МОЛЕКУЛЯРНАЯ БИОЛОГИЯ, 2009, том 43, № 3, с. 527–541

МАТЕМАТИЧЕСКАЯ И СИСТЕМНАЯ БИОЛОГИЯ

УДК 575.852

РЕКОНСТРУКЦИЯ ЭВОЛЮЦИИ БАКТЕРИАЛЬНЫХ РЕГУЛЯТОРНЫХ СИГНАЛОВ, ОСНОВАННЫХ НА ВТОРИЧНОЙ СТРУКТУРЕ

© 2009 г. К. Ю. Горбунов*, Е. В. Любецкая, Е. А. Асарин, В. А. Любецкий

Институт проблем передачи информации им. А.А. Харкевича Российской академии наук, Москва, 127994

Поступила в редакцию 08.07.2008 г.
Принята к печати 21.10.2008г.

Предлагается алгоритм восстановления эволюции регуляторных сигналов, основанных на взаимодействии с вторичной структурой РНК. Алгоритм предполагает известным филогенетическое дерево видов, основан на допущении о консервативности вторичной структуры у рассматриваемых сигналов. Он получает на входе первичную структуру сигнала во всех листьях дерева и выдает первичную и вторичную структуры сигнала во всех вершинах дерева. Одновременно алгоритм строит множественное выравнивание современных (в листьях) сайтов регуляторного сигнала с учетом его вторичной структуры. Приведены результаты успешного тестирования алгоритма на трех основных типах аттенюаторной регуляции у бактерий: классической аттенюаторной (биосинтез треонина и лейцина у гамма-протеобактерий), T-боксовой (у актинобактерий), RFN-регуляции (у эубактерий).

***Ключевые слова*: эволюционный сценарий, регуляторный сигнал, эволюция вдоль дерева, дерево видов, вторичная структура.**

MODELING EVOLUTION OF REGULATORY SIGNALS FOR GENE EXPRESSION IN BACTERIA, by *K. Yu. Gorbunov*, E. V. Lyubetskaya, E. A. Asarin, V. A. Lyubetsky* (Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, 127994 Russia; *e-mail: gorbunov@iitp.ru). A model of evolution of a regulatory signal along the phylogenetic tree of species taking into account the secondary RNA structure is suggested. Based on this model, an algorithm is presented. It inputs the extant primary structure of a signal for the leaves of the phylogenetic tree and computes the primary and secondary structures of all the nodes. Another result of the algorithm is a multiple alignment of extant sites of a regulatory signal taking into account the secondary structure of the signal. The algorithm has been implemented and successfully tested on biological data representing three types of regulation in bacteria.

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

ПОСТАНОВКА ЗАДАЧИ

Две проблемы. Проблема (далее проблема 1), которую мы рассмотрим, состоит в реконструкции предковых состояний некодирующих участков генома, а именно – участков (сайтов) РНКовых регуляций аналогично тому, как эта проблема ставится для реконструкции кодирующих участков генома (структурных генов).

Эта проблема связана, но не совпадает с известной проблемой (далее – проблема 2) построения множественного выравнивания заданного набора нуклеотидных последовательностей с учетом предполагаемой (но *заранее* не известной) общей вторичной структуры в них. Существуют алгоритмы для построения множественного выравнивания без учета общей вторичной структуры и, если известен список консервативных спиралей, составляющих эту предполагаемую общую вторичную структуру, то упомянутые алгоритмы или их усовершенствования можно использовать для решения проблемы 2. Но такой набор консервативных спиралей обычно заранее не известен и его определение естественно связать (по крайней мере, так делаем мы) с решением проблемы 1: искомые консервативные спирали – те, которые устойчивы в ходе эволюции первичной структуры. В этом смысле, решение проблемы 1 дает искомый список консервативных спиралей из вторичных структур в заданном наборе современных сайтов РНКовой регуляции и, тем самым, ведет к решению проблемы 2. Эта статья далее посвящена проблеме 1.

Обзор современного состояния проблемы. Проблема *построения филогенетического дерева* белкового семейства широко известна, по ней проводили активные исследования [1–5]. Связанные с ней вопросы принадлежат еще более широкой области

* Эл. почта: gorbunov@iitp.ru

моделирования эволюции видов, которой посвящена обширная литература [6–8]. Эволюционные модели включают реконструкцию событий молекулярного уровня, таких как дупликации, возникновения, горизонтальные переносы и потери гена, а также включают построение и оценку эволюционных сценариев, см., например, [9–14]. Например, в работе [15] она выполняется вместе с определением длин ветвей дерева методом максимального правдоподобия, а в [16] – совместно с построением предковых последовательностей при условии, что исходные данные заранее снабжены множественным выравниванием, см. также сайт http://evolution.genetics.washington.edu/phylip / software.serv.html, на котором приведен список известных программ для построения филогенетического дерева и реконструкции предковых последовательностей. Соответствующие исследования связаны с моделированием самих эволюционных событий, см., например, [17–21].

С другой стороны, в последние два десятилетия в биоинформатике значительная часть исследований посвящена механизмам регуляции экспрессии генов, особенно у бактерий. Среди таких механизмов выделяются широко исследуемая белок-ДНКовая регуляция, классическая аттенюаторная регуляция, Т-боксовая регуляция, регуляция на основе РНКовых переключателей (рибосвичей) и с использованием некоторых специальных белков, таких как TRAP, регуляция LEU-элементами и т.д. Основными проблемами здесь являются поиск регуляторных элементов (регуляторных сайтов), функциональный и эволюционный анализ найденных сигналов и моделирование процессов регуляции, см., например, [22–27]. Есть сотни публикаций на эту тему. Биологические аспекты эволюционирования регуляторных структур описаны в работе [28]. Что касается алгоритмических аспектов, то до последнего времени реконструкция предковых последовательностей происходила в основном без учета вторичных структур или лишь с косвенным их учетом. Авторы работы [29] осуществляли учет вторичных структур за счет модификации стандартной модели нуклеотидных замен, а именно, добавляли события одновременной замены двух комплементарных нуклеотидов в связанных позициях. Можно отметить применение здесь генетических алгоритмов моделирования эволюции спирали для оценки скоростей замен нуклеотидов и оценки длин ветвей дерева [30], а также учет эволюции вторичных РНКовых структур с целью построить более правдоподобное филогенетическое дерево [31].

Наш подход. Мы предполагаем, что естественным продолжением этих двух направлений исследований является моделирование процесса *эволюции регуляторного сигнала вдоль филогенетического дерева* видов или белков (например, фактора транскрипции), или вдоль другого дерева, соответствующего эволюции данной регуляции. Нами изучены такие эволюционные процессы для регуляторных сигналов метаболизма аминокислот, когда механизм этих сигналов основывается на альтернативных вторичных структурах РНК. Мы развиваем два подхода к моделированию такой эволюции. В обоих подходах *дано филогенетическое дерево* и регуляторные сайты в его листьях (одного типа, например, сайты классической аттенюаторной регуляции), и *ищутся предковые состояния этих сайтов* во всех внутренних вершинах дерева, включая *вторичные структуры* в предковых и современных последовательностях.

Наш первый подход подробно изложен в работе [32] и представлен на конференциях [33, 34], см. также интернет-сайт: http://lab6.iitp.ru, пункт 4. Опишем этот подход для полноты картины в нескольких фразах. Явно выписывается функционал $H(\sigma)$ гиббсовского типа, аргумент σ которого является конфигурацией – функцией, которая приписывает каждой внутренней вершине дерева последовательность нуклеотидов (предполагаемое предковое состояние регуляторного сигнала в этой вершине). Таким образом, конфигурация представляет собой совокупность предковых состояний регуляторного сигнала, который задан в листьях. Этот функционал $H(\sigma) = H_1(\sigma) + H_2(\sigma)$ включает два условия (ограничения) на искомую конфигурацию σ : 1) для каждой последовательности $\sigma(k)$ (т.е. для значения σ в k-й вершине дерева) и вдоль каждого ребра, выходящего из этой вершины, в каждой позиции $i = 1, \ldots, n$ этой последовательности $\sigma(k)$ происходит независимая замена букв в соответствии с матрицей замен R, а также происходят вставки и удаления; 2) последовательности $\sigma(k)$ из конфигурации σ, по возможности, сохраняют вторичную структуру от начала ребра к его концу и вдоль целого пути в дереве (в течение многих поколений), при этом функционал меньше, если такие пути длиннее и их больше. Первое условие представлено в слагаемом $H_1(\sigma)$, описывающем стандартную динамику первичной структуры регуляторного сайта, а второе условие – в слагаемом $H_2(\sigma)$, описывающем такую динамику первичной структуры сайта, при которой поддерживается высокая консервативность вторичной структуры. Ищется глобальный минимум функционала $H(\sigma)$, который, как предполагается согласно первому подходу, соответствует биологической эволюции регуляторного сигнала, заданного в листьях дерева посредством лишь первичной структуры. Таким образом, $H(\sigma)$ выражает биологически мотивированные принципы обычной *эволюции* (динамики) *первичной структуры и* одновременно *консервативности вторичной структуры.* Этот подход

эффективно компьютерно реализован и тестирован на тех же примерах, которые приводятся ниже [32].

