-UUUUUUUAUG
EC :  TCT-T-TG-CGCGGT-AGAC-GAGTGAGCGGCAT-CCA-GCATTAAG-CCAGCA-CGC-AGTCAAACAAAAAACCC-GCGCCA---T-TGCGCGGG-TTTTTTTATG
20 :  AUUGU-UG-CGCGGU-AGACCGGGUGGGCGGCAUU-CAA-CAUUAAGU-CAGC-UCG-AAGUCAAACAAAA--CCC-GCGCCG-----UGUGCGGG-UUUUUUUAUG
TY :  ATTG--TG-CGCGGT-AGACC-GGTGGGCGGCATT-CAA-CATTAAGT-CAGC-TCG-AAGTCAAACAAAA--CCC-GCGCCG-----TGTGCGGG-TTTTTTTATG
21 :  UCUCU-UG-CGCGGU-AUGGUUUGUGUGGGGCACAGAAAUCAGUAUAUGAUUCUUGCCACAAGAAUCAAAAGACCC-GCGCAA----AUGCGCGGG-UUUUUUUAUG
22 :  U-UCU-UG-CGCGGU-A-GGUU--UGUGG-GCAGACUUCAGAAC-UAAGUUUCUCGCCACACGAUACAAAA-ACCC-GCGCUG----AUGCGCGGG-UUUUUUUUUA
EO :  T-TCT-TG-CGCGGT-A-GGTT--TGTGG-GCAGACTTCAGAAC-TAAGTTTCTCGCCACACGATACAAAA-ACCC-GCGCTG----ATGCGCGGG-TTTTTTTTTA
23 :  -CUCU-UG-CGCGGU-AUG-----UGUGGUGGACGGAAAUCAG-AACUGAUUCAGCCAUCAAGAUUAACAAG-CCC-GCGCAA----AUGCGCGGG-UUUUUUUUGU
YP :  -CTCT-TG-CGCGGT-ATG-----T-TGGTGGACGGAAATCAG-AACTGATTCAGCCATCAAGATTAACAAG-CCC-GCGCAA----ATGCGCGGG-TTTTTTTTGT
```

Fig. 4. The multiple alignment (with secondary structure) of assumed regulatory sites generated by the proposed algorithm for the classic attenuation regulation of leucine biosynthesis in Gammaproteobacteria.

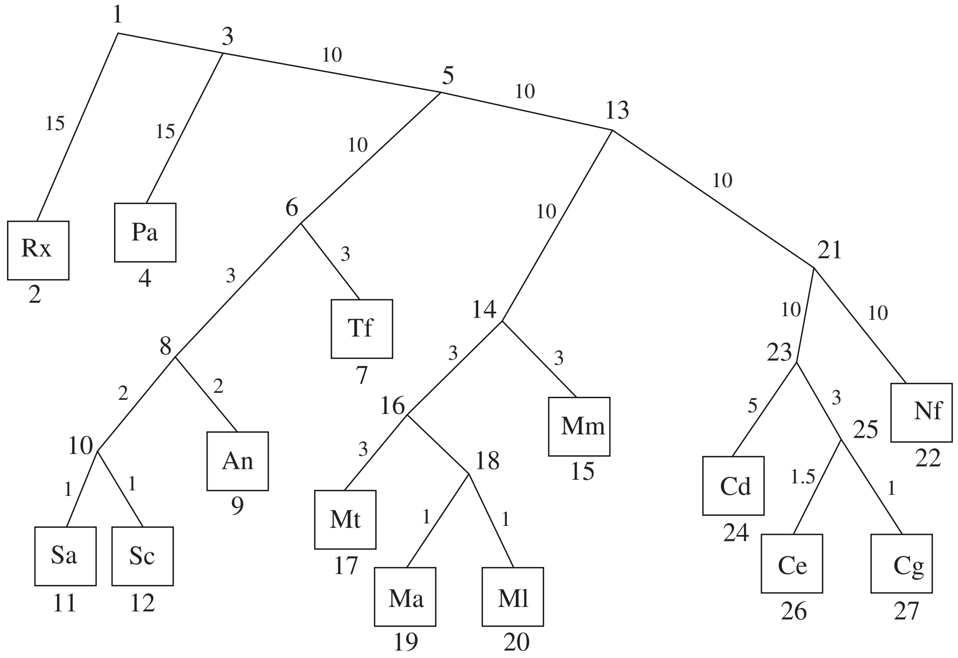


Fig. 5. Phylogenetic tree for the T-box regulation of the gene *ileS* in Actinobacteria.

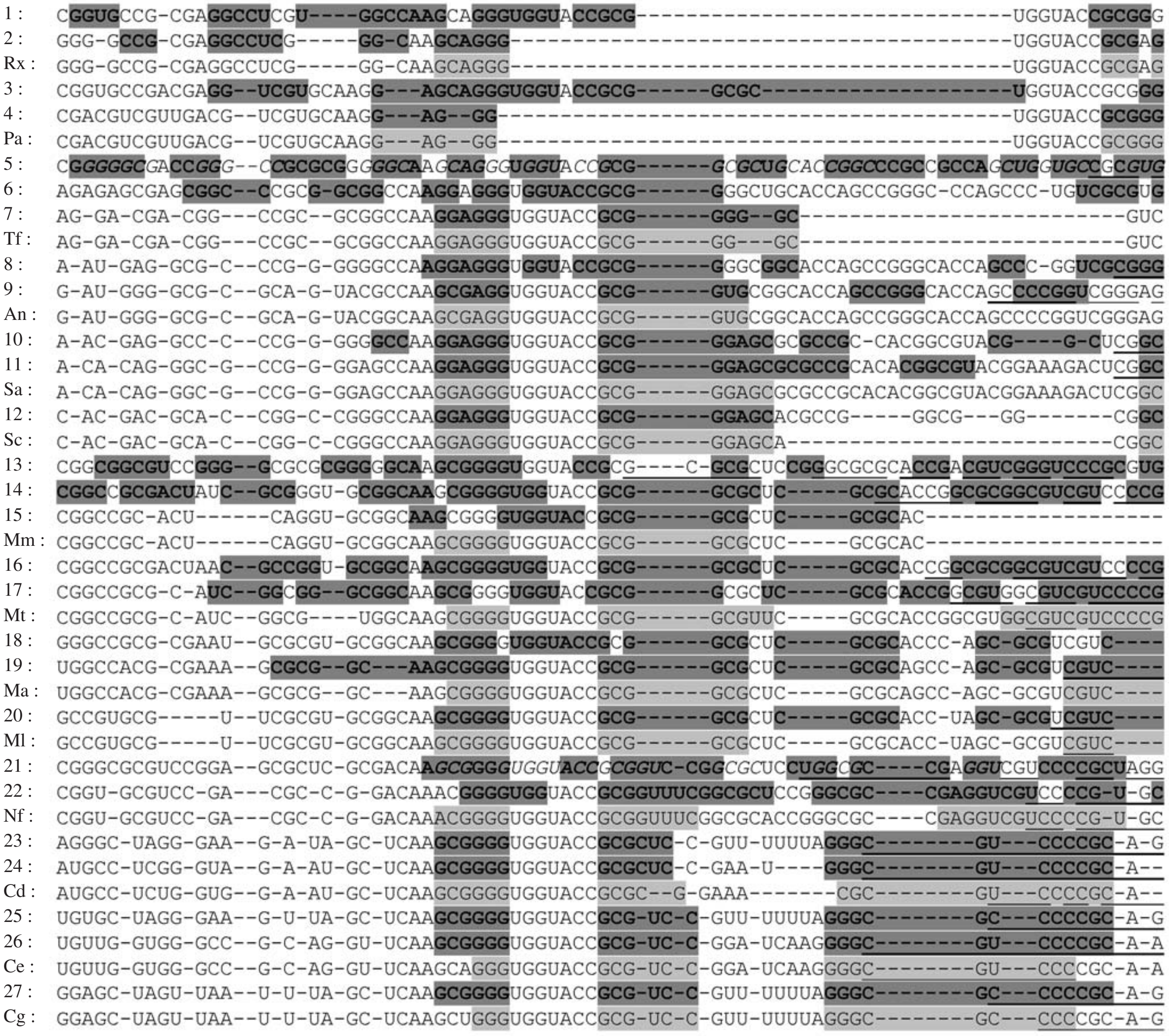
sites and altered nucleotide sequences (see Fig. 4). The designations are the same as in the previous example.

