DOI 10.3233/ISB-2010-0425
IOS Press

Evolution of Regulatory Motifs of Bacterial Transcription Factors

Konstantin Y. Gorbunov[a,*], Olga N. Laikova[b], Dmitry A. Rodionov[a,c], Mikhail S. Gelfand[a] and Vassily A. Lyubetsky[a]

[a] *Institute for Information Transmission Problems of the Russian Academy of Sciences* (*Kharkevich Institute*), *Moscow, Russia*
[b] *State Research Institute for Genetics and Selection of Industrial Microorganisms, Moscow, Russia*
[c] *Burnham Institute, La Jolla, CA, USA*

Edited by E. Wingender; received 18 August 2009; revised 16 November 2009; accepted 23 December 2009; published 9 March 2010

ABSTRACT: Unlike evolution of genes and proteins, evolution of regulatory systems is a relatively new area of research. In particular, little systematic study has been done on evolution of DNA binding motifs in transcription factor families. We suggest an algorithm that reconstructs the most parsimonious scenario for changes in DNA binding motifs along an evolutionary tree of transcription factor binding sites. The algorithm was validated on several artificial datasets and then applied to reconstruct the evolutionary history of the NrdR, MntR, LacI, FNR, Irr, Fur and Rrf2 transcription factor families. The algorithm seems to be sufficiently robust to be applicable in realistic situations. In most transcription factor families the changes in binding motifs are limited to several branches. Changes in consensus nucleotides proceed via an intermediate stage when the respective position is not conserved.

KEYWORDS: Evolutionary scenario, regulatory signal, frequency matrix, evolution along a tree, transcription factor tree

INTRODUCTION

Reconstruction of protein (gene) evolutionary trees, species trees, reconciliation of gene and species trees are traditional problems of the molecular evolution theory and bioinformatics. A somewhat different problem, addressed here, is to reconstruct evolution of transcription factor (TF)-binding DNA motifs along a given TF tree. Preliminary observations have demonstrated that the evolution of transcription factors is accompanied by the evolution of their binding DNA motifs [1], and this may be used to deduce the amino acid residues responsible for the specific recognition of the DNA binding sites [2]. On the other hand, specific positions in orthologous sites may be conserved in different species at considerable evolutionary distances [3]. The functional conservation of a motif position within one species is correlated with the number of protein-DNA contacts at this position [4], and the binding energy at this position [5].

We start with a tree G that describes the evolution of a transcription factor family, and, given sets of binding sites for the extant members of the family, aim to reconstruct the binding motifs for the ancestral TFs. This problem involves large datasets of binding sites for many members of a TF family, and has

[*] Corresponding author. E-mail: gorbunov@iitp.ru.

Electronic publication can be found in *In Silico Biol.* **10**, 0012 <http://www.bioinfo.de/isb/2010/10/0012/>, 9 March 2010.

become relevant after sequencing of hundreds of bacterial genomes and development of the appropriate comparative genomic techniques for identification of binding sites (reviewed in [6]).

The basic underlying assumption is that for each motif position $i = 1, \ldots, n$ (where n is the motif length which may be assumed to be the same for all considered TFs for a given G) there is a limited set of branches $S(i)$ of the tree G, such that the positional nucleotide frequencies at position i change strongly along branches from $S(i)$ and only weakly along the remaining branches. Intuitively, this assumption reflects the principle of the maximum parsimony [7]. It implies that positional nucleotide frequencies of binding motifs evolve independently for different positions, which is a reasonable first step approximation [8,9]. Separately, we consider the case of concordant evolution of respective positions in palindromic motifs.

The *optimal* i *-scenario* is defined as a pair of a set of S_i (called the *support* of the optimal i-scenario) and an *assignment* f_i of nucleotide frequencies at each node of the tree G, for a fixed position i. The *final optimal scenario* consists of the branches that belong to many i-scenarios (called the *support* of the final scenario) and the *assignment* h of frequencies matrices that are combined from the *assignments* f_i of the optimal i-scenarios for all i. Normally h does not need to be specified, as it can be uniquely reconstructed given the set $\{f_i|\ 1 \leqslant i \leqslant n\}$. The formal definitions and the brief description of the algorithm are given in the next section, whereas technical details can be found in [10]. In the sections Results and Discussion we describe and discuss testing of the algorithm on simulated data generated by modeling of the motif evolution along several types of artificial TF trees. Then we apply the developed technique to several families of transcription factors with a sufficient number of candidate sites identified in comparative genomic studies.

BASIC DEFINITIONS AND SKETCH OF THE ALGORITHM

Here we briefly formulate the main definitions and then describe the improvements beyond [10] that were applied to yield the presented results. Since we need to define a positional frequency matrix at each terminal vertex of the TF tree G, and the generation of such matrix requires a sufficient number of sites, we collapsed short terminal branches in the initial tree G_0 to produce terminal vertices of a new tree G. Hereinafter we use the notation G for both TF trees (for some TF families we distinguish between G_0 and G).

For each branch u of the given TF tree G denote by u_0 its start node and by u_1 its end node. Fix a motif position i. Assignment f_i is a function (with i as the argument) from the set of all nodes to the set of 4-vectors describing nucleotide frequencies (i.e. distributions of nucleotides), extant and ancestral, at position i. For simplicity, we may drop the argument i while discussing a single position. For a terminal node of G (the extant subfamily of TFs) these vectors are given by the input data (binding sites for a fixed TF subfamily). This collection of input data is denoted by θ. We aim to minimize the function F (depending on two non-numerical arguments f and S) that measures the changes in nucleotide frequencies along the branches *not belonging to the support* S:

$$F(f, S, \theta) = \sum_{u \notin S} \rho(f(u_0), f(u_1)) \tag{1}$$

where $\rho(a, b)$ is a distance measure between any two 4-vectors a and b (distributions a and b), satisfying two restrictions: the components of the vectors are non-negative, and their sum is one. Note that the support S is selected simultaneously with minimizing F with respect to f so that, firstly, the size of S

be lower, and, secondly, from the start u_0 to the end u_1 of each edge u from S, f changed considerably (that is, $f(u_0)$ and $f(u_1)$ were significantly different). The strength of this difference is measured by ρ $(f(u_0), f(u_1))$. On the other hand, on each vertex u, not belonging to S, the change should be low. The idea is simple: S should contain only those edges, on which the evolutionary change of f is higher than a fixed threshold: this is the definition of f.

The distance measure was defined by

$$\rho(a,b) = \sum_{I=1}^{4} (\sqrt{a_I} - \sqrt{b_I})^2.$$

The algorithm does not depend on a choice of the distance measure. Note that not only f, but θ and S as well depend on a fixed i, so we deal with a triple $\langle f_i, \theta, S_i \rangle$. As mentioned, the subscript i may be omitted.

A pair $\langle f_i, S_i \rangle$ is called an *i-scenario*. The size $|S|$ of an i-scenario $\langle f, S \rangle$ is the number of branches in its support S. The *main penalty* of an i-scenario $\langle f, S \rangle$ is defined by the formula

$$\tilde{F}(f, S) = \sum_{u \notin S} \delta(f(u_0), f(u_1))$$

that uses a different distance

$$\delta(a,b) = \sum_{I=1}^{4} \sqrt{|a_I - b_I|}.$$

The *auxiliary penalty* is

$$\overline{F}(f, S) = \max_{u \notin S} \rho(f(u_0), f(u_1)).$$

For a good i-scenario all these four values should be small:

$$F(f, S), \tilde{F}(f, S), \overline{F}(f, S), |S| \to \min \tag{2}$$

The pair $\langle f, S \rangle$ rang providing the minimum of Eq. (2) is called *optimal i-scenario*. More exactly, the minimum in Eq. (2) is defined and calculated by the algorithm from [10]. The basic idea of this algorithm is as follows. The size g of the set S increases, starting at 0 with step 1 until some fixed maximal value g_{max} is reached. At each g a greedy algorithm is used to generate a sufficiently large number of scenarios with the minimal possible value of F.

In a nutshell, the set of scenarios, among which the optimal i-scenario is selected (that is, the set, on which the function F is minimized) is constructed by induction. For a current set of scenarios X, each of power $g-1$, the next set of scenarios, each of power g, is obtained by extending each scenario from X by exactly one edge. Several such edges are considered, that yield new, different scenarios with lower penalty.

Among those, one optimal i-scenario for given g is selected as the one that provides the minimum of $p \cdot \tilde{F} + \overline{F}$, where p is a *parameter* reflecting the importance of the main penalty respective to the auxiliary penalty (in the examples below we assume $p = 10$). The algorithm uses the value g, maximizing

$$\frac{2 \cdot \tilde{F}(g-1)}{\tilde{F}(g) \cdot \left(\frac{\tilde{F}(g-2)}{\tilde{F}(g-1)} + \frac{\tilde{F}(g)}{\tilde{F}(g+1)} \right)} + \frac{1}{p} \cdot \frac{2 \cdot \overline{F}(g-1)}{\overline{F}(g) \cdot \left(\frac{\overline{F}(g-2)}{\overline{F}(g-1)} + \frac{\overline{F}(g)}{\overline{F}(g+1)} \right)} \tag{3}$$

where $F(g)$ (with superscripts) are the values of the corresponding F on the optimal i-scenario for a given g.

The sense of this formula is that the algorithm selects the size g, for which the main and auxiliary penalties sharply decrease, assuming that the change has been gradual below and above this value of g. The *optimal i-scenario* is defined as the optimal i-scenario for this g. This value of g is called *crucial*.

For the examples discussed below we varied g up to $g_{max} = 11$. The testing demonstrated that at further increase of g_{max}, the crucial value g does not change (data not shown). Thus, at $g_{max} = 11$, the algorithm may select some value g from 2 through 10, and the optimal i-scenario corresponds to this crucial g.

The *significance* of a branch u in an i-scenario $\langle f_i, S_i\rangle$ is defined as $\delta(f(u_0), f(u_1))$. The significance is higher at branches along which considerable changes in nucleotide frequencies have occurred. The *weight* of a branch is defined as the number of optimal i-scenarios ($i = 1, \ldots, n$) whose support includes this branch. The *final scenario* includes branches whose weights exceed some threshold, that is, the branches along which significant changes of nucleotide frequencies have occurred at many positions.