В данной работе мы описываем наш второй подход. Он основан на двух других биологически мотивированных требованиях: вторичная структура сайта должна быть как можно более консервативной вдоль путей в дереве (от листьев к корню, здесь, как и в первом подходе), см. ниже этап 1 алгоритма. И первичная структура регуляторного сайта должна допускать как можно *меньше эволюционных событий* (принцип парсимонии), см. ниже этап 2 алгоритма.

Итак, первый подход основан на обычной модели замен нуклеотидов и требовании консервативности, а второй – на принципе парсимонии и том же требовании консервативности. Эти требования отражают общепринятые представления об эволюции. Таким образом, концептуально эти два подхода отличаются по их первым требованиям. Но, конечно, они радикально отличаются еще по реализации этих требований: в первом подходе происходит минимизация методом аннилинга функционала (экспоненты от матрицы замен, инделей и некоторых сложных математических выражений, описывающих степень консервативности), а во втором – поочередные выравнивания и минимизации совершенно другого функционала.

Можно по-разному реализовать два требования, содержащиеся во втором подходе: например, определить единый (для всего алгоритма) функционал, минимум которого соответствует итоговой конфигурации предковых сайтов, или приходить к этой итоговой конфигурации итерационно так, как описано в алгоритме ниже.

Смысл нескольких подходов к решению проблемы 1 в том, чтобы сравнить их результаты между собой. Это способ проверить их адекватность, так как экспериментальное получение данных о предковых состояниях регуляторных сигналов у бактерий, по-видимому, еще более затруднительно, чем о предковых состояниях генов. Отметим, что результаты двух указанных подходов на биологических и искусственных данных дали удивительно сходные результаты счета.

Мы компьютерно реализовали модель и алгоритм, который тестировали на многих примерах классической аттенюаторной регуляции, Т-боксовой регуляции, РНКовых переключателях, LEU-регуляции и т.д. Отметим, что наша модель единообразно обслуживает эволюционный анализ этих весьма разных регуляций.

Множественное выравнивание исходных последовательностей в листьях, вообще говоря, неизвестно и ни в какой мере не предполагается данным в нашей модели. Тестируя наш алгоритм, мы заметили, что выдаваемые им выравнивания и вторичная структура близки к известным или предполагаемым (по независимым исследованиям) в последовательностях, заданных алгоритму в листьях. Кроме того, алгоритм выдает множественное выравнивание всех найденных сайтов во всех вершинах дерева с учетом совместной эволюции их вторичных структур. Фактически алгоритм в качестве побочного продукта выдает и множество эволюционно консервативных спиралей, что является принципиальным шагом в решении проблемы 2.

Как упомянуто выше, в данной работе подробно описывается наш второй подход к проблеме 1: к построению эволюции регуляторных сигналов (с вторичной структурой в них) в бактериях с использованием итеративной процедуры на конфигурациях следующего нового типа. Такая конфигурация приписывает каждой вершине дерева не последовательность нуклеотидов (как, в частности, это имеет место в нашем первом подходе), а последовательность *распределений*: распределение задает частоты всех нуклеотидов и, возможно, символа пробела.

В разделах “Описание модели”, “Алгоритм” и “Примеры применения метода к анализу биологических данных” подробно описана наша модель эволюции регуляторного сигнала с вторичной структурой, приводится алгоритм построения предковых состояний сигнала, заданного в листьях без указания вторичной структуры, и показано применение алгоритма к биологическим примерам для различных типов регуляций.

ОПИСАНИЕ МОДЕЛИ

Как отмечено выше, модель основана на двух естественных требованиях: вторичная структура сайта должна быть как можно более консервативной вдоль путей в дереве (от листьев к корню) – см. ниже этап 1, а первичная структура регуляторного сайта должна иметь как можно меньше эволюционных событий – см. ниже этап 2.

Особенность предлагаемой модели состоит в следующем. Каждой вершине дерева видов приписана последовательность, у которой в каждой позиции находятся частоты пяти знаков: четырех букв A, C, T, G и символа пробела *d*. Таким образом, к вершинам относятся последовательности разной длины, состоящие из векторов (распределений) длины 5, а каждый вектор состоит из чисел от 0 до 1, в сумме равных 1; эти числа соответствуют вероятностям букв A, C, T, G и *d* (пусть далее числа соответствуют этим буквам в указанном здесь порядке). Такую последовательность векторов будем называть *последовательностью распределений*. Затем в каждой вершине того же дерева видов приписанная ей последовательность распределений естествен-

ным образом преобразуется в обычную *нуклеотидную последовательность с пробелами* (этап 3, см. ниже). Каждая последовательность характеризует или прямо представляет собой предполагаемый сайт в данной вершине дерева для одной фиксированной регуляции, в механизме которой существенно поддержание в эволюции вторичной регуляторной структуры РНК. Листьям дерева кроме последовательностей распределений приписаны и нуклеотидные последовательности с пробелами. В начале работы алгоритма использованы исходные регуляторные сайты, которые по ходу его работы будут пополняться пробелами.

Обозначим σ какую-то последовательность распределений с переменной длиной n, пусть σ_i – ее i-й член, где $1 \le i \le n$, индекс i назовем i-й позицией в σ. Обозначим $X(i, \sigma)$ долю буквы X в i-ой позиции последовательности σ. Последовательность распределений, которая в каждой позиции имеет распределение с одной единицей и остальными нулями, и соответствующая ей нуклеотидная последовательность с пробелами далее не различаются. Обратим внимание, что для каждого листа алгоритм выдает два варианта нуклеотидной последовательности: один из них на рисунках с четными номерами помечен номером листа, а другой – именем вида. Повторим: в самом начале работы алгоритма в каждом листе дана исходная последовательность без пробелов.

В модели рассматривают выравнивания первичных структур последовательностей распределений, т.е. самих этих последовательностей как слов в бесконечном алфавите, а также их вторичных структур как слов, составленных из "спиралей" в последовательностях распределений. Выравнивание понимается как вставка пробелов в два данных слова в некотором алфавите. С этой целью к словам применяется один из обычных алгоритмов выравнивания, который использует определенные нами веса. Новым является рассмотрение вторичных структур в последовательности распределений. Определим понятие спирали, взятое выше в кавычки.

В последовательности σ распределений две позиции i и j назовем комплементарными, если следующая сумма

$$\begin{gathered}[\min(A(i,\sigma),T(j,\sigma)) + \min(T(i,\sigma),A(j,\sigma))] + \\ + [\min(C(i,\sigma),G(j,\sigma)) + \min(G(i,\sigma),C(j,\sigma))] + \\ + 0.5\,[\min(G(i,\sigma)-\min(G(i,\sigma),C(j,\sigma)),\ \min(T(j,\sigma)- \\ \min(T(j,\sigma),A(i,\sigma))) + \min(T(i,\sigma)-\min(T(i,\sigma),A(j,\sigma)), \\ \min(G(j,\sigma)-\min(G(j,\sigma),C(i,\sigma)))] + \\ + 0.25\,[\min(d(i,\sigma),d(j,\sigma)]\end{gathered}$$

больше некоторого порога, называемого *порогом комплементарности*. В примерах 1–4 его значение равно 0.5. Определение комплементарности отражает связанность двух позиций, в которых находятся соответствующие распределения. Из него обычным образом возникают определения спирали и шпильки в последовательности распределений (см., например, [26, 35]). Например, спираль образуется спариванием максимальных по длине отрезков попарно комплементарных позиций в последовательности распределений. Эти отрезки, как обычно, назовем *плечами*. Таким образом, определения комплементарности и спирали для последовательности распределений являются естественным обобщением обычных определений для нуклеотидной последовательности.