Example 3. Consider the T-box regulation of the *ileS* gene in Actinobacteria. The initial sites are taken from [27]. A standard species tree (Fig. 5) is used. Abbreviations: An for *Actinomyces naeslundii*, Cd for *Corynebacterium diphtheriae*, Ce for *Corynebacterium efficiens*, Cg for *Corynebacterium glutamicum*, Ma for *Mycobacterium avium*, Ml for *Mycobacterium leprae*, Mm for *Mycobacterium marinum*, Mt for *Mycobacterium tuberculosis*, Nf for *Nocardia farcinica*, Pa for *Propionibacterium acnes*, Rx for *Rubrobacter xylanophilus*, Sa for *Streptomyces avermitilis*, Sc for *Streptomyces coelicolor*, and Tf for *Thermobifida fusca.*

The algorithm outputs the final multiple alignment of nucleotide sequences—the assumed regulatory sites and altered nucleotide sequences (see Fig. 6). The antisequester is shown dark gray, and the sequester is underlined. Two variants of regulation are observed at node 5; in the alternative variant the antisequester is italicized and the sequester is shown on a light gray background. An analogous situation is observed at node 21.

Example 4. Consider one more type of regulation—RFN-mediated regulation of the expression of genes involved in riboflavin biosynthesis and transport in Eubacteria (gene *ribB* in BC, EC, PP, and YP and gene *ribD* in the remaining species; abbreviations are below). The initial sites are taken from [37]. A standard species tree (Fig. 7) is used. Abbreviations: BQ for *Bacillus anthracis*, BH for *Bacillus halodurans*, BS for *Bacillus subtilis*, BP for *Burkholderia pseudomallei*, CA for *Clostridium acetobutylicum*, DF for *Clostridium difficile*, DR for *Deinococcus radiodurans*, EC for *Escherichia coli*, LL for *Lactococcus lactis*, PP for *Pseudomonas putida*, SA for *Staphylococcus aureus*, TM for *Thermotoga maritima*, and YP for *Yersinia pestis.*

RFN structure is presented as the structure comprising a helical stem and four helices in its loop, numbered clockwise (helices 1, 2, 3, and 4). Our algorithm outputs a multiple alignment, shown in Fig. 8. The target RFN structure is indicated: the stem is double underlined, helices 1 and 3 are shown on a light gray fine background, helices 2 and 4 are shown on a light gray background, and variable helices are shown on a dark gray or black background. Two helices (1 and 2) can be replaced with one helix; the remaining two helices (3 and 4) can be also replaced with one helix; these alternative helices are underlined. Note that the conserved nucleotides characteristic of RFN structure were mainly aligned



```
1 :   CGGUGCCG-CGAGGCCUCGU----GGCCAAGCAGGGUGGUACCGCG------------------------------UGGUACCGCGGG
2 :   GGG-GCCG-CGAGGCCUCG-----GG-CAAGCAGGG----------------------------------------UGGUACCGCGAG
Rx :  GGG-GCCG-CGAGGCCUCG-----GG-CAAGCAGGG----------------------------------------UGGUACCGCGAG
3 :   CGGUGCCGACGAGG--UCGUGCAAGG---AGCAGGGUGGUACCGCG------GCGC--------------------UGGUACCGCGGG
4 :   CGACGUCGUUGACG--UCGUGCAAGG---AG--GG-----------------------------------------UGGUACCGCGGG
Pa :  CGACGUCGUUGACG--UCGUGCAAGG---AG--GG-----------------------------------------UGGUACCGCGGG
5 :   CGGGGGCGACCGGG--CCGCGCGGGGCAAGCAGGGUGGUACCGCG------GCGCUGCACCGGCCCGCCGCCAGCUGGUGCCGCGUG
6 :   AGAGAGCGAGCGGC--CCGCG-GCGGCCAAGGAGGGUGGUACCGCG------GGGCUGCACCAGCCGGGC-CCAGCCC-UGUCGCGUG
7 :   AG-GA-CGA-CGG---CCGC--GCGGCCAAGGAGGGUGGUACCGCG------GGG--GC--------------------------GUC
Tf :  AG-GA-CGA-CGG---CCGC--GCGGCCAAGGAGGGUGGUACCGCG------GG---GC--------------------------GUC
8 :   A-AU-GAG-GCG-C--CCG-G-GGGGCCAAGGAGGGUGGUACCGCG------GGGCGGCACCAGCCGGGCACCAGCCC-GGUCGCGGG
9 :   G-AU-GGG-GCG-C--GCA-G-UACGCCAAGCGAGGUGGUACCGCG------GUGCGGCACCAGCCGGGCACCAGCCCCGGUCGGGAG