The following essential improvements have been introduced compared to the algorithm [10].

1. *Significance* of a branch takes into account the number of sites in the corresponding subtree (that is the tree coming from the end of this branch). In particular, the branch leading to leaves with a small number of sites has little chance to become significant. This is implemented by changing the function F. Now each term corresponding to a branch is multiplied by $1-\frac{\sqrt[4]{l}}{\sqrt{m}}$, where l is the total number of sites in leaves of the subtree corresponding to this branch, m is the total number of leaves; see also Test 3 in the next section.
2. The termination criterion at increasing the support size g is set by Eq. (3).
3. In [10] the lengths of all sites were assumed to be equal. Here we allow for unequal site length.

RESULTS

Testing on artificial samples

Four different tests were performed to assess the performance of the algorithm. In the first, second and third tests a single position was modeled, hence the optimal i-scenario was the final scenario at the same time, whereas in the fourth test the final scenario was the result of the all optimal i-scenarios.

Test 1. Here G is a balanced binary tree with 64 leaves, so that each path from the root to a leaf contains exactly six branches, Fig. 1a. The input distribution of nucleotide frequencies at leaves is given in Fig. 1a. The algorithm outputs an optimal scenario with ten branches and the assignment shown in Fig. 1b.

The branches obtained for increasing g (from 0 to 11), once included, rarely leave the optimal scenario. Notably the main and auxiliary penalties even reach the global minimum at zero value.

Test 2. Here G is a comb-like, unbalanced binary tree. The tree, nucleotide frequencies at leaves, the support of the optimal scenario and the optimal assignment are shown in Fig. 2. Note that the nucleotide distribution at the root is almost uniform, despite being strongly uneven at most leaves.

Test 3. Here we analyzed the noise tolerance of the model by perturbing the nucleotide frequencies and considering uneven site numbers at the leaves of the tree G. Firstly, we added noise to nucleotide frequencies on leaves. The noise was modeled via changing the frequency by adding a value uniformly distributed in the interval $[-d, d]$, where d was the noise level. At $d = 0.05$, the same optimal scenario

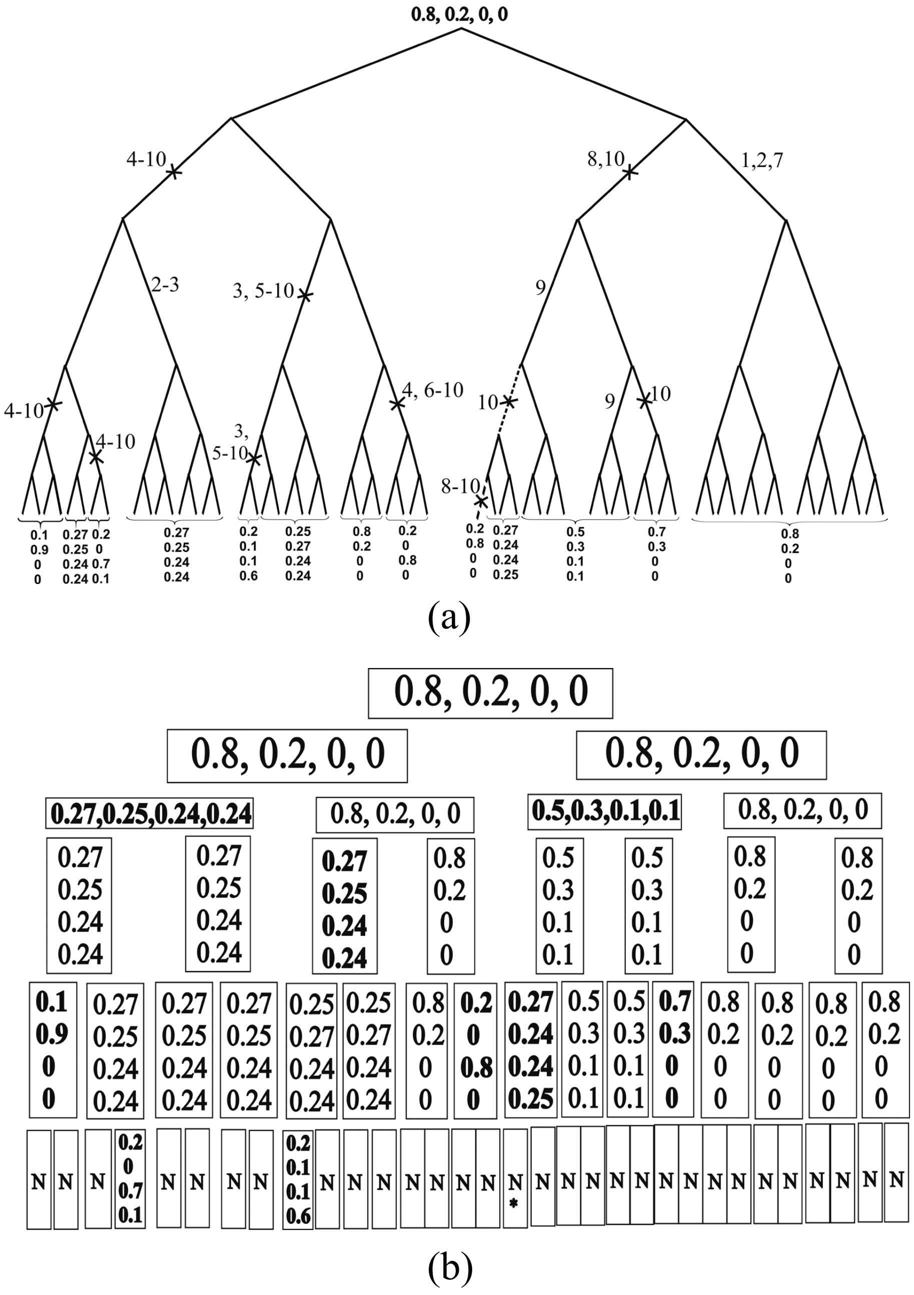


Fig. 1. (a) Balanced artificial tree. The nucleotide frequencies for a single position are given for terminal nodes. The branches forming the support of the optimal scenario are shown by crosses. From the optimal assignment, only the distribution at the root is shown; all distributions are given for four nucleotides in the alphabetic order A, C, G, T. The complete optimal assignment is shown in Fig. 1b. A number assigned to a branch is the value of the parameter g, at which the branch was included in the optimal scenario. The crucial value of the algorithm is $g = 10$. Two branches shown by dotted lines are used separately in Test 3 below. (b) The optimal assignment corresponding to the support in Fig. 1a is shown. "N" means the same distribution as above in the figure. In the bottom row, which is not shown, only one of the two vertices, followed the vertex marked by "*", has a different distribution, (0.2,0.8,0,0), compared to the parent node. Bold: distributions at termini of branches belonging to the optimal support shown in Fig. 1a.

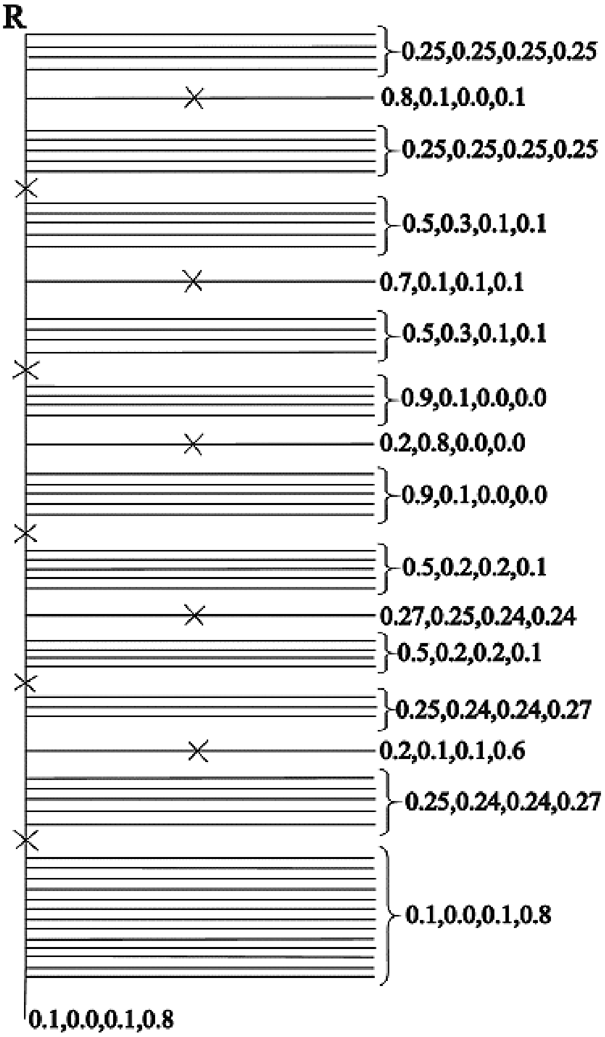


Fig. 2. Unbalanced artificial tree. "R" denotes the root. The branches forming the optimal scenario are shown by crosses. The values of the optimal assignment at each internal node coincide with the given values of the assignment at the leaf that joins this node by a route not containing crosses. Other notation as in Fig. 1a.

was obtained at the crucial value $g = 10$, whereas the nucleotide frequencies at internal nodes fluctuated slightly. The optimal scenario was the best relative to the main penalty in all cases, and was the best relative to the auxiliary penalty in 80% cases; in the remaining 20% cases the top position was occupied by another ("alternative") scenario. More details about the dependence of the results on the noise level are given in Table 1.

Table 1 demonstrates that the result quality is worse for the balanced tree compared to the unbalanced one at the same noise level. This may be explained by the more complex topology of the former that creates possibilities for competing scenarios. At low noise level, the alternative scenario was worse but close to the optimal one and it had overcome the latter when the noise increased and then the algorithm outputted just the alternative one.

Further, we modeled uneven distribution of site numbers at the tree leaves. To do that, we used an extended version of the algorithm that additionally takes into account the number of sites at the leaves.