Чтобы определить понятие *энергии спирали,* обобщим стандартное определение (см., например, [26]), в котором при подсчете энергии суммирование выполняется по четверкам нуклеотидов (по парам соседних спариваний нуклеотидов). У нас же оно выполняется по аналогичным четверкам распределений. В случае такой фиксированной четверки F распределений обозначим $X(i)$ частоту встречаемости нуклеотида X в i-ой компоненте четверки. Тогда слагаемое в энергии, соответствующее F, равно $\sum_{f = \langle X, Y, Z, V\rangle} E_f \dfrac{X(1)Y(2)Z(3)V(4)}{S}$, где суммирование выполняют по всем возможным четверкам f нуклеотидов, а E_f – вклад четверки f в энергию при стандартном ее вычислении, нормализующий делитель S равен $S = \sum_{f = \langle X, Y, Z, V\rangle} X(1)Y(2)Z(3)V(4)$. Как обычно, к вычисленной таким образом энергии добавляются слагаемые, учитывающие внутренние и внешние петли. Длиной петли считаем сумму всех частот нуклеотидов в ней. Дополнительные слагаемые (энтропия) вычисляют для любых рациональных аргументов путем линейной интерполяции известных значений, взятых из [26], для целых аргументов. Например, если $f(x, y)$ – стандартная функция, определяющая энтропию от внутренней петли с длинами сторон $x = 2.75$ и $y = 3.25$, то

$$\begin{aligned} f(2.75, 3.25) &= \frac{1}{4}f(2, 3.25) + \frac{3}{4}f(3, 3.25) = \\ &= \frac{1}{4}\left(\frac{3}{4}f(2, 3) + \frac{1}{4}(2, 4)\right) + \frac{3}{4}\left(\frac{3}{4}f(3, 3) + \right. \\ &\left. + \frac{1}{4}f(3, 4)\right) = \frac{3}{16}f(2, 3) + \frac{1}{16}f(2, 4) + \\ &+ \frac{9}{16}f(3, 3) + \frac{3}{16}f(3, 4). \end{aligned}$$

Цена за сопоставление двух распределений или распределения и пробела при парном выравнивании двух последовательностей распределений определяется следующим образом. Обозначим $X(1)$ и $X(2)$ частоты знака X в первом и, соответственно, во втором распределениях; пробел, как обычно, обозначается символом d. Тогда эта цена равна

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

где $C = \sum_{X = \{A, C, T, G\}} \min(X(1), X(2))$, R – стандартный приз за совпадение двух нуклеотидов, P_1 – стандартный штраф за сопоставление нуклеотиду пробела и P_2 – стандартный штраф за несовпадение двух нуклеотидов. В некоторых более сложных случаях нами за сопоставление нуклеотидов A и G или соответственно C и T брался меньший штраф, чем за сопоставление других пар нуклеотидов.

АЛГОРИТМ

Пусть каждому листу дерева приписана одна исходная нуклеотидная последовательность без пробелов. Итерации состоят в последовательной смене этапов 1 и 2. Предлагаемый алгоритм приписывает последовательности распределений всем вершинам дерева от листьев к корню указанным ниже способом и одновременно выполняет множественное выравнивание всех этих последовательностей. При этом фактически приходится выполнять выравнивание только двух слов, что является легкой задачей.

Этап 1. Каждому листу дерева приписана одна последовательность распределений, возникшая в ходе вычислений на предыдущей итерации, этапе 2, и называемая *концевой последовательностью распределений*. В начале работы алгоритма концевая последовательность распределений в листе задается просто как копия исходной нуклеотидной последовательности в этом листе, в которой каждое распределение состоит из единицы и нулей. Нам потребуется еще понятие измененной *нуклеотидной последовательности*; в общем случае это – результат работы этапа 1, а в начале это – копия исходной нуклеотидной последовательности. Выравниваем эти концевую и измененную нуклеотидную последовательности для каждого листа между собой без учета их вторичных структур. При этом они могут получить некоторое число пробелов. В самом начале эти последовательности в каждом листе совпадают и соответственно их выравнивание тривиальное. Результатом этапа 1, передаваемым на этап 2, является измененная нуклеотидная последовательность в каждом листе. Результатом этапа 2, передаваемым на этап 1, служит концевая последовательность распределений в каждом листе. На протяжении всего алгоритма измененная нуклеотидная последовательность в листе отличается от исходного данного в этом листе только добавлением пробелов в последнее.

Пусть двум сыновьям ν_1 и ν_2 вершины ν уже приписаны последовательности распределений σ_1 и σ_1. Алгоритм должен определить последовательность распределений σ в ν. Для этого он сначала *выравнивает вторичные структуры* этих последовательностей следующим образом. По σ_1 и σ_2 определим соответственно два множества Ω_1 и Ω_2, состоящие из спиралей (соответственно в первой и во второй последовательностях) с энергией выше порога, который в примерах 1–4 принимается равным 10 ккал/моль. Этот порог назовем энергетическим.

От множеств Ω_1 и Ω_2 алгоритм переходит к линейным упорядочениям плеч спиралей из этих множеств. Точнее, алгоритм упорядочивает плечи в линейном порядке: грубо говоря, в том порядке, в котором они расположены в последовательности, а точнее по середине плеча и при совпадении этих середин по началам плеч. Каждое плечо включается в это упорядочение как буква из некоторого нового алфавита, за которой стоит информация о начале и конце плеча вместе с его окрестностью некоторого фиксированного размера, нуклеотидном составе плеча и окрестности, номере спирали, от которой оно взято, информация о том, является ли оно левым или правым. Эти упорядочения назовем *словами* соответственно в вершинах ν_1 и ν_2 и обозначим их также Ω_1 и Ω_2 (хотя можно брать и деревья в качестве Ω_1 и Ω_2 – получается результат, близкий к варианту слов). Выравниваем эти слова. Затем, с учетом таким образом полученного выравнивания вторичных структур в σ_1 и σ_2, выравниваем сами последовательности σ_1 и σ_2. Подробно такая процедура описана в работе [36]. Нами определяются веса для таких выравниваний: за соответствие двух распределений или двух плеч, за соответствие распределения или плеча пробелу, которые согласуются с обычной моделью замен нуклеотидов. Здесь веса не приводятся, отметим только случай двух плеч. Тогда вес берется следующим образом: качество, полученное при выравнивании плеч вместе с их окрестностями как нуклеотидных последовательностей, складывается с призом за каждое соответствие в выравнивании левого и правого плеч от одной спирали, при этом разрешается соответствие только левого плеча на левое и правого плеча на правое.

Теперь для каждой позиции i в качестве распределения σ_i берем взвешенное среднее распределений σ_{1i} и σ_{2i} с весами, которые определяются отношением длин двух ребер из ν в ν_1 и в ν_2, т.е.

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

где $l(\nu,\nu_i)$ обозначает длину ребра из вершины ν в вершину ν_i.

Получаем последовательность распределений σ в вершине ν, и затем бывшие σ_1 и σ_2 в ν_1 и ν_2 заме-

няем на полученные в результате выравнивания; последние будем обозначать также σ_1 и σ_2. Очевидно, новые σ_1 и σ_2 получаются из старых путем согласованного добавления в старые некоторого числа пробелов. Затем продолжим добавление пробелов во все последовательности распределений, приписанные ниже σ, включая измененные и *исходные* нуклеотидные последовательности, приписанные листьям дерева. Теперь все эти потомки последовательности σ имеют одинаковую длину.

Когда этап 1 доходит до корня, в листьях образуются измененные нуклеотидные последовательности, которые являются единственным *результатом этапа* 1 – он передается этапу 2. Все последовательности (распределений и нуклеотидные), приписанные вершинам в конце этапа 1, имеют одинаковую длину, которую назовем *длиной зоны*. Отметим, что каждому листу приписывается как последовательность распределений, так и измененная нуклеотидная последовательность с пробелами. На протяжении работы алгоритма измененная нуклеотидная последовательность в листе отличается от исходного данного в этом листе только добавлением пробелов. Переходим к этапу 2.