An :  G-AU-GGG-GCG-C--GCA-G-UACGGCAAGCGAGGUGGUACCGCG------GUGCGGCACCAGCCGGGCACCAGCCCCGGUCGGGAG
10 :  A-AC-GAG-GCC-C--CCG-G-GGGGCCAAGGAGGGUGGUACCGCG------GGAGCGCGCCGC-CACGGCGUACG----G-CUCGGC
11 :  A-CA-CAG-GGC-G--CCG-G-GGAGCCAAGGAGGGUGGUACCGCG------GGAGCGCGCCGCACACGGCGUACGGAAAGACUCGGC
Sa :  A-CA-CAG-GGC-G--CCG-G-GGAGCCAAGGAGGGUGGUACCGCG------GGAGCGCGCCGCACACGGCGUACGGAAAGACUCGGC
12 :  C-AC-GAC-GCA-C--CGG-C-CGGGCCAAGGAGGGUGGUACCGCG------GGAGCACGCCG-----GGCG---GG-------CGGC
Sc :  C-AC-GAC-GCA-C--CGG-C-CGGGCCAAGGAGGGUGGUACCGCG------GGAGCA--------------------------CGGC
13 :  CGGCGGCGUCCGGG--GCGCGCGGGGGCAAGCGGGGUGGUACCGCG----C-GCGCUCCGGGCGCGCACCGACGUCGGGUCCCGCGUG
14 :  CGGCCGCGACUAUC--GCGGGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCACCGGCGCGGCGUCGUCCCCG
15 :  CGGCCGC-ACU------CAGGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCAC-------------------
Mm :  CGGCCGC-ACU------CAGGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCAC-------------------
16 :  CGGCCGCGACUAAC--GCCGGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCACCGGCGCGGCGUCGUCCCCG
17 :  CGGCCGCG-C-AUC--GGCGG--GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCACCGGCGUGGCGUCGUCCCCG
Mt :  CGGCCGCG-C-AUC--GGCG----UGGCAAGCGGGGUGGUACCGCG------GCGUUC-----GCGCACCGGCGUGGCGUCGUCCCCG
18 :  GGGCCGCG-CGAAU--GCGCGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCACCC-AGC-GCGUCGUC----
19 :  UGGCCACG-CGAAA--GCGCG--GC---AAGCGGGGUGGUACCGCG------GCGCUC-----GCGCAGCC-AGC-GCGUCGUC----
Ma :  UGGCCACG-CGAAA--GCGCG--GC---AAGCGGGGUGGUACCGCG------GCGCUC-----GCGCAGCC-AGC-GCGUCGUC----
20 :  GCCGUGCG-----U--UCGCGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCACC-UAGC-GCGUCGUC----
Ml :  GCCGUGCG-----U--UCGCGU-GCGGCAAGCGGGGUGGUACCGCG------GCGCUC-----GCGCACC-UAGC-GCGUCGUC----
21 :  CGGGCGCGUCCGGA--GCGCUC-GCGACAAGCGGGGUGGUACCGCGGUC-CGGCGCUCCUGGCGC----CGAGGUCGUCCCCGCUAGG
22 :  CGGU-GCGUCC-GA---CGC-C-G-GACAAACGGGGUGGUACCGCGGUUUCGGCGCUCCGGGCGC----CGAGGUCGUCCCCG-U-GC
Nf :  CGGU-GCGUCC-GA---CGC-C-G-GACAAACGGGGUGGUACCGCGGUUUCGGCGCACCGGGCGC----CGAGGUCGUCCCCG-U-GC
23 :  AGGGC-UAGG-GAA--G-A-UA-GC-UCAAGCGGGGUGGUACCGCGCUC-C-GUU-UUUUAGGGC--------GU---CCCCGC-A-G
24 :  AUGCC-UCGG-GUA--G-A-AU-GC-UCAAGCGGGGUGGUACCGCGCUC-C-GAA-U----GGGC--------GU---CCCCGC-A--
Cd :  AUGCC-UCUG-GUG--G-A-AU-GC-UCAAGCGGGGUGGUACCGCGC--G-GAAA-------CGC--------GU---CCCCGC-A--
25 :  UGUGC-UAGG-GAA--G-U-UA-GC-UCAAGCGGGGUGGUACCGCG-UC-C-GUU-UUUUAGGGC--------GC---CCCCGC-A-G
26 :  UGUUG-GUGG-GCC--G-C-AG-GU-UCAAGCGGGGUGGUACCGCG-UC-C-GGA-UCAAGGGGC--------GU---CCCCGC-A-A
Ce :  UGUUG-GUGG-GCC--G-C-AG-GU-UCAAGCAGGGUGGUACCGCG-UC-C-GGA-UCAAGGGGC--------GU---CCCCGC-A-A
27 :  GGAGC-UAGU-UAA--U-U-UA-GC-UCAAGCGGGGUGGUACCGCG-UC-C-GUU-UUUUAGGGC--------GC---CCCCGC-A-G
Cg :  GGAGC-UAGU-UAA--U-U-UA-GC-UCAAGCUGGGUGGUACCGCG-UC-C-GUU-UUUUAGGGC--------GC---CCCCGC-A-G
```

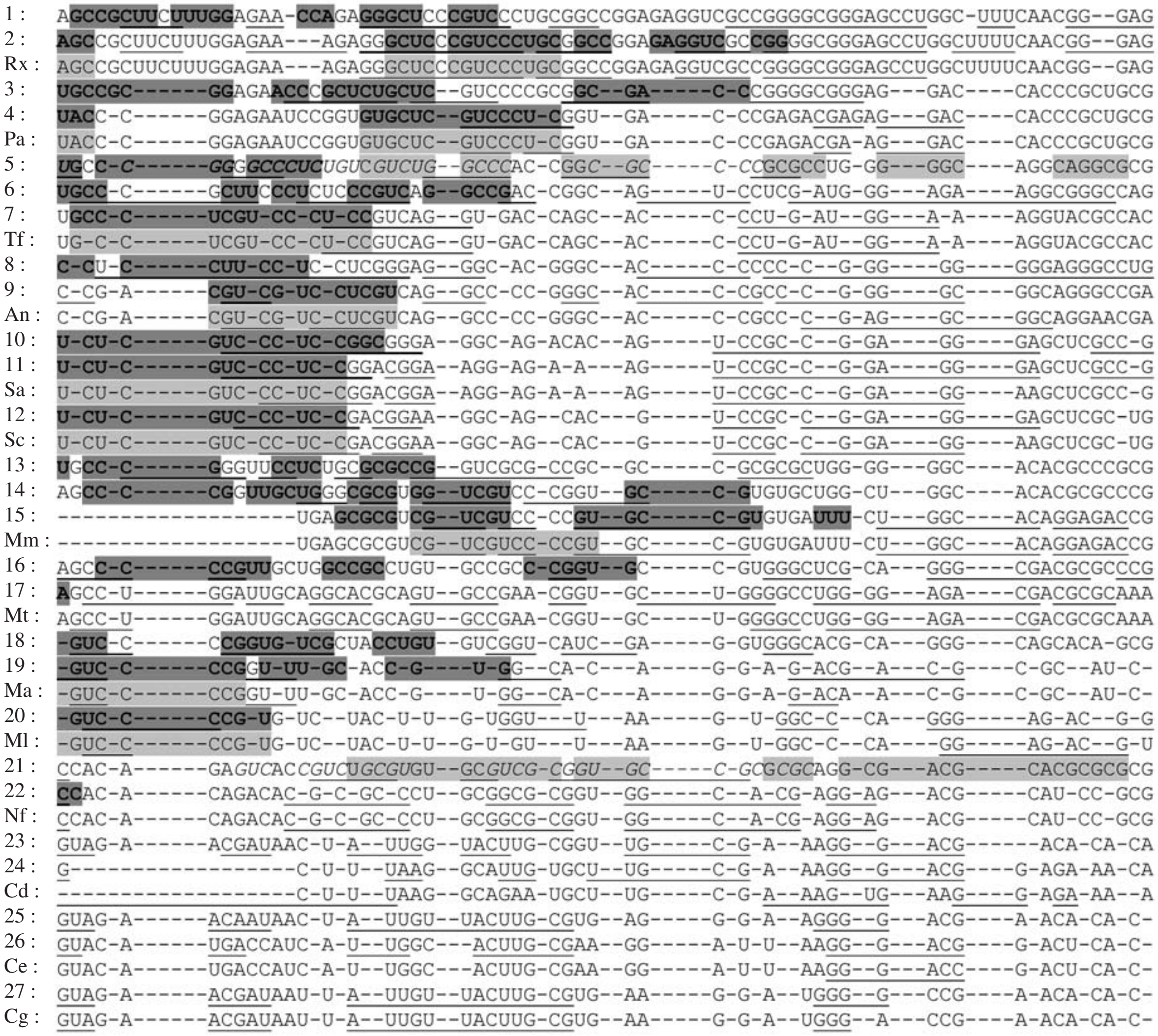
Fig. 6. The multiple alignment (with secondary structure) of assumed regulatory sites generated by the proposed algorithm for the T-box regulation of the gene *ileS* in Actinobacteria.

by columns. In ancestor 19, helix 4 has an alternative (italicized), which continues in the progenies.

DISCUSSION

The goal of this work was to present a new method for the modeling of evolution and multiple alignment of related RNA sequences taking into account their assumed common and coevolving secondary structure as well as the initial testing of this method. This method is based on two assumptions: (1) the initial sequences in leaves have common coevolving secondary structure and (2) the phylogenetic tree in the leaves of which these sequences are specified without the secondary structure or using the secondary structure (it does not assumed known) is considered known.