Table 1
Dependence of the result quality on the noise level

1	Noise level	0.05	0.08	0.1	0.15	0.2	0.25	0.3
2	Result quality	1	0.7	0.5	0.4	0.3	0.26	0.2
3	Result quality	1	1	0.9	0.6	0.5	0.3	0.25

The result quality is defined as the ratio of the number of branches in the intersection of the supports of the optimal scenarios at the zero and given noise d levels to the number of branches in the union of these supports. "Union" means branches belonging to at least one of these scenarios, "intersection" – branches belonging to both scenarios. Lines: (1) noise level d; (2) result quality for the data from Fig. 1a with the balanced tree; (3) result quality for the date from Fig. 2 with the unbalanced (comb-like) tree.

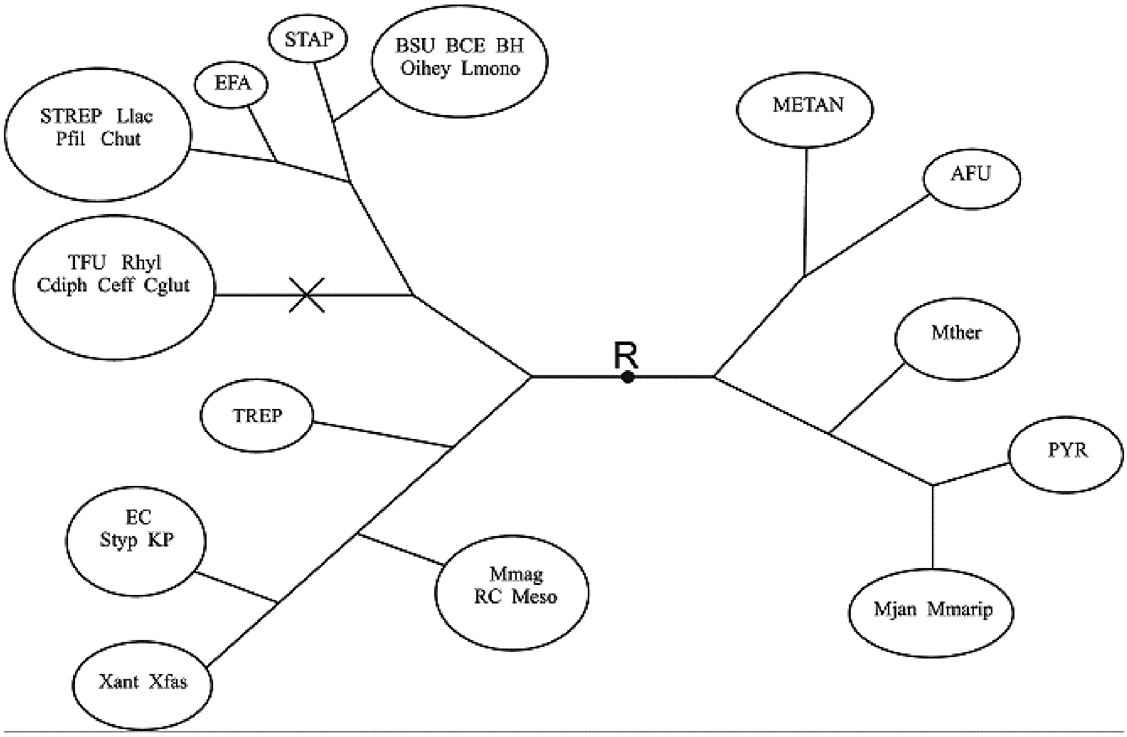


Fig. 3. The species tree G for the genomes containing the MntR transcription factors. Species names are abbreviated in ovals, see Appendix, Table 1. The branch forming the support of the final scenario is shown by the cross.

We describe one example in detail. The number of sites was set as follows: the single leaf corresponding to one selected branch (shown as dotted line in Fig. 1a), two or three sites were assumed; for the three other leaves corresponding to the other selected branch (also shown as dotted line in Fig. 1a), four or five sites were assumed; for the remaining leaves, ten through thirty sites were assumed. The site numbers at leaves were selected randomly many times, satisfying the above constraints. At zero noise, in all cases the relatively lower reliability of branches with a small number of sites was manifested only in the relatively lower significance of these branches. However, already at a relatively low noise $d = 0.05$, the optimal scenario was obtained exactly only in 10% of cases, in 60% cases it was losing the branches with a small number of sites, whereas in 30% cases the alternative scenario was output, again, without these two branches.

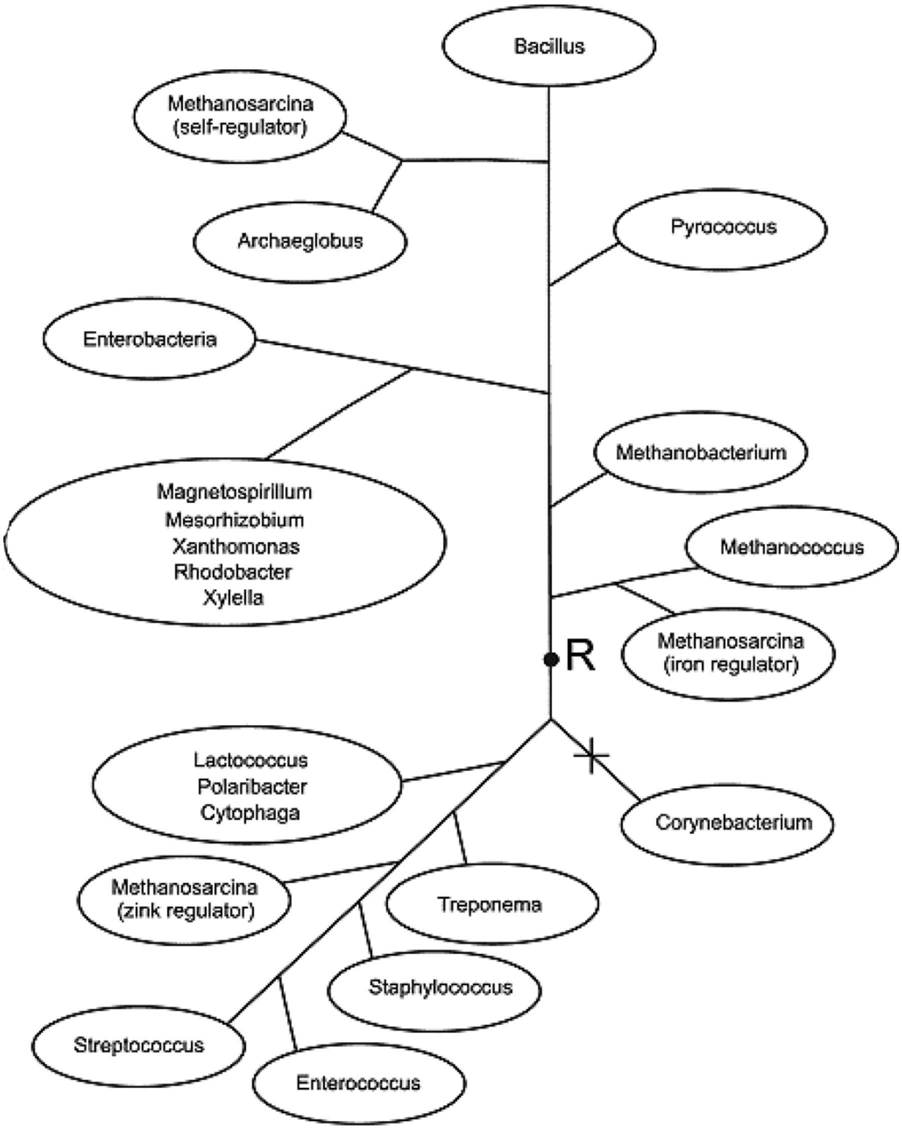


Fig. 4. The gene tree G of the MntR family. The input tree G, is constructed by a TF tree G_0 by merging close several leaves of G_0 in one cluster (clusters are shown by ovals in the Figure). Each such cluster serves as a single terminal node of the tree G. The branch forming the support of the final scenario is shown by cross.

Test 4. Finally, we tested the robustness of the algorithm on natural data with additional random noise. Many positions and hence the final scenario were considered here. As above, nucleotide frequencies at leaves were modified by addition of a value uniformly distributed in the interval $[-d, d]$. We observed a common behavior illustrated here for the LacI family, (see below). At increasing noise level d, the final scenario was losing branches, starting with branches having the lowest weight and significance. For the LacI family without noise, the optimal scenario contained three branches, see below. At $d = 0.2$, the first branch (the one coming from node 1) was not included into the final scenario; at $d = 0.3$, the second branch (coming from node 3) was lost, and, finally, at a very high noise level $d = 0.7$, the final scenario lost the last remaining branch, and become empty.