Этап 2. Для *каждой позиции* полученных на этапе 1 измененных нуклеотидных последовательностей выполняем следующее. Каждой вершине ν эволюционного дерева сопоставляется набор $\delta(\nu)$ частот пяти возможных символов в рассматриваемой позиции. На этапе 2 эти значения считаются переменными, и по всем по ним производится минимизация (поиск ближайшего локального минимума) для описанного ниже функционала F.

Пусть ρ – это некоторая мера близости между векторами (в простейшем случае – сумма квадратов разностей компонент); для каждого листа ν через $\sigma(\nu)$ обозначим константный вектор, в котором символу измененной нуклеотидной последовательности, приписанной этому листу, соответствует 1, а остальные элементы – 0; через $e_н$ и $e_к$ обозначим концы ребра e. Определим функционал F следующей формулой (первое суммирование происходит по всем ребрам дерева, второе – по всем его листьям):

$$F = \sum_e \rho(\delta(e_н), \delta(e_к)) \cdot w(e) + \sum_\nu \rho(\delta(\nu), \sigma(\nu)) \cdot w(\nu).$$

Здесь $w(e)$, $w(\nu)$ – весовые коэффициенты. Вес $w(e)$ тем больший, чем меньше длина ребра e; вес $w(\nu)$ равен весу $w(e)$ ребра e от листа ν к родителю умноженному на *специальный* параметр, который регулирует близость распределений в листьях к исходным данным относительно консервативности распределений во внутренних вершинах дерева. В примерах 1–4 этот параметр равен 1.5.

Минимизация (поочередно для каждой позиции) проводится по всем переменным, т.е. по долям пяти символов во всех вершинах дерева. Накладываются естественные ограничения: все переменные неотрицательны и в каждой вершине сумма соответствующих пяти переменных равна единице. В упомянутом простейшем случае функционал квадратичный с единственной точкой минимума, так что эта точка легко находится методом квадратичного программирования. Полученные в результате этой минимизации концевые (в листьях) последовательности распределений – единственный *результат этапа* 2, который передается этапу 1.

Мы рассматривали усложнение этого алгоритма, в котором отдельным этапом учитывали мутации букв по одной из моделей замен, а также вставки и делеции в первичной структуре. Это не приводит к заметной разнице для классической аттенюаторной регуляции. В случае других регуляций ситуация более сложная и здесь не рассматривается.

Алгоритм заканчивает чередование этапов 1–2 (на этапе 1), если длина зоны перестает расти. Такой выбор критерия для остановки чередования этапов 1–2 выбран из следующего эвристического наблюдения: когда найдено хорошее множественное выравнивание нуклеотидных последовательностей в листьях с учетом консервативной вторичной структуры, алгоритм зацикливается и длина зоны перестает расти. Мы не утверждаем, что так будет на любых исходных данных, но в рассматриваемых примерах происходило так. После этого алгоритм переходит к этапу 3, используя в нем только *итоговое множественное выравнивание* последовательностей распределений во всех вершинах дерева, полученное после этапов 1–2.

Этап 3. В итоговом множественном выравнивании переходим от последовательностей распределений к *новым* нуклеотидным последовательностям с пробелами, приписанным всем вершинах дерева, включая листья. Все эти новые последовательности имеют ту же постоянную длину, что и в итоговом множественном выравнивании последовательностей распределений; именно эти новые последовательности показаны на рисунках с четными номерами, где они помечены номером соответствующей вершины дерева; при этом именем вида помечены исходные нуклеотидные последовательности с пробелами, которые в них вставились по ходу алгоритма. А именно, если позиция не входит в спираль, то в ней располагаем символ (нуклеотид или пробел), который имеет наибольшую частоту в распределении соответствующем этим позиции и вершине де-

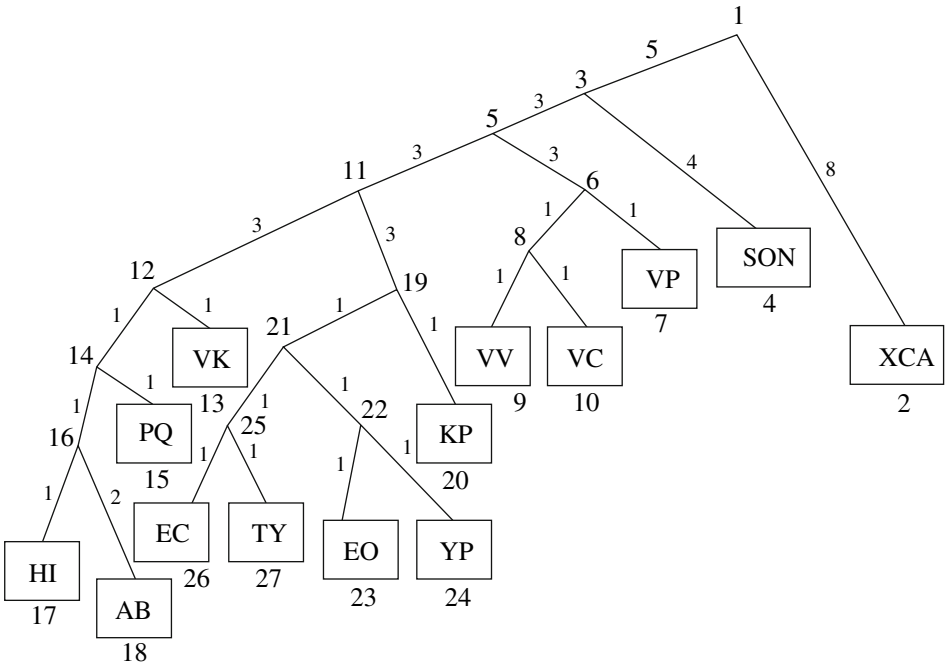

Рис. 1. Филогенетическое дерево для классической аттенюаторной регуляции биосинтеза треонина у гамма-протеобактерий.

рева. Если эта позиция участвует в одной или в нескольких спиралях, то в ней располагаем букву, которая с наибольшей энергией спарилась в составе наибольшего числа спиралей и еще имеет достаточную частоту. Смотрим, какая из четырех букв имеет наилучшее значение следующего показателя: сумма энергий всех спиралей, в которые входит данная позиция, где каждое слагаемое умножается на вес, равный той части частоты буквы, которая участвует в данной спирали; если ни одна из четырех букв не набирает некоторого порога по этому показателю, то ставим пробел. На рисунках с четными номерами приводится *итоговое множественное выравнивание* новых нуклеотидных последовательностей, полученных после этапа 3, вместе с измененными *нуклеотидными последовательностями* с пробелами, полученными при последнем применении этапа 1 – то же, что исходные последовательности с вставленными по ходу алгоритма пробелами. На рисунках первые последовательности помечаются номером вершины, а вторые – именем вида.

Этап 4. Отбираем по некоторому порогу наиболее консервативные спирали в итоговом множественном выравнивании новых нуклеотидных последовательностей в листьях и по выравниванию указываем соответствующие спирали в измененных нуклеотидных последовательностях (см. упомянутые выше рисунки) и выдаем индуцированные ими спирали в исходных нуклеотидных последовательностях. Таким образом, в исходных данных указывается эволюционно индуцированная вторичная структура, которая не была *заранее* известна алгоритму. Это позволяет независимо провести множественное выравнивание исходных нуклеотидных последовательностей с учетом этой вторичной структуры. Получается результат близкий к тому, который возникает после этапа 3.

ПРИМЕРЫ ПРИМЕНЕНИЯ МЕТОДА К АНАЛИЗУ БИОЛОГИЧЕСКИХ ДАННЫХ

Ввиду предварительного характера публикации и недостатка места мы ограничиваемся рассмотрением четырех примеров: по 1–2 примерам на каждый из основных типов регуляции, основанной на вторичной структуре мРНК. На этих примерах известные программы PAML и PAUP (см. раздел “Обсуждение результатов”) показали существенно худшие, чем у нас, результаты (данные не приводятся). Более полные результаты тестирования приведены на сайте лаборатории по адресу http://lab6.iitp.ru, пункт 9.