In the examples described here and for other examples of regulations, our algorithm constructs a reasonable secondary structure at ancestral nodes. It is close to the regulatory structure predicted by bioinformatics and determined experimentally. The secondary structures generated by the algorithm based on the initial sequences in leaves also practically coincide with the known structures. The multiple alignment of primary



```
1 :  AGCCGCUUCUUUGGAGAA-CCAGAGGGCUCCCGUCCCUGCGGCCGGAGAGGUCGCCGGGGCGGGAGCCUGGC-UUUCAACGG--GAG
2 :  AGCCGCUUCUUUGGAGAA---AGAGGGCUCCCGUCCCUGCGGCCGGAGAGGUCGCCGGGGCGGGAGCCUGGCUUUUCAACGG--GAG
Rx : AGCCGCUUCUUUGGAGAA---AGAGGGCUCCCGUCCCUGCGGCCGGAGAGGUCGCCGGGGCGGGAGCCUGGCUUUUCAACGG--GAG
3 :  UGCCGC------GGAGAACCCGCUCUGCUC--GUCCCCGCGGC--GA-----C-CCGGGGCGGGAG---GAC----CACCCGCUGCG
4 :  UACC-C------GGAGAAUCCGGUGUGCUC--GUCCCU-CGGU--GA-----C-CCGAGACGAGAG---GAC----CACCCGCUGCG
Pa : UACC-C------GGAGAAUCCGGUGUGCUC--GUCCCU-CGGU--GA-----C-CCGAGACGA-AG---GAC----CACCCGCUGCG
5 :  UGCC-C------GGGGCCCUCUGUCGUCUG--GCCCAC-CGGC--GC-----C-CCGCGCCUG-GG---GGC----AGGCAGGCGCG
6 :  UGCC-C------GCUUCCCUCUCCCGUCAG--GCCGAC-CGGC--AG-----U-CCUCG-AUG-GG---AGA----AGGCGGGCCAG
7 :  UGCC-C------UCGU-CC-CU-CCGUCAG--GU-GAC-CAGC--AC-----C-CCU-G-AU--GG---A-A----AGGUACGCCAC
Tf : UG-C-C------UCGU-CC-CU-CCGUCAG--GU-GAC-CAGC--AC-----C-CCU-G-AU--GG---A-A----AGGUACGCCAC
8 :  C-CU-C------CUU-CC-UC-CUCGGGAG--GGC-AC-GGGC--AC-----C-CCCC-C--G-GG----GG----GGGAGGGCCUG
9 :  C-CG-A------CGU-CG-UC-CUCGUCAG--GCC-CC-GGGC--AC-----C-CGCC-C--G-GG----GC----GGCAGGGCCGA
An : C-CG-A------CGU-CG-UC-CUCGUCAG--GCC-CC-GGGC--AC-----C-CGCC-C--G-AG----GC----GGCAGGAACGA
10 : U-CU-C------GUC-CC-UC-CGGCGGGA--GGC-AG-ACAC--AG-----U-CCGC-C--G-GA----GG----GAGCUCGCC-G
11 : U-CU-C------GUC-CC-UC-CGGACGGA--AGG-AG-A-A---AG-----U-CCGC-C--G-GA----GG----GAGCUCGCC-G
Sa : U-CU-C------GUC-CC-UC-CGGACGGA--AGG-AG-A-A---AG-----U-CCGC-C--G-GA----GG----AAGCUCGCC-G
12 : U-CU-C------GUC-CC-UC-CGACGGAA--GGC-AG--CAC---G-----U-CCGC-C--G-GA----GG----GAGCUCGC-UG
Sc : U-CU-C------GUC-CC-UC-CGACGGAA--GGC-AG--CAC---G-----U-CCGC-C--G-GA----GG----AAGCUCGC-UG
13 : UGCC-C------GGGUUCCUCUGCGCGCCG--GUCGCG-CCGC--GC-----C-GCGCGCUGG-GG---GGC----ACACGCCCGCG
14 : AGCC-C------CGGUUGCUGGGCGCGUGG--UCGUCC-CGGU--GC-----C-GUGUGCUGG-CU---GGC----ACACGCGCCCG
15 : -------------------UGAGCGCGUCG--UCGUCC-CCGU--GC-----C-GUGUGAUUU-CU---GGC----ACAGGAGACCG
Mm : -------------------UGAGCGCGUCG--UCGUCC-CCGU--GC-----C-GUGUGAUUU-CU---GGC----ACAGGAGACCG
16 : AGCC-C------CCGUUGCUGGCCGCCUGU--GCCGCC-CGGU--GC-----C-GUGGGCUCG-CA---GGG----CGACGCGCCCG
17 : AGCC-U------GGAUUGCAGGCACGCAGU--GCCGAA-CGGU--GC-----U-GGGGCCUGG-GG---AGA----CGACGCGCAAA
Mt : AGCC-U------GGAUUGCAGGCACGCAGU--GCCGAA-CGGU--GC-----U-GGGGCCUGG-GG---AGA----CGACGCGCAAA
18 : -GUC-C------CCGGUG-UCGCUACCUGU--GUCGGU-CAUC--GA-----G-GUGGGCACG-CA---GGG----CAGCACA-GCG
19 : -GUC-C------CCGGU-UU-GC-ACC-G----U-GG--CA-C---A-----G-G-A-G-ACG--A---C-G----C-GC--AU-C-
Ma : -GUC-C------CCGGU-UU-GC-ACC-G----U-GG--CA-C---A-----G-G-A-G-ACA--A---C-G----C-GC--AU-C-
20 : -GUC-C------CCG-UG-UC--UAC-U-U--G-UGGU---U---AA-----G--U-GGC-C--CA---GGG-----AG-AC--G-G
Ml : -GUC-C------CCG-UG-UC--UAC-U-U--G-U-GU---U---AA-----G--U-GGC-C--CA----GG-----AG-AC--G-U
21 : CCAC-A------GAGUCACCGUCUGCGUGU--GCGUCG-CGGU--GC-----C-GCGCGCAGG-CG---ACG-----CACGCGCGCG
22 : CCAC-A------CAGACAC-G-C-GC-CCU--GCGGCG-CGGU--GG-----C--A-CG-AGG-AG---ACG-----CAU-CC-GCG
Nf : CCAC-A------CAGACAC-G-C-GC-CCU--GCGGCG-CGGU--GG-----C--A-CG-AGG-AG---ACG-----CAU-CC-GCG
23 : GUAG-A------ACGAUAAC-U-A--UUGG--UACUUG-CGGU--UG-----C-G-A--AAGG--G---ACG------ACA-CA-CA
24 : G------------------C-U-U--UAAG--GCAUUG-UGCU--UG-----C-G-A--AAGG--G---ACG----G-AGA-AA-CA
Cd : -------------------C-U-U--UAAG--GCAGAA-UGCU--UG-----C-G-A--AAG--UG---AAG----G-AGA-AA--A
25 : GUAG-A------ACAAUAAC-U-A--UUGU--UACUUG-CGUG--AG-----G-G-A--AGGG--G---ACG----A-ACA-CA-C-
26 : GUAC-A------UGACCAUC-A-U--UGGC---ACUUG-CGAA--GG-----A-U-U--AAGG--G---ACG----G-ACU-CA-C-
Ce : GUAC-A------UGACCAUC-A-U--UGGC---ACUUG-CGAA--GG-----A-U-U--AAGG--G---ACC----G-ACU-CA-C-
27 : GUAG-A------ACGAUAAU-U-A--UUGU--UACUUG-CGUG--AA-----G-G-A--UGGG--G---CCG----A-ACA-CA-C-
Cg : GUAG-A------ACGAUAAU-U-A--UUGU--UACUUG-CGUG--AA-----G-G-A--UGGG--A---CCG----A-ACA-CA-C-
```

Fig. 6. Contd.