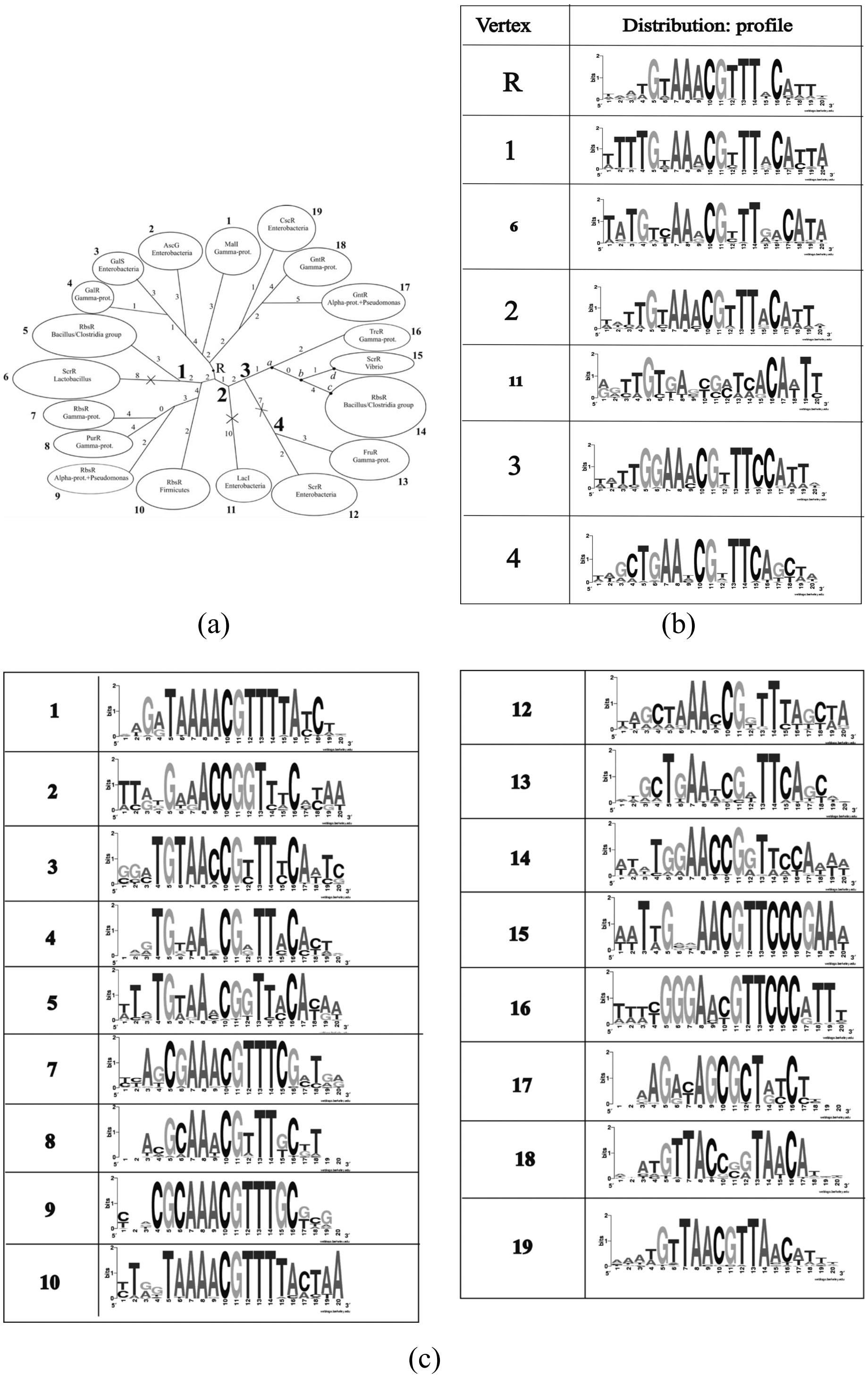


Fig. 5. The input tree G for the LacI family. To each leaf of tree G representing a pair (protein subfamily, taxon), circled by an oval, a number is assigned. Other notation as in Fig. 4. Figures 5b and 5c provide the initial data, that is, the alignment profiles of the respective site alignments, and the profiles at the termini of branches belonging to the final scenario. For each branch the number of positions i that optimal i-scenario contains this branch is shown (in order to avoid a confusion this numbers are small in contrast to large and bold vertex numbers). The nodes marked by italic a, b, c, d are related to the discussion in the text and Table 2. Distributions at the termini of branches from the final scenario in Fig. 5a. (c) Profiles reflecting the initial data at leaves that are not present in Fig. 5b.

Table 2
Nucleotide frequencies in some ancestor nodes of LacI tree for the optimal (9,12)-scenario

	$a, 9$	$a, 12$	$b, 9$	$b, 12$	$c, 9$	$c, 12$	$d, 9$	$d, 12$
A	0.89	0.01	0.68	0.02	0.00	0.24	1.00	0.00
T	0.01	0.92	0.01	0.73	0.03	0.00	0.00	1.00
G	0.00	0.07	0.00	0.25	0.00	0.76	0.00	0.00
C	0.10	0.00	0.31	0.00	0.97	0.00	0.00	0.00

Palindromic function $\tilde{F}$ was used. In the column headers, a letter denotes the ancestor node shown in Fig. 5a and an integer denotes position: 9 or 12 in LacI signal.

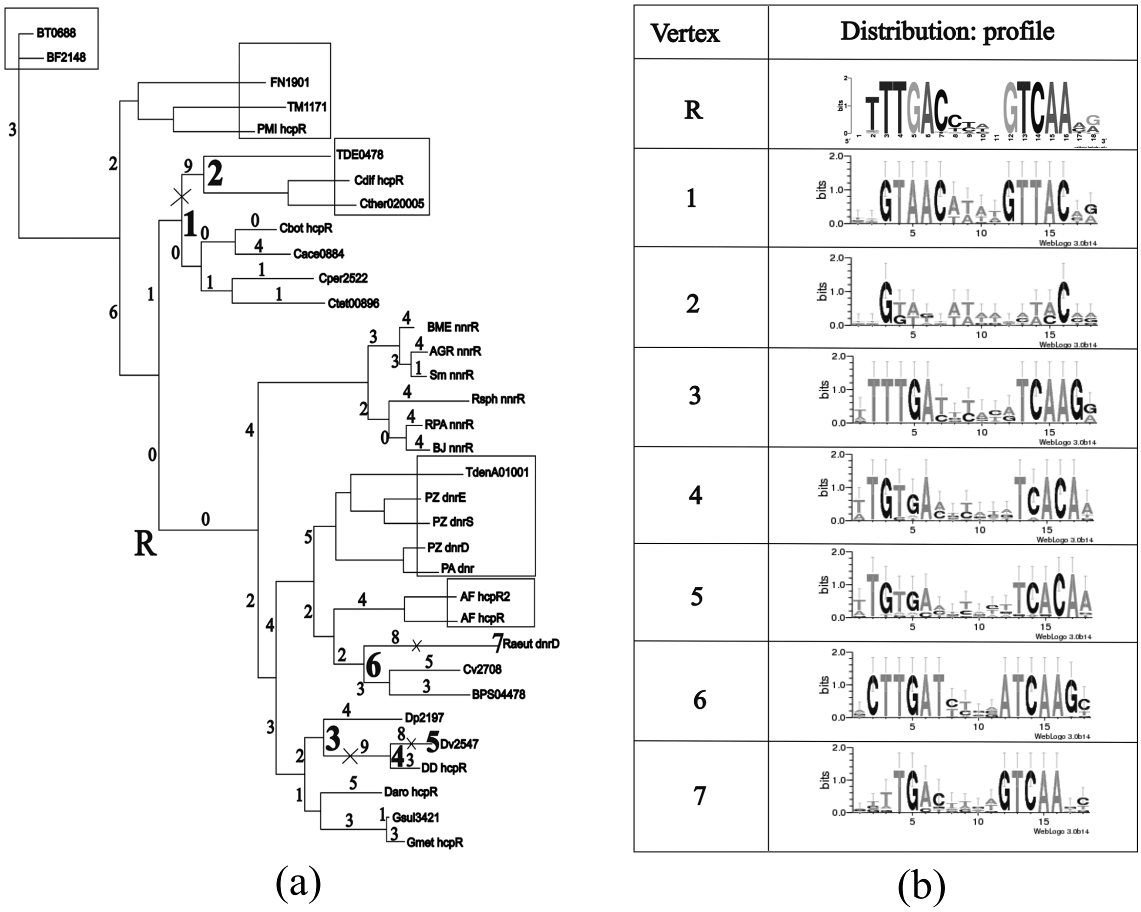


Fig. 6. (a) The gene tree for nitrosative stress TFs from the FNR/CRP family. (b) Profiles from the final scenario are shown at internal nodes of the tree in Fig. 6a.

Application to transcription factor families

After validation of the approach on simulated data, we applied the algorithm to several transcription factor (TF) families where a large number of binding sites had been previously identified by comparative genomic methods.

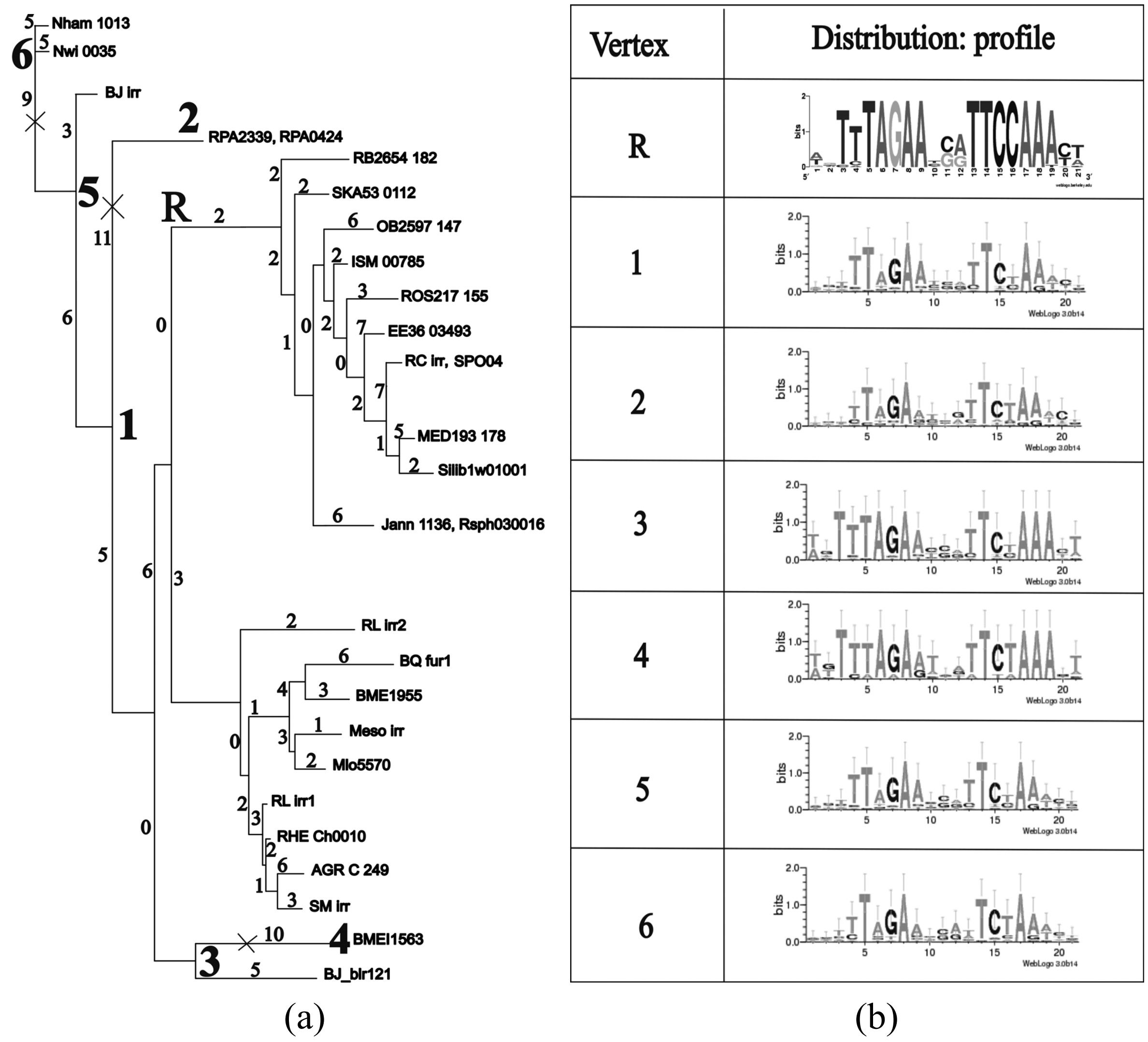


Fig. 7. (a) The gene tree of the Irr subfamily. (b) Profiles from the final scenario are shown at branches of the tree in Fig. 7a.