Пример 1. Рассмотрим классическую аттенюаторную регуляцию биосинтеза треонина у гамма-протеобактерий. Исходные сайты в листьях взяты из работы [22] (алгоритм не использует ни заранее известную информацию о вторичной структуре, ни множественное выравнивание). Берется стандартное дерево видов (рис. 1), в котором вершины нуме-

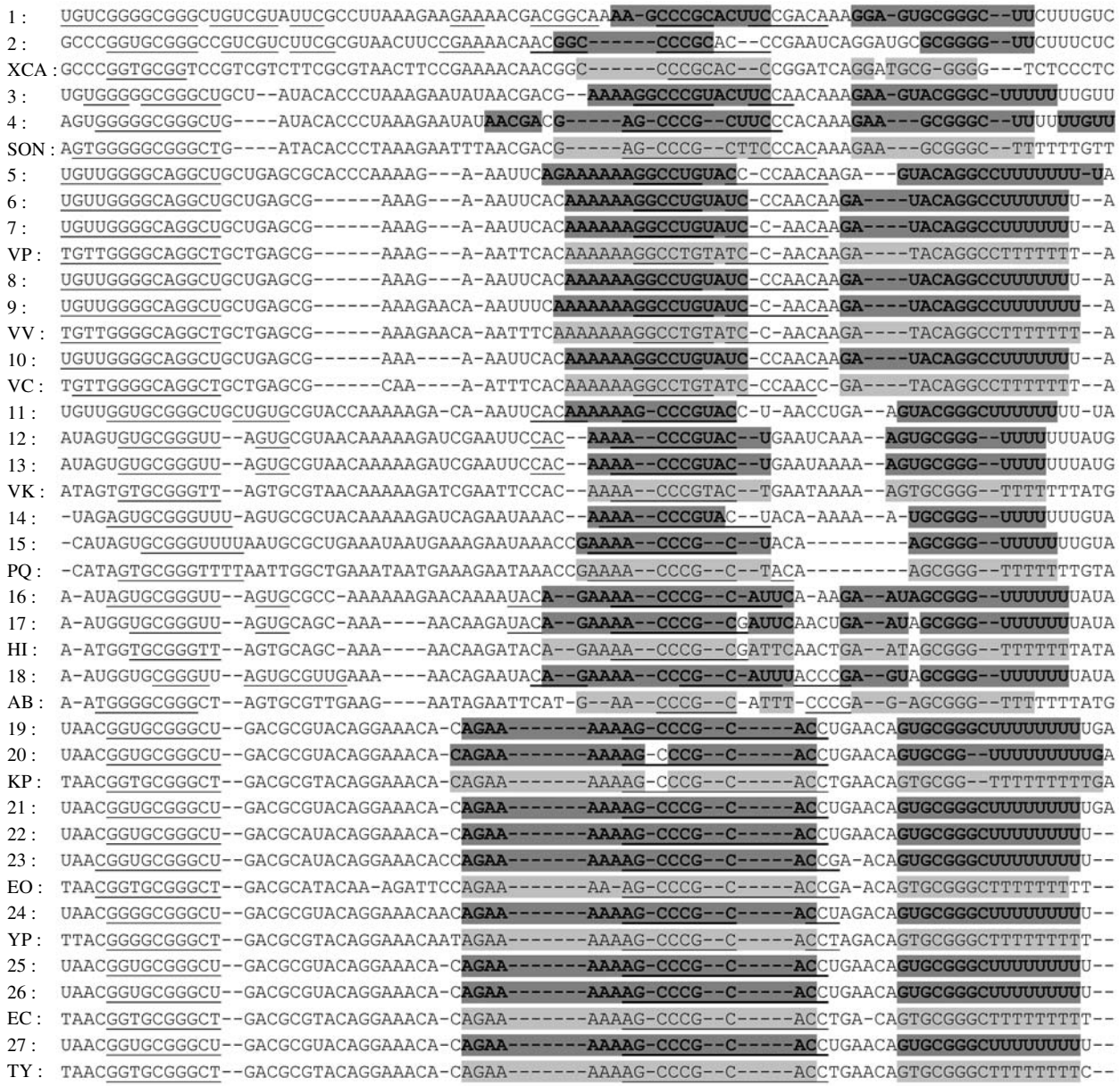

Рис. 2. Полученное алгоритмом множественное выравнивание (с вторичной структурой) предполагаемых регуляторных сайтов для классической аттенюаторной регуляции биосинтеза треонина у гамма-протеобактерий.

рованы числами от 1 до 27 (номера листьев стоят под прямоугольниками), каждому ребру приписана его филогенетическая длина в условных единицах, число меньшего размера. Использованы сокращения: EC – *Escherichia coli*, TY – *Salmonella typhi*, KP – *Klebsiella pneumoniae*, EO – *Erwinia carotovora*, YP – *Yersinia pestis*, HI – *Haemophylus influenzae*, VK – *Pasterella multocida*, AB – *Actinobacillus actinomycetemcomitans*, PQ – *Mannheimia haemolytica*, VC – V*ibrio cholerae*, VV – *Vibrio vulnificus*, VP – *Vibrio parahaemolyticus*, SON – *Shewanella oneidensis*, XCA – *Xanthomonas campestris*.

Алгоритм выдает итоговое множественное выравнивание нуклеотидных последовательностей – предполагаемые регуляторные сайты и измененные нуклеотидные последовательности, см. рис. 2. Терминатор показан темно-серым, антитерминатор – подчеркиванием. Терминатор в измененных нуклеотидных последовательностях показан светло-серым. Полученные в листьях вторичные структуры (рис. 2) лишь немногим отличаются от вто-

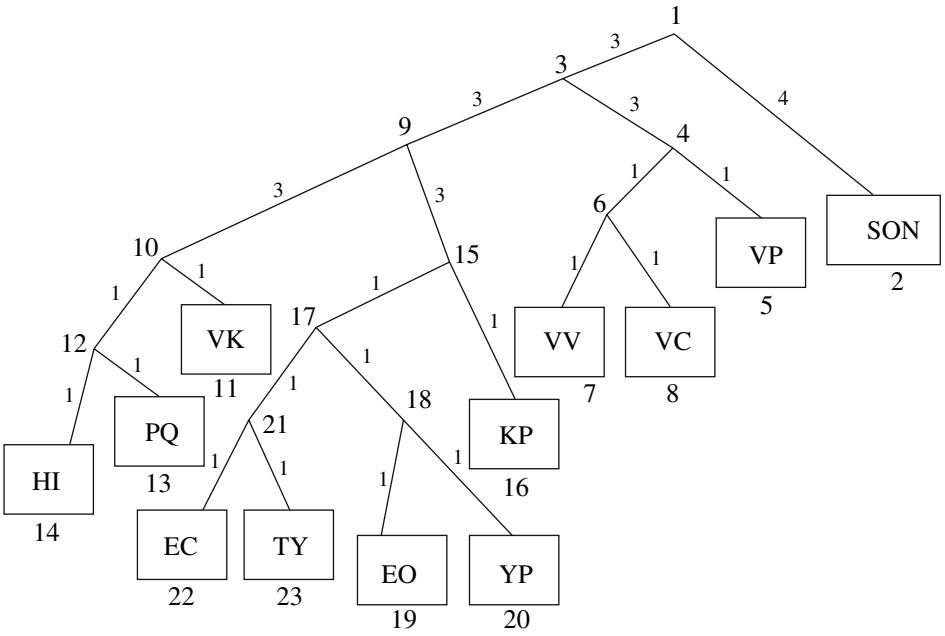

Рис. 3. Филогенетическое дерево для классической аттенюаторной регуляции биосинтеза лейцина у гамма-протеобактерий.

```
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------TTCCAGCAAGACTATAAACCC-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
```

Рис. 4. Полученное алгоритмом множественное выравнивание (с вторичной структурой) предполагаемых регуляторных сайтов для классической аттенюаторной регуляции биосинтеза лейцина у гамма-протеобактерий.