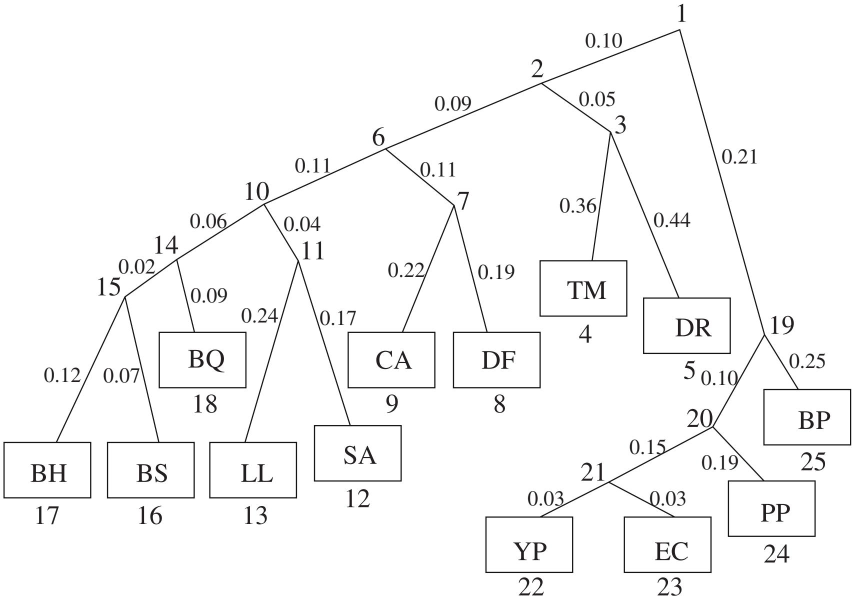
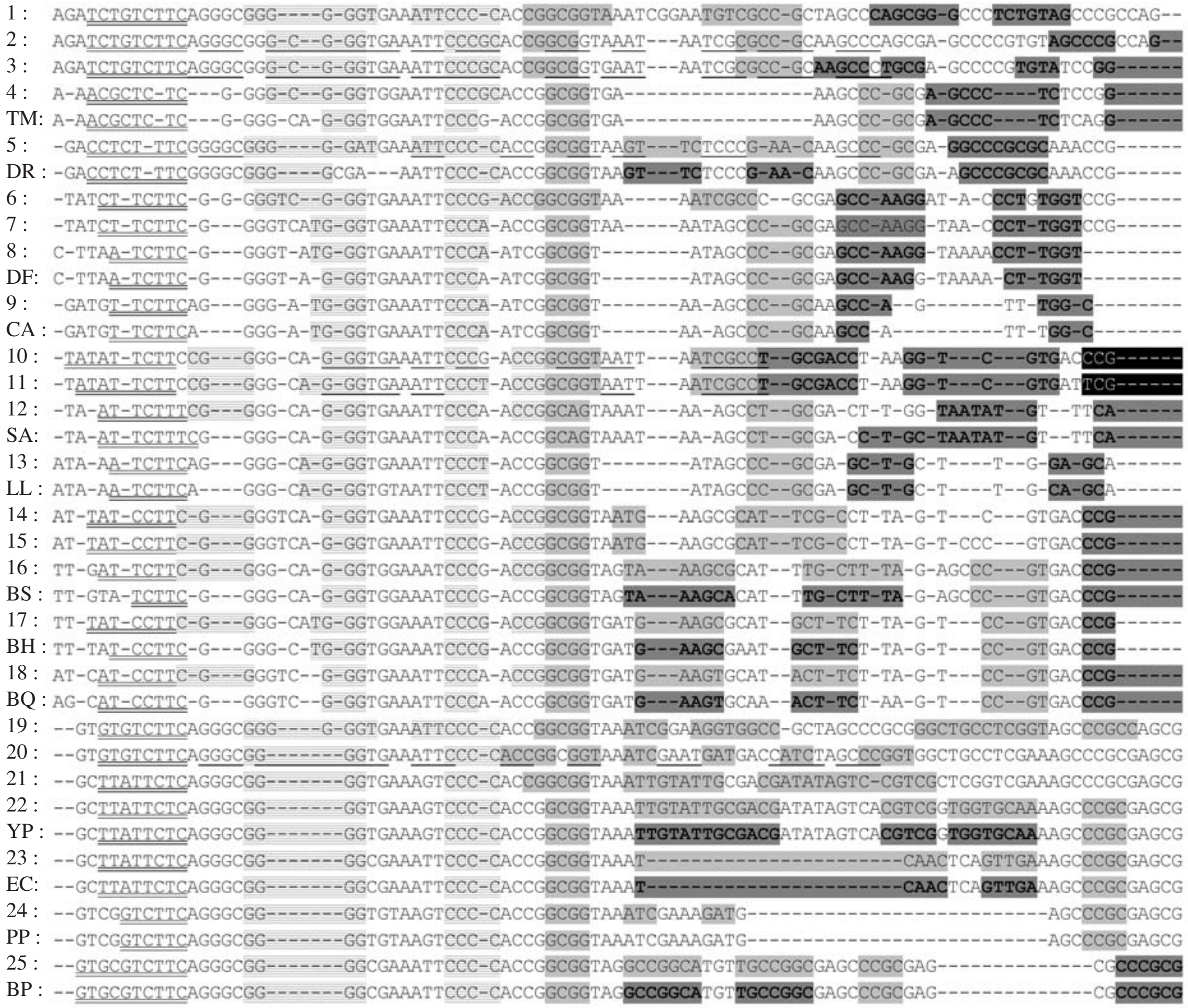


Fig. 7. Phylogenetic tree of Eubacteria.



```
1 :  AGATCTGTCTTCAGGGCGGG----G-GGTGAAATTCCC-CACCGGCGGTAAATCGGAATGTCGCC-GCTAGCCCAGCGG-GCCCTCTGTAGCCCGCCAG--
2 :  AGATCTGTCTTCAGGGCGGG-C--G-GGTGAAATTCCCGCACCGGCGGTAAAT---AATCGCGCC-GCAAGCCCAGCGA-GCCCCGTGTAGCCCGCCAG--
3 :  AGATCTGTCTTCAGGGCGGG-C--G-GGTGAAATTCCCGCACCGGCGGTGAAT---AATCGCGCC-GCAAGCCCTGCGA-GCCCCGTGTATCCGG------
4 :  A-AACGCTC-TC---G-GGG-C--G-GGTGGAATTCCCGCACCGGCGGTGA-----------------AAGCCC-GCGA-GCCC----TCTCCGG------
TM:  A-AACGCTC-TC---G-GGG-CA-G-GGTGGAATTCCCG-ACCGGCGGTGA-----------------AAGCCC-GCGA-GCCC----TCTCAGG------
5 :  -GACCTCT-TTCGGGGCGGG----G-GATGAAATTCCC-CACCGGCGGTAAGT---TCTCCCG-AA-CAAGCCC-GCGA-GGCCCGCGCAAACCG------
DR : -GACCTCT-TTCGGGGCGGG----GCGA---AATTCCC-CACCGGCGGTAAGT---TCTCCCG-AA-CAAGCCC-GCGA-AGCCCGCGCAAACCG------
6 :  -TATCT-TCTTC-G-G-GGGTC--G-GGTGAAATTCCCG-ACCGGCGGTAA-----AATCGCCC--GCGAGCC-AAGGAT-A-CCCTGTGGTCCG------
7 :  -TATCT-TCTTC-G---GGGTCATG-GGTGAAATTCCCA-ACCGGCGGTAA-----AATAGCCC--GCGAGCC-AAGG-TAA-CCCT-TGGTCCG------
8 :  C-TTAA-TCTTC-G---GGGT-ATG-GGTGAAATTCCCA-ATCGGCGGT--------ATAGCCC--GCGAGCC-AAGG-TAAACCT-TGGT---------
DF:  C-TTAA-TCTTC-G---GGGT-A-G-GGTGAAATTCCCA-ATCGGCGGT--------ATAGCCC--GCGAGCC-AAGG-TAAAA-CT-TGGT---------
9 :  -GATGT-TCTTCAG---GGG-A-TG-GGTGAAATTCCCA-ATCGGCGGT-------AA-AGCCC--GCAAGCC-A--G-------TT-TGG-C--------
CA : -GATGT-TCTTCA----GGG-A-TG-GGTGAAATTCCCA-ATCGGCGGT-------AA-AGCCC--GCAAGCC-A---------TT-TGG-C--------
10 : -TATAT-TCTTCCG---GGG-CA-G-GGTGAAATTCCCG-ACCGGCGGTAATT---AATCGCCT--GCGACCT-AAGG-T---C---GTGACCCG------
11 : -TATAT-TCTTCCG---GGG-CA-G-GGTGAAATTCCCT-ACCGGCGGTAATT---AATCGCCT--GCGACCT-AAGG-T---C---GTGATTCG------
12 : -TA-AT-TCTTTCG---GGG-CA-G-GGTGAAATTCCCA-ACCGGCAGTAAAT---AA-AGCCT--GCGA-CT-T-GG-TAATAT--GT--TTCA------
SA:  -TA-AT-TCTTTCG---GGG-CA-G-GGTGAAATTCCCA-ACCGGCAGTAAAT---AA-AGCCT--GCGA-CC-T-GC-TAATAT--GT--TTCA------
13 : ATA-AA-TCTTCAG---GGG-CA-G-GGTGAAATTCCCT-ACCGGCGGT--------ATAGCCC--GCGA-GC-T-GC-T----T--G-GA-GCA------
LL : ATA-AA-TCTTCA----GGG-CA-G-GGTGTAATTCCCT-ACCGGCGGT--------ATAGCCC--GCGA-GC-T-GC-T----T--G-CA-GCA------
14 : AT-TAT-CCTTC-G---GGGTCA-G-GGTGAAATTCCCG-ACCGGCGGTAATG---AAGCGCAT--TCG-CCT-TA-G-T---C---GTGACCCG------
15 : AT-TAT-CCTTC-G---GGGTCA-G-GGTGAAATTCCCG-ACCGGCGGTAATG---AAGCGCAT--TCG-CCT-TA-G-T-CCC---GTGACCCG------
16 : TT-GAT-TCTTC-G---GGG-CA-G-GGTGGAAATCCCG-ACCGGCGGTAGTA---AAGCGCAT--TTG-CTT-TA-G-AGCCC---GTGACCCG------
BS : TT-GTA-TCTTC-G---GGG-CA-G-GGTGGAAATCCCG-ACCGGCGGTAGTA---AAGCACAT--TTG-CTT-TA-G-AGCCC---GTGACCCG------
17 : TT-TAT-CCTTC-G---GGG-CATG-GGTGGAAATCCCG-ACCGGCGGTGATG---AAGCGCAT--GCT-TCT-TA-G-T---CC--GTGACCCG------
BH : TT-TAT-CCTTC-G---GGG-C-TG-GGTGGAAATCCCG-ACCGGCGGTGATG---AAGCGAAT--GCT-TCT-TA-G-T---CC--GTGACCCG------
18 : AT-CAT-CCTTC-G---GGGTC--G-GGTGAAATTCCCA-ACCGGCGGTGATG---AAGTGCAT--ACT-TCT-TA-G-T---CC--GTGACCCG------
BQ : AG-CAT-CCTTC-G---GGGTC--G-GGTGAAATTCCCA-ACCGGCGGTGATG---AAGTGCAA--ACT-TCT-AA-G-T---CC--GTGACCCG------
19 : --GTGTGTCTTCAGGGCGGG----G-GGTGAAATTCCC-CACCGGCGGTAAATCGGAAGGTGGCC-GCTAGCCCGCGGGCTGCCTCGGTAGCCCGCCAGCG
20 : --GTGTGTCTTCAGGGCGG-------GGTGAAATTCCC-CACCGGCGGTAAATCGAATGATGACCATCTAGCCCGGTGGCTGCCTCGAAAGCCCGCGAGCG
21 : --GCTTATTCTCAGGGCGG-------GGTGAAAGTCCC-CACCGGCGGTAAATTGTATTGCGACGATATAGTC-CGTCGCTCGGTCGAAAGCCCGCGAGCG