The **MntR** family: Two trees were considered here as the tree G: the species tree reflecting the bacterial taxonomy, Fig. 3, and the phylogenetic tree of the TF family, Fig. 4. In both cases the optimal scenario contains one branch leading to the *Corynebacteria* in the latter case and to the taxon $\{$*Thermobifida fusca*, *Rubrobacter xylanophilus*, *Corynebacterium diphtheriae*, *Corynebacterium efficiens*, *Corynebacterium glutamicum*$\}$ in the former case. The weight of this branch is more than half of all positions.

Both trees produce the same final scenario, with most changes in the *Corynebacterium* spp.

The **NrdR** family experienced changes in the *Thermus/Deinococcus* group [10].

The **LacI** family: Most TFs from this family regulate sugar catabolism genes. The motif length is 20 nucleotides. The input tree G is shown in Fig. 5a [11,12]. The input data on the leaves and the final distributions at some nodes are shown in Fig. 5b-c. In Fig. 5a each branch is assigned the number of optimal i-scenarios containing the branch (in lowest case). The changes occurred mainly in the ScrR, LacI, FruR branches in the *Lactobacillus*, *Enterobacteria*, gamma-Proteobacteria groups.

Table 3
The number of sites at leaves of the trees G for five families of TFs

a) Nitrosative stress regulators from the FNR/CRP family					
BT0688; BF2148	3	AGR nnrR	6	CV2708	6
FN1901 – PMI hcpR	4	Sm nnrR	8	BP S04478	3
TDE0478 – Cther020005	5	Rsph nnrR	6	DP 2197	4
Cbot hcpR	4	RPA nnrR	6	DV2547	10
Cace0884	6	BJ nnrR	9	DD hcpR	10
Cper2522	4	TdenA01001 – PAdnr	27	Daro hcpR	4
Ctet00896	7	AF hcpR2; AF hcpR	4	Gsul3421	8
BME nnrR	5	Raeut dnrD	7	Gmet hcpR	9
b) The Irr subfamily of the FUR family from the alpha-Proteobacteria					
Nham 1013	18	EE36 03493	5	Mlo5570	9
Nwi 0035	24	RC irr; SPO04	6	RL irr1	9
BJ irr	26	MED193 178	3	RHE CH0010	7
RPA2339; RPA0424	26	Silib 1w01001	3	ARG C 249	14
RB2654 182	3	Jann 1136; Rsph030016	5	SM irr	11
SKA53 0112	5	RL irr2	9	BMEI1563	7
OB2597 147	5	BQ furr1	3	BJ_blr121	26
ISM 00785	4	BME1955	7		
ROS217 155	4	Meso irr	5		
c) The iron response regulators FUR from the delta-Proteobacteria					
DD 394232; DD 395878	25	Gsul 381665	4	Dace 392427; Dace 391943	13
DV 206374	17	Gmet 379927	6		
d) The iron response regulators FUR from the alpha- and gamma-Proteobacteria, the Firmicutes, and the manganese regulator MUR from the alpha-Proteobacteria					
Meso030031 – SM mur	6	AGR C 620 – RL mur	7	Saro020011	7
GOX0771	7	PB2503 048	9	ELI1325	16
PU1002 fur – ISM 15430	17	OA2633 102	14	BSU02348	50
Rrub020011	17	CC0057	10	ECO4589	64
MM amb1009; MM amb4460	17	ZM01412	14		
RPA0450 – Nwi 0013	8	Sala 1452	10		
e) The Rrf2 family					
DR-1; Dgeo-1	4	Raeut-3; Rmet-3	7	Silib-4	4
SO-2; IDL-2	8	Acin-3 – Cviol-3	12	SPO-4	4
ECA-2 - YE-2	19	SCO-3 - TFU-3	3	BQ-4	4
Ppro-2 - Vvul-2	10	GOX-3 - BSt-3	11	Meso-4	5
MS-2 - HI-2	6	Bcepa-3 - Mdeg-3	30	BME-4	15
HD-2; Aple-2	5	SO-3; SPU-3	6	RL-4	24
Mdeg-2 - PP-2	10	ECA-3; ECH-3	9	AGR-4	24
Nmen-2 - BPS-2	23	Styp-3	4	Smel-4	22
RC-2 - SPO-2	4	KP-3	4	AGR-1; Smel-1	3
Rcon-2; Rsib-2	3	EC-3	4	BME-1; Mlo-1	7
Gsul-2; Gmet-2	5	YP-3; YE-3	8	Gmet-1; Gsul-1	4
Dace1-2; Dace2-2	3	Vpar-3 - Vvul-3	16	Rrub-1 - RL-1	11
Cdif-2 - TTE-2	9	RC-4	5	MSMEG-1 - Mmic-1	6
Oihey-2 - SA-2	5	Rsph-4	4		

Nitrosative stress regulators from the FNR/CRP family: The TF tree G is shown in Fig. 6a. The motif length is 18 positions [1]. The final scenario contains four branches: a branch leading to the node with three genomes $\{$*Clostridium difficile*, *Clostridium thermocellum*, *Treponema denticola*$\}$, two consecutive branches leading to *Desulfovibrio vulgaris*, and a branche leading to *Ralstonia* spp. The motifs corresponding to starts and ends of these branches are shown in Fig. 6b. As expected from the palindromic structure of the motif, positions that have the best i-scenarios form symmetric pairs 6–13, 5–14, 4–15, and 3–16. The results do not change if only the helix-turn-helix DNA-binding domain is used to construct the TF tree; only the last branch is not included in the optimal scenario (data not shown). The numbers of sites at the leaves for this and subsequent families are shown in Table 3.

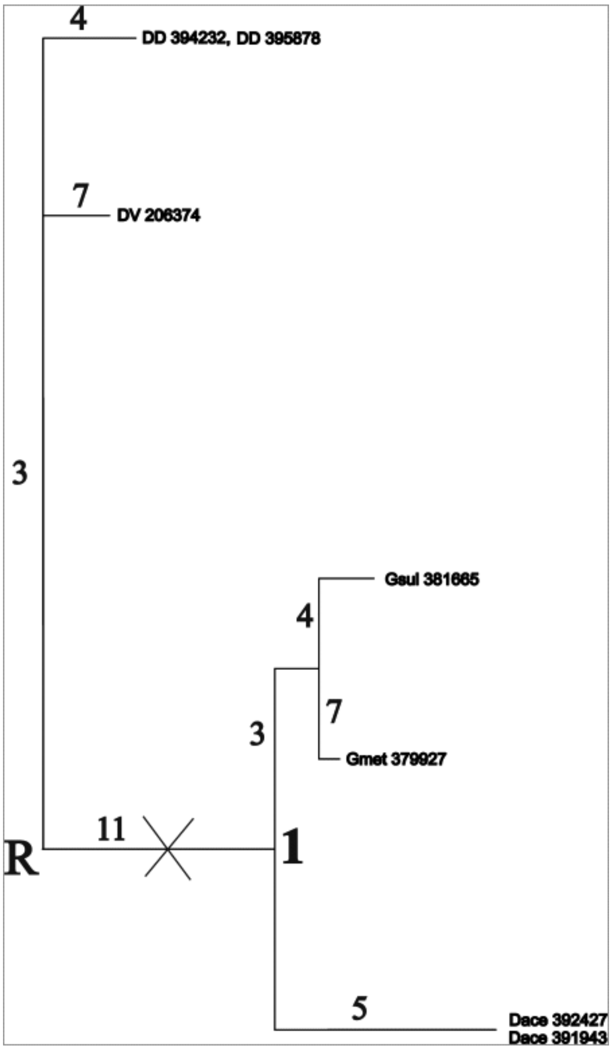


Fig. 8a. The gene tree of the FUR TFs from the delta-Proteobacteria.

The Irr subfamily of the FUR family from the alpha-Proteobacteria: The TF tree is shown in Fig. 7a; the motif contains 21 position [13]. The final scenario contains three branches: a branch leading to *Rhodopseudomonas palustris*, a branch leading to *Brucella melitensis*, and a branch leading to the last common ancestor of *Nitrobacter winogradskyi* and *Nitrobacter hamburgensis*. The changes in the motifs corresponding to these branches are shown in Fig. 7b. Again, the palindromic motif yields pairs of symmetric positions with optimal i-scenarios: 4–18, 5–17, 7–15, 8–14, 9–13.

The iron response regulators FUR from the delta-Proteobacteria: The TF tree is shown in Fig. 8a; the motif length is 17 nucleotides [14]. This case is somewhat more complicated, since for many positions i there exist two optimal i-scenarios with equal main and auxiliary penalties. This is caused by the fact that many optimal i-scenarios contain a branch coming from the root. In these case two optimal i-scenarios containing each just one of two branches connected to the root may arise such that they have the same penalties. Fig. 8b features profiles corresponding to the final scenario found in this example. This scenario contains one branch coming from the root and marked in Fig. 8a.