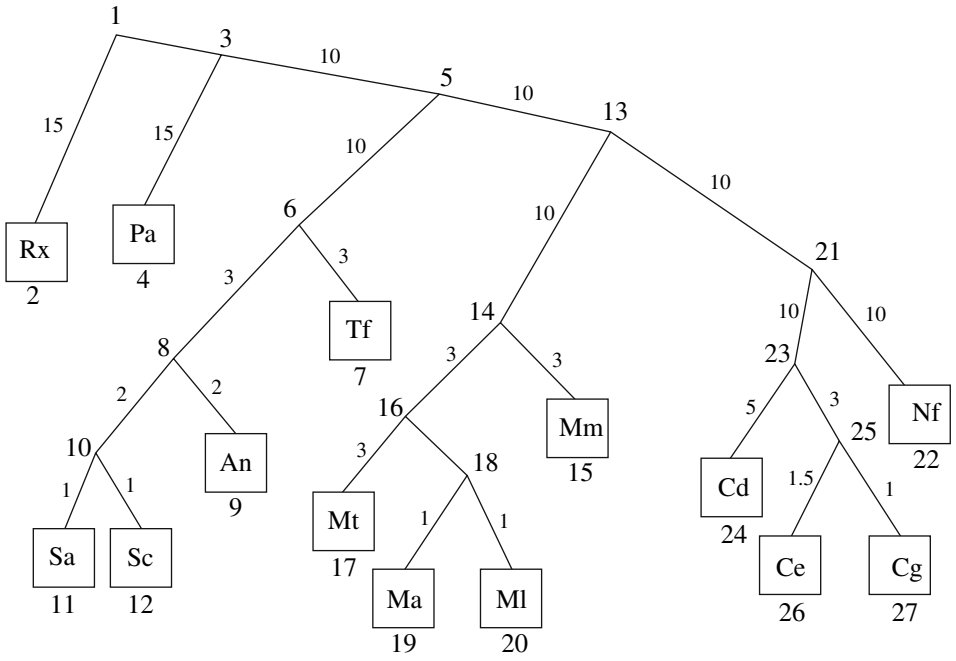

Рис. 5. Филогенетическое дерево для Т-боксовой регуляции гена *ileS* у актинобактерий.

ричных структур, предсказанных другими биоинформатическими методами, либо известных из эксперимента [26].

Пример 2. Рассмотрим классическую аттенюаторную регуляцию биосинтеза лейцина у гамма-протеобактерий. Исходные сайты в листьях взяты из работы [22]. Возьмем стандартное дерево видов (рис. 3).

Алгоритм выдает итоговое множественное выравнивание нуклеотидных последовательностей – предполагаемые регуляторные сайты и измененные нуклеотидные последовательности (рис. 4). Обозначения и цвета те же, что и в предыдущем примере.

Пример 3. Рассмотрим Т-боксовую регуляцию гена *ileS* в актинобактериях. Исходные сайты в листьях взяты из работы [27]. Использовано стандартное дерево видов (рис. 5). Сокращения: An – *Actinomyces naeslundii*, Cd – *Corynebacterium diphtheriae,* Ce – *Corynebacterium eficiens*, Cg – *Corynebacterium glutamicum*, Ma – *Mycobacterium avium*, Ml – *Mycobacterium leprae*, Mm – *Mycobacterium marinum*, Mt – *Mycobacterium tuberculosis*, Nf – *Nocardia farcinica*, Pa – *Propionibacterium acnes*, Rx – *Rubrobacter xylanophilus*, Sa – S*treptomyces avermitilis*, Sc – *Streptomyces coelicolor*, Tf – *Thermobifida fusca.*

Алгоритм выдает итоговое множественное выравнивание нуклеотидных последовательностей – предполагаемые регуляторные сайты и измененные нуклеотидные последовательности (рис. 6). Антисеквестр выделен темно-серым цветом, секвестр – подчеркиванием. В вершине 5 наблюдаются два варианта регуляции: в альтернативном варианте антисеквестр показан курсивными буквами, а секвестр – светло-серым фоном. Аналогичная ситуация наблюдается в вершине 21.

Пример 4. Рассмотрим еще один, третий, тип регуляции: RFN-регуляцию экспрессии генов биосинтеза и транспорта рибофлавина у эубактерий (ген *ribB* у BP, EC, PP, YP и ген *ribD* у остальных видов, см обозначения ниже). Исходные сайты регуляторного RFN-элемента в листьях взяты из работы [37]. Берем стандартное дерево видов (рис. 7). Использованы сокращения: BQ – *Bacillus anthracis*, BH – *Bacillus halodurans*, BS – *Bacillus subtilis*, BP – *Burkholderia pseudomallei*, CA – *Clostridium acetobutylicum*, DF – *Clostridium difficile*, DR – *Deinococcus radiodurans*, EC – *Escherichia coli*, LL – *Lactococcus lactis*, PP – *Pseudomonas putida*, SA – *Staphylococcus aureus*, TM – *Thermotoga maritime*, YP – *Yersinia pestis*.

RFN-структура представлена как структура, состоящая из спирали-черенка и четырех спиралей в его петле, пронумерованных по часовой стрелке (спирали 1, 2, 3 и 4). Наш алгоритм выдает множественное выравнивание, показанное на рис. 8. На нем выделена искомая RFN-структура: двойным подчеркиванием выделен черенок, светло-серым разреженным фоном – первая и третья спирали, светло-серым фоном – вторая и четвертая спирали, темно-серым или черным фоном – вариабельные спирали. Две спирали, первая и вторая, могут заменяться на одну спираль; две другие спирали, третья

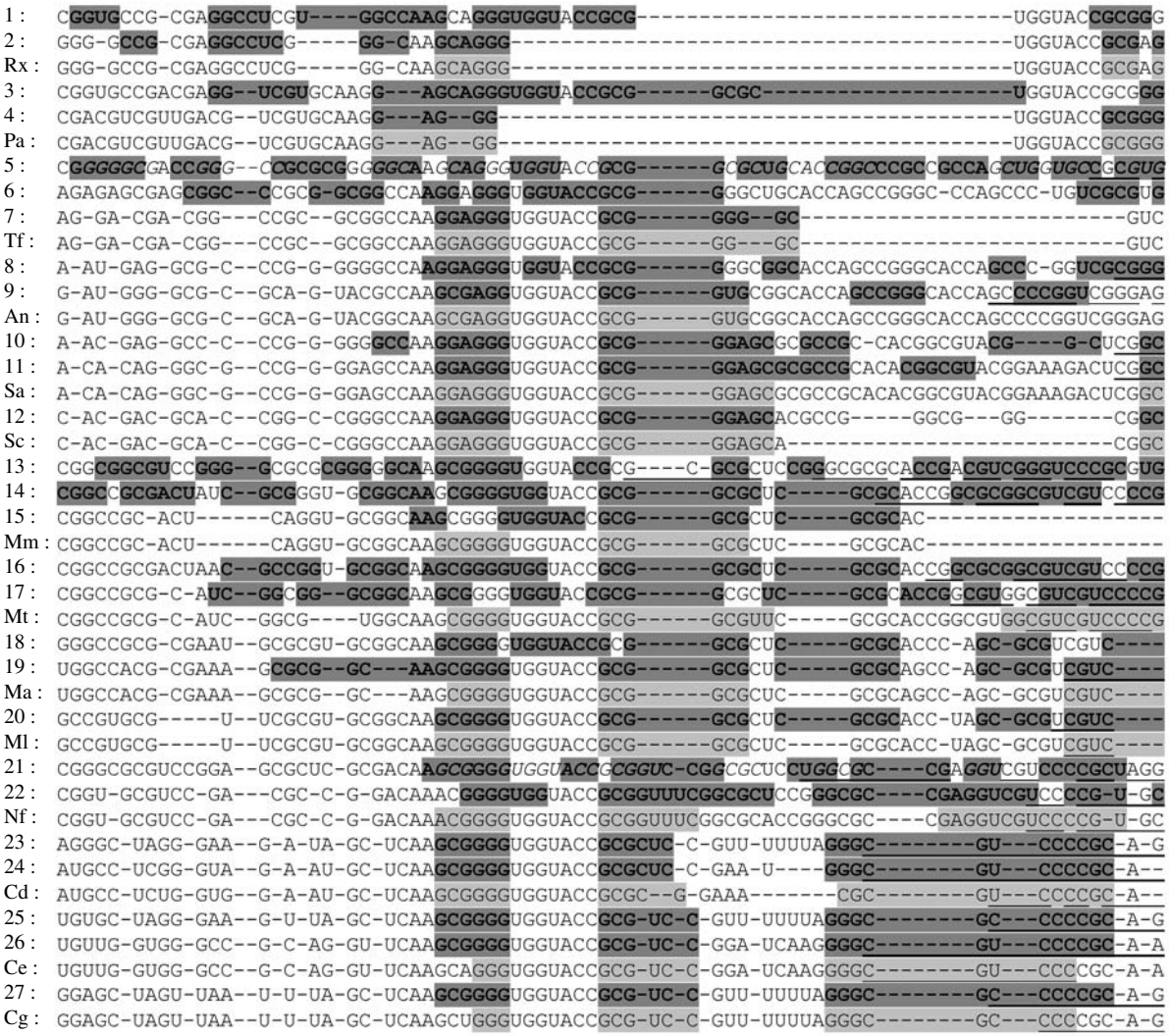

Рис. 6. Полученное алгоритмом множественное выравнивание (с вторичной структурой) предполагаемых регуляторных сайтов для T-боксовой регуляции гена ileS у актинобактерий.