22 : --GCTTATTCTCAGGGCGG-------GGTGAAAGTCCC-CACCGGCGGTAAATTGTATTGCGACGATATAGTCACGTCGGTGGTGCAAAAGCCCGCGAGCG
YP : --GCTTATTCTCAGGGCGG-------GGTGAAAGTCCC-CACCGGCGGTAAATTGTATTGCGACGATATAGTCACGTCGGTGGTGCAAAAGCCCGCGAGCG
23 : --GCTTATTCTCAGGGCGG-------GGCGAAATTCCC-CACCGGCGGTAAAT----------------------CAACTCAGTTGAAAGCCCGCGAGCG
EC:  --GCTTATTCTCAGGGCGG-------GGCGAAATTCCC-CACCGGCGGTAAAT----------------------CAACTCAGTTGAAAGCCCGCGAGCG
24 : --GTCGGTCTTCAGGGCGG-------GGTGTAAGTCCC-CACCGGCGGTAAATCGAAAGATG--------------------------AGCCCGCGAGCG
PP : --GTCGGTCTTCAGGGCGG-------GGTGTAAGTCCC-CACCGGCGGTAAATCGAAAGATG--------------------------AGCCCGCGAGCG
25 : --GTGCGTCTTCAGGGCGG-------GGCGAAATTCCC-CACCGGCGGTAGGCCGGCATGTTGCCGGCGAGCCCGCGAG-------------CGCCCGCG
BP : --GTGCGTCTTCAGGGCGG-------GGCGAAATTCCC-CACCGGCGGTAGGCCGGCATGTTGCCGGCGAGCCCGCGAG-------------CGCCCGCG
```

Fig. 8. The multiple alignment (with secondary structure) of assumed regulatory sites generated by the proposed algorithm for the RFN-mediated regulation of expression of the genes involved in riboflavin biosynthesis and transport in Eubacteria.

structures in leaves and even along the overall tree has a good quality. We have tested the model adding noise to both artificial and biological examples and observed a stable operation of the algorithm.

The first approach, mentioned in the Problem Statement section and detailed in [32–34, 38], gave the same or very similar secondary structures for the initial data of examples 1–4, despite the fact that these approaches are different. We have tested the obtained ancestral signals using the model of classic attenuation regulation from [26, 35] and the software available at http://lab6.iitp.ru, item 3. This testing has confirmed the functionality of ancestral signals for this regulation.

Comparison with other methods and programs. To compare the result given by our algorithm with the results of other standard algorithms, we applied the known programs, such as PAML and PAUP (http:// evolution.genetics.washington.edu/phylip /software. serv. html), to the same data. When only initial primary structures (without alignment) were input, none of these programs was able to reconstruct the ancestral regulatory elements of the type present in leaves. When the alignment that took into account the secondary structure (for example, given in [22] for classic attenuation regulation) was also input, the result depended on the program used. PAML was unable to predict the secondary structure of the required type in ancestral sequences. PAUP was able

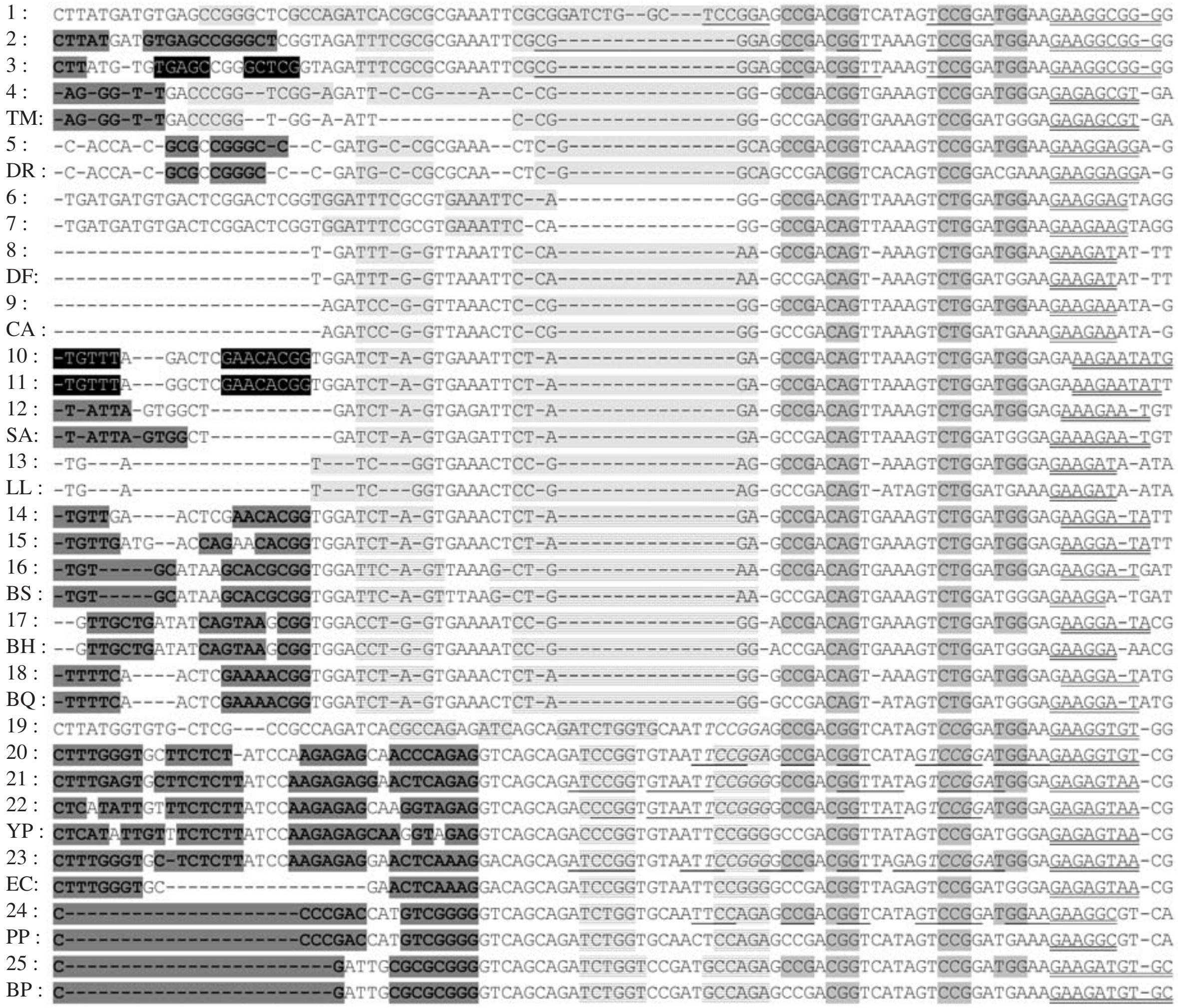

1 : CTTATGATGTGAGCCGGGCTCGCCAGATCACGCGCGAAATTCGCGGATCTG--GC---TCCGGAGCCGACGGTCATAGTCCGGATGGAAGAAGGCGG-GG
2 : CTTATGATGTGAGCCGGGCTCGGTAGATTTCGCGCGAAATTCGCG----------------GGAGCCGACGGTTAAAGTCCGGATGGAAGAAGGCGG-GG
3 : CTTATG-TGTGAGCCGGGCTCGGTAGATTTCGCGCGAAATTCGCG----------------GGAGCCGACGGTTAAAGTCCGGATGGAAGAAGGCGG-GG
4 : -AG-GG-T-TGACCCGG--TCGG-AGATT-C-CG----A--C-CG----------------GG-GCCGACGGTGAAAGTCCGGATGGAGAGAGCGT-GA
TM: -AG-GG-T-TGACCCGG--T-GG-A-ATT------------C-CG----------------GG-GCCGACGGTGAAAGTCCGGATGGGAGAGAGCGT-GA
5 : -C-ACCA-C-GCGCCGGGC-C--C-GATG-C-CGCGAAA--CTC-G---------------GCAGCCGACGGTCAAAGTCCGGATGGAAGAAGGAGGA-G
DR : -C-ACCA-C-GCGCCGGGC-C--C-GATG-C-CGCGCAA--CTC-G---------------GCAGCCGACGGTCACAGTCCGGACGAAAGAAGGAGGA-G