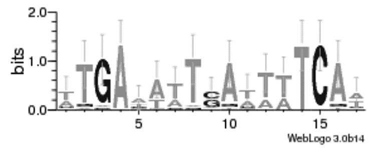
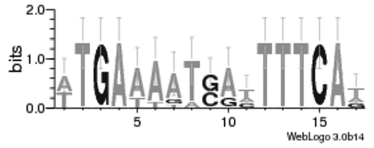
Vertex	Distribution: profile
R	
1	

Fig. 8b. Profiles from the final scenario are shown at internal nodes of the tree in Fig. 8a.

The iron response regulators FUR from the alpha- and gamma-Proteobacteria, the Firmicutes, and the manganese regulator MUR from the alpha-Proteobacteria: The tree is shown in Fig. 9a; the motif contains nineteen positions [13]. The final scenario contains four branches of approximately equal weights: a branch leading to FUR TFs from gamma-Proteobacteria, a branch leading to *Zymomonas mobilis*, and two consecutive branches leading to *Rhodospirillum rubrum*. The respective profiles are shown in Fig. 9b.

The NikR family: The algorithm has generated the empty support and the null assignment (the tree is not shown).

The Rrf2 family: The HTH-region is well defined: according to the PFAM database it occupies positions 29–85; or according to InterPro positions 3-135 (in *Ervinia carotovora*). The latter fragment was used to construct the tree shown in Fig. 10a. The motif includes nineteen positions [13]. The final scenario contains only one branch. This branch leads to an internal node corresponding to the last common ancestor of the IscR TFs from the *Firmicutes* (*Bacilli* and *Clostridia*). The respective profiles are shown in Fig. 10b.

DISCUSSION

The case of the LacI family illustrates several interesting features of the model

1. The crucial criterion should be applied independently for each position i. Indeed, if one fixes a universal value for all i (e.g. $g = 5$), the final scenario for the LacI family would include branches 11, 6, and 12–13 (that is, the branches leading to the last common ancestor of leaves 12 and 13) with weights, respectively, 10, 8 and 7. But the visual analysis shows that while this is an adequate solution as regards branches 11 and 12–13, since the respective profiles are radically different from the neighboring profiles, but the profile at the end of branch 6 does not seem different from the neighboring profiles. Indeed, the high weight of this branch is caused by the fact that this branch is included in a large number of high-penalty optimal i-scenarios.
2. The algorithm automatically yields a symmetric solution if the given sites are palindromic. We can see that on the following example. The low-penalty i-scenarios were obtained for positions 5, 7, 8,

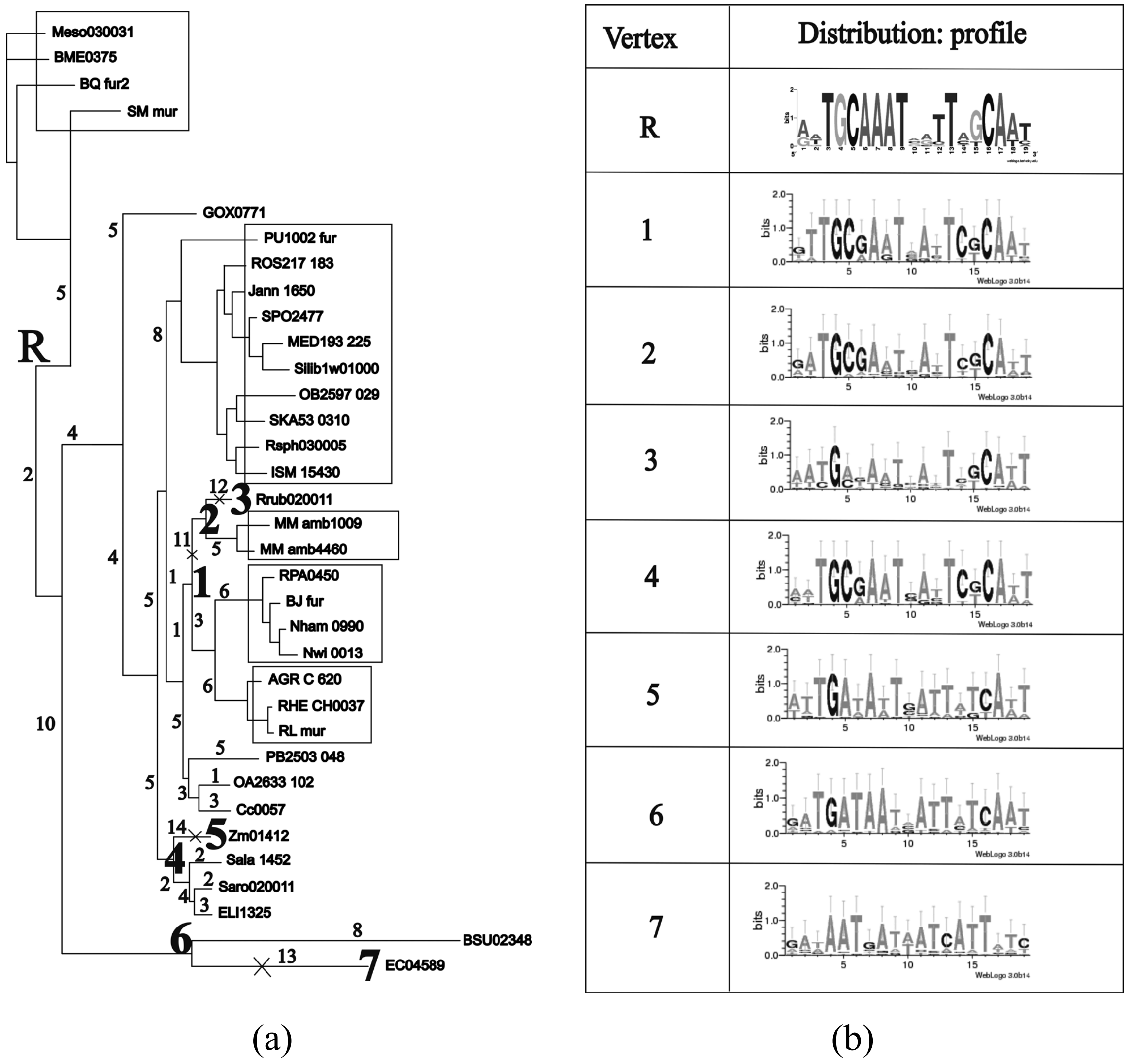


Fig. 9. (a) The gene tree of the FUR TFs from the alpha-Proteobacteria, gamma-Proteobacteria, *Firmicutes*, and the MUR TFs from alpha-Proteobacteria. (b) Profiles from the final scenario are shown at internal nodes of the tree in Fig. 9a.

10, 11, 13, 14, 16; and a large number of common branches was calculated for symmetric pairs of positions 5 and 16, 7 and 14), 8 and 13), 10 and 11. To see, consider a palindromic penalty function

$$\breve{F}(f, g, S) = F(f, S) + F(g, S) + 2 \cdot \sum_{v} \rho(f(v), \overline{g}(v)) \tag{4}$$

3. Now the algorithm constructs the optimal scenario for a pair of complementary positions (i, j), where f and g are the assignments corresponding to positions i and j, respectively; v enumerates all internal nodes of tree G, and the line above g denotes permutation of frequencies in g according to nucleotides complementarity. Table 4 shows the final scenario constructed using $\breve{F}$. It turned out that scenarios corresponding to position pairs (5,16), (7,14), (8,13), (10,11) and only for them have low main and auxiliary penalties. The same result for other examples. It shows that function

Table 4

Optimal i-scenarios corresponding to function F for each position i, and optimal (i, j)-scenarios corresponding to function for each symmetrical pair (i, j)

Position i	1	2	3	4	5	6	7	8	9	10
Crucial value g	5	5	5	5	5	5	5	4	5	5
Optimal i-scenario	2–4; **11**; 2; 3; 7–10	2; 3; 5–10; 6; 7–9	5–6; 14; 11–16; **12–13**; 5	6; 7–10; 9; 8; **12–13**	1; 6; 10; 7; **12–13**	1–4; 7; 17; 12 12–16	17–19; 17; **11**; 16; 15	18; 4; **11**; 13	17–18; 17; 2–4; **11**; **12–13**	18; **11**; 13; 16; 14
Main penalty of the optimal i-scenario	21.8	22.2	24.7	25.8	8.4	27.3	6.0	0.0	21.1	3.6
Position i	11	12	13	14	15	16	17	18	19	20
Crucial value g	5	5	4	5	5	5	5	5	5	5
Optimal i-scenario	19; 18; 5–6; **11**; 13	17–18; 5; 2–4; 18; **11**	8; **11**; 12; 14	17–19; 17; **11**; 14–16; 14	7; 7–10; 17; 12; 12–16;	1; 6; 10; 7; **12–13**	1; 6; 9; 8; **12–13**	1–4; 3; 6; 9; **12–13**	2; 5; 7–10; 7–9; 8	2–4; 3–4; 5–10; 7–9; **11**
Main penalty of the optimal i-scenario	1.8	22.8	0.5	4.5	26.3	7.7	26.0	21.7	23.8	24.6
Position (i, j)	(1,20)	(2,19)	(3,18)	(4,17)	(5,16)	(6,15)	(7,14)	(8,13)	(9,12)	(10,11)
Crucial value g	5	5	5	5	5	5	5	4	5	5
Optimal (i, j)-scenario	4; 3; 9; 10; **11**	2; 3; 6; 7–9; 5–10	3; 6; **11**; 16; 14–15	1; 9; 8; 12–13; 15	**1; 6; 10; 7; 12–13**	17; 7–10; 12–16; 7; 12	**17–19**; **17**; **11**; 16; 15	8; **11**; 12; 13	17; 18; 2–4; 5; **11**	**18**; **11**; **13**; 16; 14
Main penalty of the optimal (i, j)-scenario	20.7	21.7	21.0	22.0	6.6	23.0	5.2	0.2	19.0	2.3

Lines 1 and 5: the position number. Lines 2 and 6: the value g, at which the algorithm has terminated. Line 3: optimal i-scenario; boldface: branches included in the final scenario. For each optimal i-scenario, the most significant branch is underlined (branch 11 has the highest significance for $i = 1$, branch 6 has the highest significance for $i = 2$, etc.). Branch 6 has low significance, but belongs to the best i-scenario for many positions i. Line 4: the main penalty of the optimal i-scenario. Line 9: all pairs of palindromic positions. Line 10-12 are analogous to the previous ones, but for the function that operates with pairs of symmetrical positions. Branch 11 that belongs to most optimal i-scenarios is set in bold and underlined. Other branches set in bold belong to four optimal (i, j)-scenarios with the lowest main penalty (for other pairs (i, j), the optimal (i, j)-scenarios have much higher penalties). These branches also belong to optimal i-scenarios and j-scenarios that also have the lowest main penalties compared to all other k-scenarios.