и четвертая, также могут замениться на одну спираль, эти альтернативные спирали показаны подчеркиванием. Отметим, что консервативные нуклеотиды, характерные для RFN-структуры, в основном выровнялись по столбцам. В предке 19 четвертая спираль имеет альтернативу, показанную курсивными буквами, которая продолжается в потомках.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Цель настоящей работы – представление нового метода эволюционной реконструкции и множественного выравнивания родственных последовательностей РНК с учетом их предполагаемой общей и совместно эволюционирующей вторичной структуры, а также получение предварительных результатов тестирования метода. В основе метода лежат предположения о том, что: 1) исходные последовательности в листьях имеют общую совместно эволюционировавшую вторичную структуру и 2) известно филогенетическое дерево, в листьях которого заданы эти последовательности без всякого упоминания или использования самой вторичной структуры в них (она не предполагается известной).

В случае приведенных здесь и также для других примеров регуляций наш алгоритм строит разумную вторичную структуру в предковых вершинах. Она близка к той регуляторной структуре, которая предсказывается биоинформатическими и экспериментальными исследованиями. Вторичные структуры, полученные алгоритмом на исходных последо-

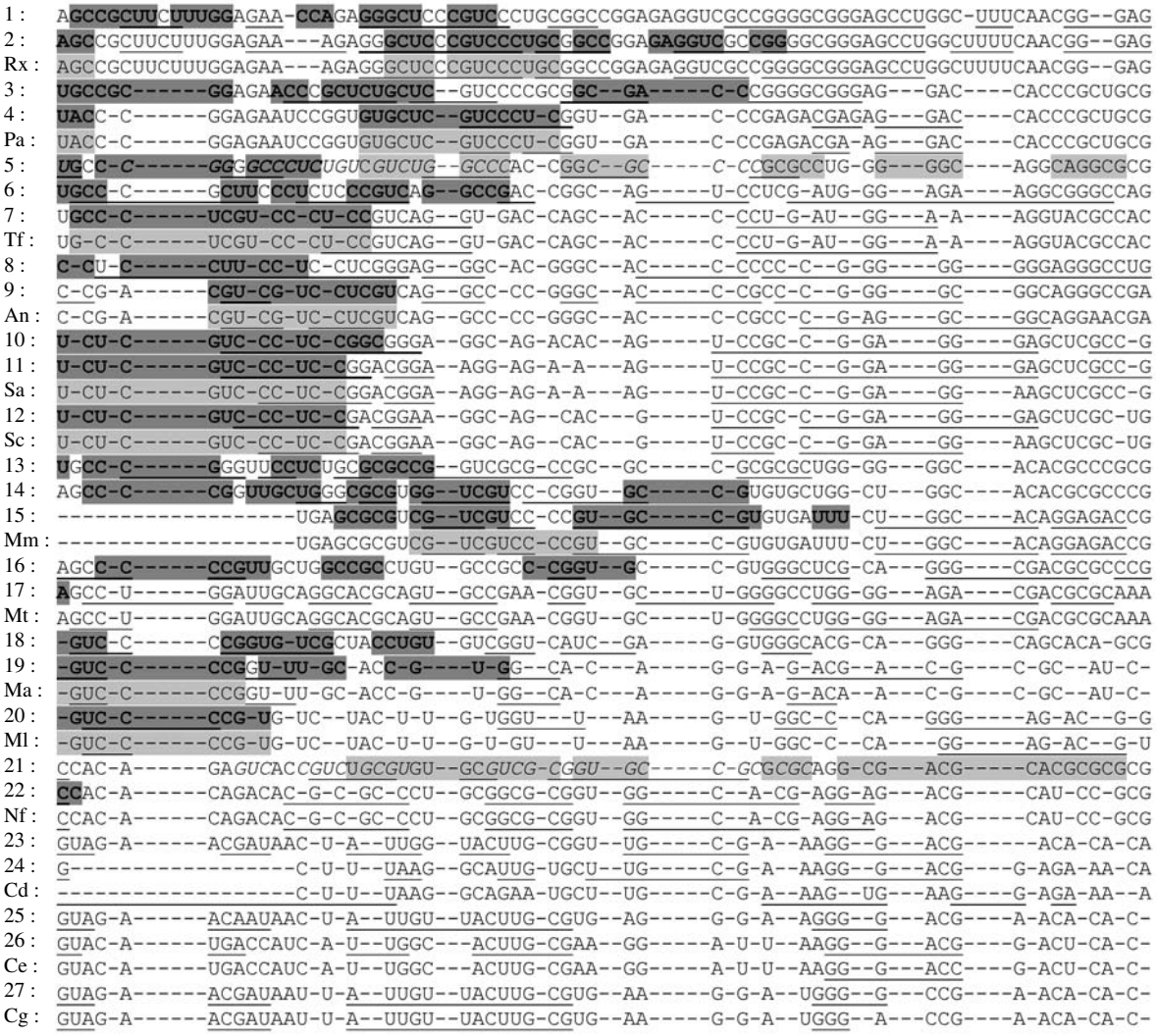

Рис. 6. Окончание.

вательностях в листьях, также практически совпадают с известными. Множественное выравнивание первичных структур в листьях и даже по всему дереву имеет хорошее качество. Мы тестировали модель, добавляя шум как в искусственные, так и в биологические примеры, и наблюдали устойчивую работу алгоритма.

Первый подход, упомянутый в разделе "Постановка задачи" и подробно описанный в работах [32–34, 38], выдал на исходных данных из примеров 1–4 те же или очень похожие вторичные структуры, несмотря на различие этих подходов. Мы тестировали полученные предковые сигналы, используя модель классической аттенюаторной регуляции из работ [26, 35] и программные средства, представленные на сайте http://lab6.iitp.ru, пункт 3. Этот тест подтвердил функциональность предковых сигналов для данной регуляции.