6 : -TGATGATGTGACTCGGACTCGGTGGATTTCGCGTGAAATTC--A----------------GG-GCCGACAGTTAAAGTCTGGATGGAAGAAGGAGTAGG
7 : -TGATGATGTGACTCGGACTCGGTGGATTTCGCGTGAAATTC-CA----------------GG-GCCGACAGTTAAAGTCTGGATGGAAGAAGAAGTAGG
8 : -----------------------T-GATTT-G-GTTAAATTC-CA----------------AA-GCCGACAGT-AAAGTCTGGATGGAAGAAGATAT-TT
DF: -----------------------T-GATTT-G-GTTAAATTC-CA----------------AA-GCCGACAGT-AAAGTCTGGATGGAAGAAGATAT-TT
9 : ------------------------AGATCC-G-GTTAAACTC-CG----------------GG-GCCGACAGTTAAAGTCTGGATGGAAGAAGAAATA-G
CA : ------------------------AGATCC-G-GTTAAACTC-CG----------------GG-GCCGACAGTTAAAGTCTGGATGAAAGAAGAAATA-G
10 : -TGTTTA---GACTCGAACACGGTGGATCT-A-GTGAAATTCT-A----------------GA-GCCGACAGTTAAAGTCTGGATGGGAGAAAGAATATG
11 : -TGTTTA---GGCTCGAACACGGTGGATCT-A-GTGAAATTCT-A----------------GA-GCCGACAGTTAAAGTCTGGATGGGAGAAAGAATATT
12 : -T-ATTA-GTGGCT-----------GATCT-A-GTGAGATTCT-A----------------GA-GCCGACAGTTAAAGTCTGGATGGGAGAAAGAA-TGT
SA: -T-ATTA-GTGGCT-----------GATCT-A-GTGAGATTCT-A----------------GA-GCCGACAGTTAAAGTCTGGATGGGAGAAAGAA-TGT
13 : -TG---A----------------T---TC---GGTGAAACTCC-G----------------AG-GCCGACAGT-AAAGTCTGGATGGGAGAAGATA-ATA
LL : -TG---A----------------T---TC---GGTGAAACTCC-G----------------AG-GCCGACAGT-ATAGTCTGGATGAAAGAAGATA-ATA
14 : -TGTTGA----ACTCGAACACGGTGGATCT-A-GTGAAACTCT-A----------------GA-GCCGACAGTGAAAGTCTGGATGGGAGAAGGA-TATT
15 : -TGTTGATG--ACCAGAACACGGTGGATCT-A-GTGAAACTCT-A----------------GA-GCCGACAGTGAAAGTCTGGATGGGAGAAGGA-TATT
16 : -TGT-----GCATAAGCACGCGGTGGATTC-A-GTTAAAG-CT-G----------------AA-GCCGACAGTGAAAGTCTGGATGGGAGAAGGA-TGAT
BS : -TGT-----GCATAAGCACGCGGTGGATTC-A-GTTTAAG-CT-G----------------AA-GCCGACAGTGAAAGTCTGGATGGGAGAAGGA-TGAT
17 : --GTTGCTGATATCAGTAAGCGGTGGACCT-G-GTGAAAATCC-G----------------GG-ACCGACAGTGAAAGTCTGGATGGGAGAAGGA-TACG
BH : --GTTGCTGATATCAGTAAGCGGTGGACCT-G-GTGAAAATCC-G----------------GG-ACCGACAGTGAAAGTCTGGATGGGAGAAGGA-AACG
18 : -TTTTCA----ACTCGAAAACGGTGGATCT-A-GTGAAACTCT-A----------------GG-GCCGACAGT-AAAGTCTGGATGGGAGAAGGA-TATG
BQ : -TTTTCA----ACTCGAAAACGGTGGATCT-A-GTGAAACTCT-A----------------GG-GCCGACAGT-ATAGTCTGGATGGGAGAAGGA-TATG
19 : CTTATGGTGTG-CTCG---CCGCCAGATCACGCCAGAGATCAGCAGATCTGGTGCAATTCCGGAGCCGACGGTCATAGTCCGGATGGAAGAAGGTGT-GG
20 : CTTTGGGTGCTTCTCT-ATCCAAGAGAGCAACCCAGAGGTCAGCAGATCCGGTGTAATTCCGGAGCCGACGGTCATAGTCCGGATGGAAGAAGGTGT-CG
21 : CTTTGAGTGCTTCTCTTATCCAAGAGAGGAACTCAGAGGTCAGCAGATCCGGTGTAATTCCGGGGCCGACGGTTATAGTCCGGATGGGAGAGAGTAA-CG
22 : CTCATATTGTTTCTCTTATCCAAGAGAGCAAGGTAGAGGTCAGCAGACCCGGTGTAATTCCGGGGCCGACGGTTATAGTCCGGATGGGAGAGAGTAA-CG
YP : CTCATATTGTTCTCTTATCCAAGAGAGCAAGGTAGAGGTCAGCAGACCCGGTGTAATTCCGGGGCCGACGGTTATAGTCCGGATGGGAGAGAGTAA-CG
23 : CTTTGGGTGC-TCTCTTATCCAAGAGAGGAACTCAAAGGACAGCAGATCCGGTGTAATTCCGGGGCCGACGGTTAGAGTCCGGATGGGAGAGAGTAA-CG
EC: CTTTGGGTGC------------------GAACTCAAAGGACAGCAGATCCGGTGTAATTCCGGGGCCGACGGTTAGAGTCCGGATGGGAGAGAGTAA-CG
24 : C---------------------CCCGACCATGTCGGGGGTCAGCAGATCTGGTGCAATTCCAGAGCCGACGGTCATAGTCCGGATGGAAGAAGGCGT-CA
PP : C---------------------CCCGACCATGTCGGGGGTCAGCAGATCTGGTGCAACTCCAGAGCCGACGGTCATAGTCCGGATGAAAGAAGGCGT-CA
25 : C------------------------GATTGCGCGCGGGGTCAGCAGATCTGGTCCGATGCCAGAGCCGACGGTCATAGTCCGGATGGAAGAAGATGT-GC
BP : C------------------------GATTGCGCGCGGGGTCAGCAGATCTGGTCCGATGCCAGAGCCGACGGTCATAGTCCGGATGAAAGAAGATGT-GC

Fig. 8. Contd.

to predict such structure; however, it was not conserved along the tree edges.