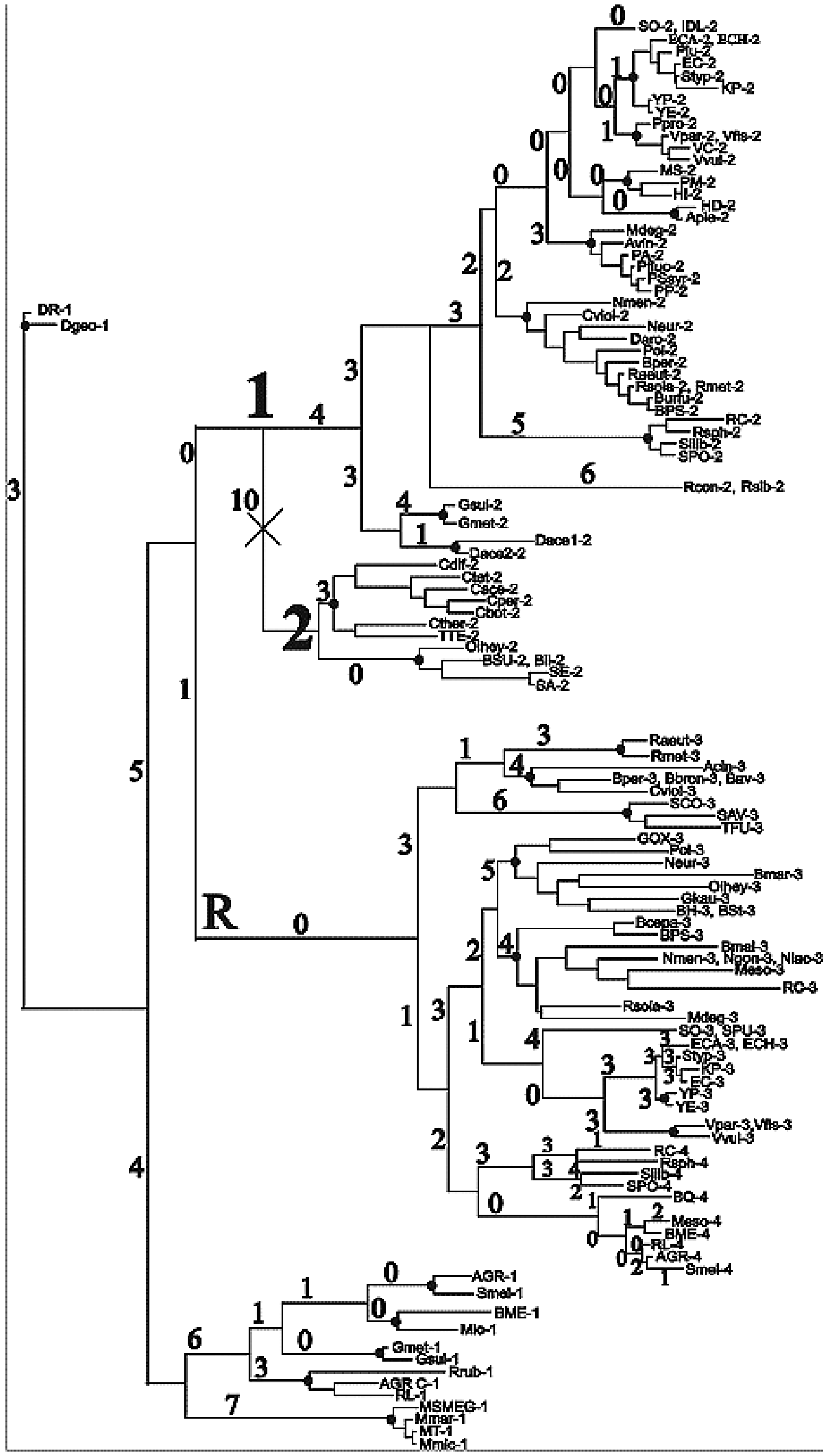


Fig. 10a. (a) The gene tree G_0 of the Rrf2 family.

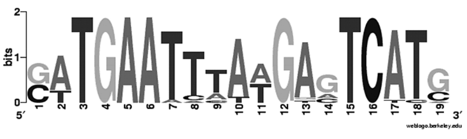
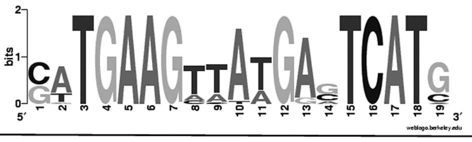
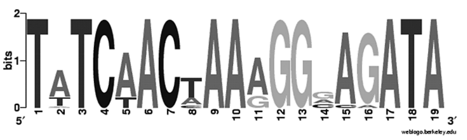
Vertex	Distribution: profile
R	
1	
2	

Fig. 10b. Profiles from the final scenario are shown at internal nodes of the tree in Fig. 10a.

F pays attention the structure of sites; and there is tight correlation F and $\breve{F}$. Each branch from the optimal $(i,\, j)$-scenarios belongs to the optimal i-scenario or the optimal j-scenario, or usually both (as in Table 4), if the standard function F from (1) is used. Thus the optimal $(i,\, j)$-scenario from Eq. (4) agrees with the optimal i- and optimal j-scenarios from Eq. (1).

4. The change of one consensus nucleotide to another one may include an intermediate stage of loss of the conservativity. For example, consider a pair of positions (9,12) in nodes a, b, c and d (from Fig. 5a), and the optimal (9,12)-assignment shown in Table 2.

The pair A-T was markedly preferred at node a (the average frequency is 0.9). At b the frequency of the pair A-T decreased and the frequency of C-G increased. At one descendant node, c, pair C-G was fixed (with average frequency 0.87), whereas in the other descendant, d, the A-T pair was reconstituted (with frequency 1.0).

The validity of the model has been established in several tests. Firstly, its low sensitivity as regards noise was established in numerical simulations with artificial and natural trees. Secondly, we have tried several different definitions of the main parameters, in particular, another definition of the branch quality, that would take into account not only the weight of the branch but the penalties of the optimal i-scenario, the significance of the branch in this scenario, etc., and they produced the same results as the simplest definition used here. Thirdly, when the algorithm was applied to palindromic motifs, it reconstructed scenarios with simultaneous changes in symmetric positions, and the results were the same when a general function F or a special palindromic function was applied. Finally, the results agree with the intuition gained by visual analysis of the data.

In most cases the changes were concentrated in few branches (at most, four), and these were usually terminal or almost terminal branches. This may be a consequence of the ascertainment bias in the data collection. Indeed, the analyzed sites were generated by the comparative genomic analysis, and thus strongly diverged motifs would have been missed. Another problem is that the contrast between distributions in some cases is insufficient to identify a change in the motif. It may be resolved when more data are available, thus increasing significance of the observed differences in positional nucleotide frequencies. A different problem is the small number of natural sites for many TFs that regulate just one or two operons. This makes estimation of nucleotide frequencies given counts insufficiently robust. We have dealt with that by merging site sets for closely related TFs, but it introduces an element of subjective decision that may influence the outcome.

On the other hand, the observed results seem to indicate that co-evolution of TFs and their binding motifs is shaped mainly by rare events with strong effects. Given that the most conserved positions in binding sites are those forming the largest number of contacts with the TF [4], that many specificity-determining positions in TFs are in direct contact with DNA [2], and that in some cases there are direct correlations between the type of a contacting residue in the TF and the consensus base pair in the DNA motif [1], it is likely that such events are caused by mutations in TF DNA-binding domains, more exactly, in amino acid residues forming immediate contact with DNA. At that, it is noteworthy that in many cases changes of consensus nucleotides occur via an intermediate step when the nucleotide frequencies at a given position are nearly uniform.

DATA AND METHODS

Protein multiple alignments were constructed using ProbCons [15]. Phylogenetic trees of transcription factors were constructed using Phyml [16]. Species trees were constructed using TiqMax [17], based on respective tree sets. Sequence logos were drawn using the program WebLogo [18]. DNA binding sites and motifs were taken from published and unpublished observations. They were generated using comparative genomic approaches described in detail in [6,19]. In a nutshell, motifs were identified by comparison of upstream regions of co-regulated and/or orthologous genes using SignalX [19], and the constructed recognition profiles were used to scan genomes in order to identify new sites of the same type using GenomeExplorer [19]. A candidate site was accepted if it was present upstream of several orthologous genes from sufficiently distant genomes.

ACKNOWLEDGEMENTS

This study was partially supported by grants from ISTC (3807), RFBR (07-01-00445), the Howard Hughes Medical Institute (55005610), and the Program “Molecular and Cellular Biology” of the Russian Academy of Sciences.