Сравнение с другими методами и программами. Чтобы сравнить результаты нашего алгоритма с результатами других стандартных алгоритмов, мы применяли к тем же данным известные программы такие, как PAML, PAUP и другие, см интернет-сайт http://evolution. genetics.washington.edu / phylip / software.serv.html. Если на вход подавались лишь исходные первичные структуры (без выравнивания), то никакая из этих программ не могла реконструировать предковые регуляторные элементы того типа, который был представлен в листьях. Если на вход подавалось и выравнивание, выполненное с учетом вторичной структуры (например, указанное в работе [22] для классической аттенюаторной регуляции), то результат зависел от используемой про-

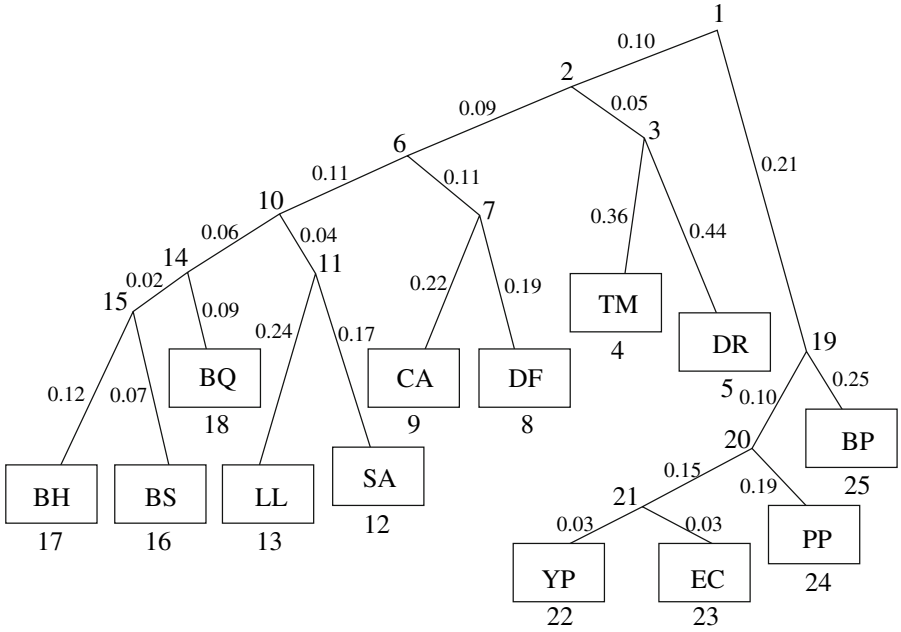

Рис. 7. Филогенетическое дерево эубактерий.

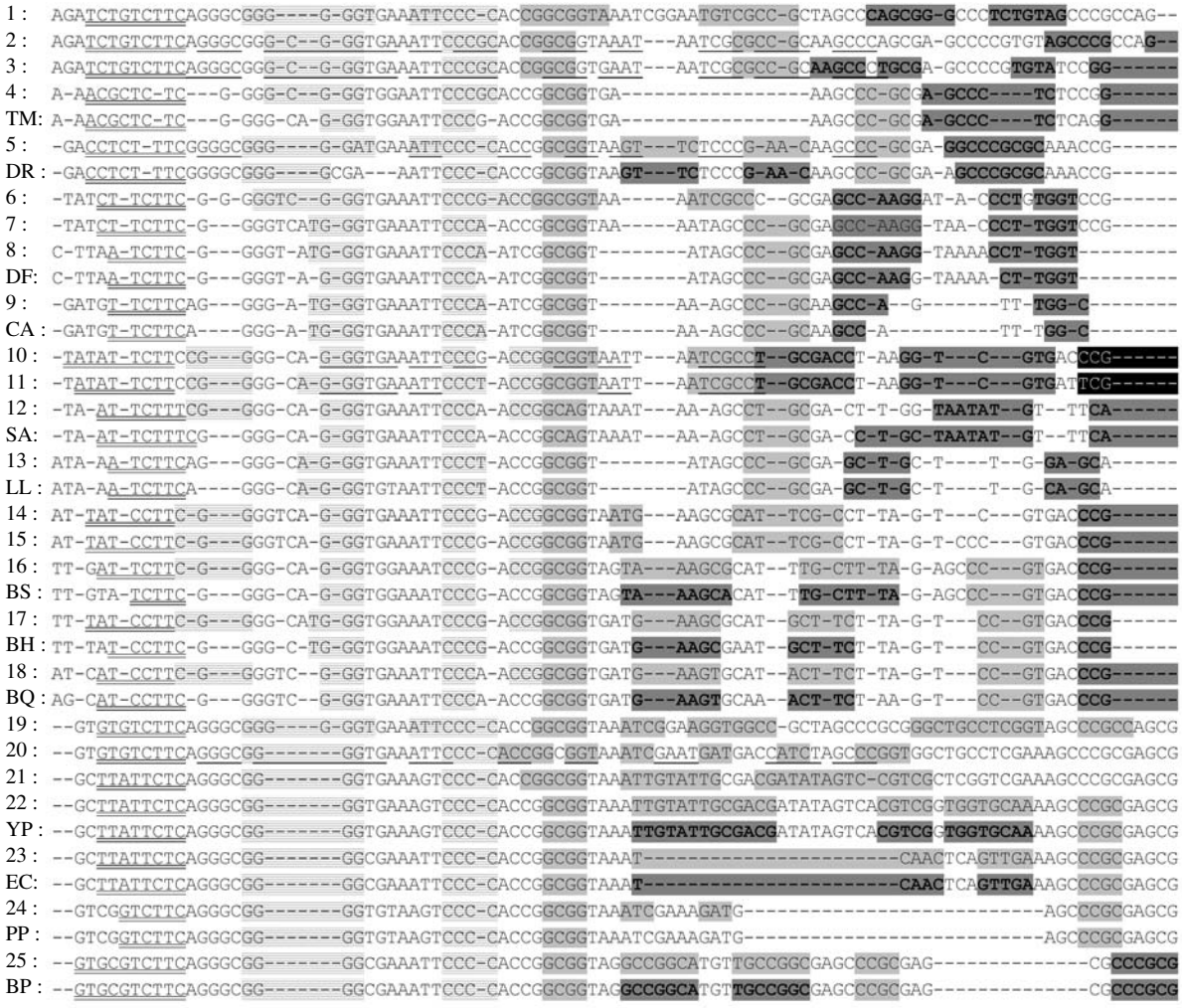

Рис. 8. Полученное алгоритмом множественное выравнивание (с вторичной структурой) предполагаемых регуляторных сайтов для RFN-регуляции экспрессии генов биосинтеза и транспорта рибофлавина у эубактерий.

Рис. 8. Окончание.

граммы. PAML не смогла предсказать вторичную структуру требуемого типа в предковых последовательностях. PAUP смогла предсказать такую структуру, однако эта структура не была консервативной вдоль ребер дерева.

Авторы благодарны рецензенту, глубокие замечания которого позволили значительно улучшить текст статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Nei M., Kumar S. 2000. *Molecular Evolution and Phylogenetics*. Oxford University Press US.
2. *Gascuel O., Steel M.* 2007. *Reconstructing Evolution: New Mathematical and Computational Advances*. Oxford University Press.
3. Page R.D.M., Holmes E.C. 1998. *Molecular Evolution: A Phylogenetic Approach*. 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. (Editor). 2004. *Mathematics of Evolution and Phylogeny*. Oxford University Press.
7. Felsenstein J. 2004. *Inferring phylogenies*. Sinauer Associates.
8. Nakhleh L., Warnov T., Linder C.R. 2004. Reconstructing reticulate evolution in species: theory and practice. In: *Proc* 8*th 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. Phyl. 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. Phylogenet. 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., **4751** of LNCS. 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: *Proceedings of the fourth 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: *Proceedings of the eighth Annual International Conference on Research in Computational Molecular Biology, RECOMB*. Eds Bourne P.E., Gusfield D. ACM, pp. 326–335.
19. Bonizzoni P., Vedova G. Della, 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. 2008. *RNA.* **14**, 717–735.
28. McAdams H.H., Srinivasan B., Arkin A.P. 2004. The evolution of genetic regulatory systems in bacteria. *Nat. 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. *Internet-site* http://www.itee.uq.edu.au/nic/papers/csss-rna.pdf.
32. Любецкий В., Жижина Е., Рубанов Л. 2008. Гиббсовский подход к проблеме эволюции биологических последовательностей. *Проблемы передачи информации РАН*. В печати.
33. Gorbunov K.Yu., Lyubetsky V.A. 2007. Modeling evolution of the nucleotide sequence with secondary structure. In: *Proceedings of 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: *Proceedings of 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*, **4545** of LNCS. Springer-Verlag, pp. 81–94.
36. Горбунов К.Ю., Миронов А.А., Любецкий В.А. 2003. Поиск консервативных вторичных структур РНК. *Молекуляр. биология.* **37**, 850–860.
37. Vitreschak A.G., Rodionov D.A., Mironov A.A., Gelfand M.S. 2002. Regulation of riboflavin biosynthesis and transport genes in bacteria by transcriptional and translational attenuation. *Nucleic Acids Res.* **30**, 3141–3151.
38. Горбунов К.Ю., Любецкий В.А. 2007. Реконструкция предковых регуляторных сигналов вдоль дерева эволюции фактора транскрипции. *Молекуляр. биология.* **41**, 918–925.