ACKNOWLEDGMENTS

We are grateful to the reviewer, whose profound criticism allowed the paper to be considerably improved.

REFERENCES

1. Nei M., Kumar S. 2000. *Molecular Evolution and Phylogenetics*. Oxford: Oxford Univ. Press.
2. Gascuel O., Steel M. 2007. *Reconstructing Evolution: New Mathematical and Computational Advances*. Oxford: Oxford Univ. Press.
3. Page R.D.M., Holmes E.C. 1998. *Molecular Evolution: A Phylogenetic Approach*. Oxford: Blackwell Publishing.
4. Wolf Y., Rogozin I., Grishin N., Tatusov R., Koonin E. 2001. Genome trees constructed using five different approaches suggest new major bacterial clades. *BMC Evol. Biol*. **1**, 1–22.
5. Durand D., Haldorsson B.V., Vernot B. 2006. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *J. Comput. Biol*. **13**, 320–335.
6. Gascuel O. (Ed.). 2004. *Mathematics of Evolution and Phylogeny*. Oxford: Oxford Univ. Press.
7. Felsenstein J. 2004. *Inferring phylogenies*. Sunderland, MA: Sinauer Assoc.
8. Nakhleh L., Warnov T., Linder C.R. 2004. Reconstructing reticulate evolution in species: Theory and practice. In: *Proc 8th Annual Conference on Research in Computational Molecular Biology*. ACM, pp. 337–346.
9. Mirkin B.G., Fenner T.I., Galperin M.Y., Koonin E.V. 2003. Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene

transfer in the evolution of prokaryotes. *BMC Evol. Biol.* **3**, 1–34.

10. Guigo R., Muchnik I., Smith T. 1996. Reconstruction of ancient molecular phylogeny. *Mol. Phylog. Evol.* **6**, 189–213.

11. Page R.D.M., Charlstone M.A. 1997. From gene to organismal phylogeny: Reconciled trees and gene tree/species tree problem. *Mol. Phylog. Evol.* **7**, 231–240.

12. Page R.D.M. 1998. GeneTree: Comparing gene and species phylogenies using reconciled trees. *Bioinformatics.* **14**, 819–820.

13. Zmasek C.M., Eddy S.R. 2001. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics.* **17**, 821–828.

14. Chauve C., Doyon J.-P., El-Mabrouk N. 2007. Inferring a duplication, speciation and loss history from a gene tree (extended abstract). In: *Comparative Genomics, RECOMB 2007 International Workshop*. Eds. Tesler G., Durand D., *LNCS (LNBI)*, vol. 4751, Heidelberg: Springer, pp. 45–57.

15. Elias I., Tuller T. 2007. Reconstruction of ancestral genomic sequences using likelihood. *J. Comput. Biol.* **14**, 216–237.

16. Hudek A.K., Brown D.G. 2005. Ancestral sequence alignment under optimal conditions. *BMC Bioinformatics.* **6**, 1–14.

17. Hallett M.T., Lagergren J. 2000. New algorithms for the duplication-loss model. In: *Proc. 4th Annual International Conference on Computational Molecular Biology, RECOMB 2000*. ACM, pp. 138–146.

18. Berglung A.-C., Lagergren J., Sennblad B. 2004. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In: *Proc 8th Annual International Conference on Research in Computational Molecular Biology, RECOMB*. Eds Bourne P.E., Gusfield D. ACM, pp. 326–335.

19. Bonizzoni P., Della Vedova G., Dondi R. 2005. Reconciling a gene tree to a species tree under the duplication cost model. *Theor. Comput. Sci.* **347**, 36–53.

20. Gorecki P., Tiutyn J. 2006. DLS-trees: A model of evolutionary scenarios.*Theor. Comput. Sci.* **359**, 378–399.

21. Lyubetsky V.A., Gorbunov K.Yu., Rusin L.Y., V'yugin V.V. 2005. Algorithms to reconstruct evolutionary events at molecular level and infer species phylogeny. In: *Bioinformatics of Genome Regulation and Structure II*. Springer Science & Business Media, Inc., pp. 189–204.

22. Vitreschak A.G., Lyubetskaya E.V., Shirshin M.A., Gelfand M.S., Lyubetsky V.A. 2004. Attenuation regulation of amino acid biosynthetic operons in proteobacteria: Comparative genomics analysis. *FEMS Microbiol. Lett.* **234**, 357–370.

23. Gelfand M.S., Gerasimova A.V., Kotelnikova E.A., Laikova O.N., Makeev V.Y., Mironov A.A., Panina E.M., Ravcheev D.A., Rodionov D.A., Vitreschak A.G. 2005. Comparative genomics and evolution of bacterial regulatory systems. In: *Bioinformatics of Genome Regulation and Structure II*. Springer Science & Business Media, Inc., pp. 111–119.

24. Seliverstov A.V., Putzer H., Gelfand M.S., Lyubetsky V.A. 2005. Comparative analysis of RNA regulatory elements of amino acid metabolism genes in Actinobacteria. *BMC Microbiol.* **5**, 1–14.

25. Seliverstov A.V., Lyubetsky V.A. 2006. Translation regulation of intron containing genes in chloroplasts. *J. Bioinform. Comp. Biol.* **4**, 783–793.

26. Lyubetsky V.A., Pirogov S.A., Rubanov L.I., Seliverstov A.V. 2007. Modeling classic attenuation regulation of gene expression in bacteria. *J. Bioinform. Comp. Biol.* **5**, 155–180.

27. Vitreschak A.G., Mironov A.A., Lyubetsky V.A., Gelfand M.S. 2008. Comparative genomic analysis of T-box regulatory systems in bacteria. *RNA.* **14**, 717–735.

28. McAdams H.H., Srinivasan B., Arkin A.P. 2004. The evolution of genetic regulatory systems in bacteria. *Nature Rev. Genet.* **5**, 169–178.

29. Savill N.J., Hoyle D.C., Higgs P.G. 2001. RNA sequence evolution with secondary structure constraints: Comparison of substitution rate models using maximum-likelihood methods. *Genetics.* **157**, 399–411.

30. Kosakovsky Pond S.L., Mannino F.V., Gravenor M.B., Muse S.V., Frost S.D.W. 2007. Evolutionary model selection with a genetic algorithm: A case study using stem RNA. *Mol. Biol. Evol.* **24**, 159–170.

31. Fischer W., Geard N. Reconstructing phylogeny from RNA secondary structure via simulated evolution. http://www.itee.uq.edu.au/nic/papers/csss-rna.pdf.

32. Lyubetsky V., Zhizhina E., Rubanov L. 2008. Gibbs field approach to the problem of evolution of biological sequences. *Probl. Pered. Inform.* (in press).

33. Gorbunov K.Yu., Lyubetsky V.A. 2007. Modeling evolution of the nucleotide sequence with secondary structure. In: *Proc. Computational Phylogenetics and Molecular Systematics: CPMS'2007*. Moscow: KMK Scientific Press, pp. 68–75.

34. Lyubetsky V.A., Seliverstov A.V., Gorbunov K.Yu. 2007. Models of gene expression regulation and evolution of regulatory elements. In: *Proc. Computational Phylogenetics and Molecular Systematics: CPMS'2007*. Moscow: KMK Scientific Press, pp. 158–165.

35. Asarin E., Cachat Th., Seliverstov A.V., Touili T., Lyubetsky V.A. 2007. Attenuation regulation as a term rewriting system. In: *Algebraic Biology. LNCS (LNBI)*, vol. 4545, Springer, pp. 81–94.

36. Gorbunov K.Yu., Mironov A.A., Lyubetsky V.A. 2003. Search for conserved secondary structures of RNA. *Mol. Biol.* **37**, 850–860.

37. Vitreschak A.G., Rodionov D.A., Mironov A.A., Gelfand M.S. 2002. Regulation of riboavin biosynthesis and transport genes in bacteria by transcriptional and translational attenuation. *Nucleic Acids Res.* **30**, 3141–3151.

38. Gorbunov K.Yu., Lyubetsky V.A. 2007. Reconstruction of ancestral regulatory signals along a transcription factor tree. *Mol. Biol.* **41**, 918–925.