Appendix

Table A1
List of bacterial genomes and abbreviations

Abbr.	Bacteria	Abbr.	Bacteria
AF	*Acidithiobacillus ferrooxidans*	Mmic	*Mycobacterium microti*
AGR	*Agrobacterium tumefaciens*	Mther	*Methanosaeta thermophila*
Aple	*Actinobacillus pleuropneumonia*	Mjan	*Methanococcus jannaschii*
Avin	*Azotobacter vinelandii*	Mmarip	*Methanococcus maripaludis*
AFU	*Archaeoglobus fulgidus*	Mdeg	*Microbulbifer degradans*

Table A1, continued

Abbr.	Bacteria	Abbr.	Bacteria
Acin	*Acinetobacter* sp.	Nham	*Nitrobacter hamburgensis X14*
BSU	*Bacillus subtilis*	Nwi	*Nitrobacter winogradskyi*
Bli	*Bacillus licheniformis*	Nmen	*Neisseria meningitidis* MC58
BH	*Bacillus halodurans*	Ngon	*Neisseria gonorhoeae*
BCE	*Bacillus Cereus*	Nlac	*Neisseria lactamica*
BSt	*Bacillus stearothermophilus*	Neur	*Nitrosomonas europeae*
BT	*Bacteroides thetaiotaomicron*	OA	*Oceanicaulis alexandrii* HTCC2633
BF	*Bacteroides fragilis*	OB	*Oceanicola batsensis* HTCC2597
BME	*Brucella melitensis*	Oihey	*Oceanobacillus iheyensis*
BJ	*Bradyrhizobium japonicum*	Pu	*Pelagibacter ubique* HTCC1002
BPS	*Burkholderia pseudomallei*	PA	*Pseudomonas aeruginosa*
Burfu	*Burkholderia fungorum*	PZ	*Pseudomonas stutzeri*
Bmal	*Burkholderia mallei*	Pfluo	*Pseudomonas fluorescens*
Bcepa	*Burkholderia cepacia* R1808	PSsyr	*Pseudomonas syringae*
Bper	*Bordetella pertussis*	PP	*Pseudomonas putida*
Bbron	*Bordetella bronchiseptica*	PMI	*Petrotoga miotherma*
Bpar	*Bordetella parapertussis*	PM	*Pasteurella multocida*
Bav	*Bordetella avium*	PB	*Parvularcula bermudensis* HTCC2503
BQ	*Bartonella quintana*	Plu	*Photorhabdus luminiscens*
Bmar	*Bacteriovorax marinus*	Ppro	*Photobacterium profundum*
Cviol	*Chromobacterium violaceum*	Pol	*Polaromonas* sp. JS666
Cper	*Clostridium perfringens*	Pfil	*Polaribacter filamentus*
Cace	*Clostridium acetobutylicum*	PYR	*Pyrococcus* sp.
Cbot	*Clostridium botulinum*	RPA	*Rhodopseudomonas palustris*
Ctet	*Clostridium tetani*	Rsph	*Rhodobacter sphaeroides*
Cdif	*Clostridium difficile*	ROS	*Roseovarius* sp.
Cther	*Clostridium thermocellum*	Rrub	*Rhodospirillum rubrum*
Cdiph	*Corynebacterium diphtheriae*	Raeut	*Ralstonia eutropha*
Ceff	*Corynebacterium efficiens*	Rsola	*Ralstonia solanacearum*
Cglut	*Corynebacterium glutamicum*	Rmet	*Ralstonia metallidurans*
Chut	*Cytophaga hutchinsonii*	Rxyl	*Rubrobacter xylanophilus*
CC	*Caulobacter crescentus*	RHE	*Rhizobium etli*
DD	*Desulfovibrio desulfuricans G20*	RL	*Rhizobium leguminosarum*
DV	*Desulfovibrio vulgaris*	RB	*Rhodobacterales bacterium* HTCC2654
DP	*Desulfotalea psychrophila*	RC	*Rhodobacter capsulatus*
Daro	*Dechloromonas aromatica*	Rcon	*Rickettsia conorii*
Dace	*Desulfuromonas acetoxidans*	Rsib	*Rickettsia sibirica*
DR	*Deinococcus radiodurans*	SM	*Sinorhizobium meliloti*
Dgeo	*Deinococcus geothermalis*	SPO	*Silicibacter pomeroyi*
EC	*Escherichia coli*	Silib	*Silicibacter* sp.
EE	*Sulfitobacter* sp. EE-36	SKA	*Loktanella vestfoldensis* SKA53
EFA	*Enterococcus faecalis*	STAP	*Staphylococcus* sp.
ELI	*Erythrobacter litoralis*	STREP	*Streptococcus* sp.
ECA	*Erwinia carotovora*	Sala	*Sphinopyxis alaskensis* RB2256
ECH	*Erwinia chrysanthemi*	Saro	*Novosphingobium aromaticivorans*
ISM	*Roseovarius nubinhibens* ISM	SO	*Shewanella oneidensiss*
IDL	*Idiomarina loihiensis*	SPU	*Shewanella putrefaciens*
Jann	*Jannaschia* sp.	Styp	*Salmonella typhimurium*
FN	*Fusobacterium nucleatum*	SE	*Staphylococcus epidermidis*
HI	*Haemophilus influenzae*	SA	*Staphylococcus aureus*
HD	*Haemophilus ducreyi*	SCO	*Streptomyces coelicolor*
Gsul	*Geobacter sulfurreducens*	SAV	*Streptomyces avermilis*
Gmet	*Geobacter metallireducens*	Smel	*Sinerhizobium meliloti*
GOX	*Gluconobacter oxydans*	Tden	*Thiobacillus dentrificans*
Gkau	*Geobacillus kaustophilus*	TM	*Thermotoga maritime*

Table A1, continued

Abbr.	Bacteria	Abbr.	Bacteria
KP	*Klebsiella pneumoniae*	TTE	*Thermoanaerobacter tengcongensis*
Llac	*Lactococcus lactis*	TFU	*Thermobifida fusca*
Lmono	*Listeria monocytogenes*	TREP	*Treponema* sp.
Meso	*Mesorhizobium* sp.	TDE	*Treponema denticola*
METAN	*Methanosarcina* sp.	Vpar	*Vibrio parahaemolyticus*
MED	*Roseobacter* sp. MED193	VC	*Vibrio cholerae*
MM	*Magnetospirillum magneticum*	Vvul	*Vibrio vulnificus*
Mmag	*Magnetospirillum magnetotacticum*	Vfis	*Vibro fischeri*
Mlo	*Mesorhizobium loti*	XANT	*Xanthomonas* sp.
MS	*Mannheimia succiniciproducens*	Xfas	*Xylella fastidiosa*
MSMEG	*Mycobacterium smegmatis*	YE	*Yersinia enterocolitica*
Mmar	*Mycobacterium marinum*	YP	*Yersinia pestis*
MT	*Mycobacterium tuberculosis*	ZMO	*Zymomonas mobilis*

REFERENCES

[1] Rodionov, D. A., Dubchak, I. L., Arkin, A. P., Alm, E. J. and Gelfand, M. S. (2005). Dissimilatory metabolism of nitrogen oxides in bacteria: comparative reconstruction of transcriptional networks. PLoS Comput. Biol. **1**, e55.

[2] Kalinina, O. V., Mironov, A. A., Gelfand, M. S. and Rakhmaninova, A. B. (2004). Automated selection of positions determining functional specificity of proteins by comparative analysis of orthologous groups in protein families. Protein Sci. **13**, 443-456.

[3] Kotelnikova, E. A., Makeev, V. J. and Gelfand, M. S. (2005). Evolution of transcription factor DNA binding sites. Gene **347**, 255-263.

[4] Mirny, L. A. and Gelfand, M. S. (2002). Structural analysis of conserved base pairs in protein-DNA complexes. Nucleic Acids Res. **30**, 1704-1711.

[5] Lässig, M. (2007). From biophysics to evolutionary genetics: statistical aspects of gene regulation. BMC Bioinformatics **8 Suppl. 6**, S7.

[6] Rodionov, D. A. (2007). Comparative genomic reconstruction of transcriptional regulatory networks in bacteria. Chem. Rev. **107**, 3467-3497.

[7] Fitch, W. M. (1971). Towards defining the course of evolution: Minimum change for a specific tree topology. Syst. Zool. **20**, 406-416.

[8] Berg, O. G. and von Hippel, P. H. (1987). Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. J. Mol. Biol. **193**, 723-750.

[9] Stormo, G. D. (1991). Probing information content of DNA-binding sites. Methods Enzymol. **208**, 458-468.

[10] Gorbunov, K. and Lyubetsky, V. (2007). Reconstruction of ancestral regulatory signals along a transcription factor tree. Mol. Biol. (Mosk.) **41**, 836-842.

[11] Laikova, O. N. (2002). Systematic prediction of regulatory interactions in the LacI family of transcriptional regulators. Abstracts of the 5th Int. Conf. on Bioinformatics of Genome Regulation and Structure (BGRS'2002). IC&G, Novosibirsk **2**, 26-28.

[12] Gelfand, M. S. and Laikova, O. N. (2003). Prolegomena to the evolution of transcriptional regulation in bacterial genomes. In: Frontiers in computational genomics. Caister Academic Press, Wymondham, UK, pp. 195-216.

[13] Rodionov, D. A., Gelfand, M. S., Todd, J. D., Curson, A. R. and Johnston, A. W. (2006). Computational reconstruction of iron- and manganese-responsive transcriptional networks in alpha-Proteobacteria. PloS Comput. Biol. **2**, e163.

[14] Rodionov, D. A., Dubchak, I., Arkin, A., Alm, E. and Gelfand, M. S. (2004). Reconstruction of regulatory and metabolic pathways in metal-reducing delta-Proteobacteria. Genome Biol. **5**, R90.

[15] Do, C. B., Mahabhashyam, M. S. P., Brudno, M. and Batzoglou, S. (2005). ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Res. **15**, 330-340.

[16] Guindon, S. and Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. **52**, 696-704.

[17] Lyubetsky, V., Gorbunov, K., Rusin, L., V'yugin, V. (2006). Algorithms to reconstruct evolutionary events at molecular level and infer species phylogeny. An article in the book: Bioinformatics of Genome Regulation and Structure II. Springer Science & Business Media, Inc. 189-204.

[18] Crooks, G. E., Hon, G., Chandonia, J.-M. and Brenner, S. E. (2004). WebLogo: A sequence logo generator. Genome Res. **14**, 1188-1190.

[19] Mironov, A. A., Vinokurova, N. P., Gelfand, M. S. (2000). Software for analyzing bacterial genomes. Mol. Biol. (Mosk.) **34**, 253-262 (in Russian).