МОЛЕКУЛЯРНАЯ БИОЛОГИЯ, 2012, том 46, № 1, с. 176–183

БИОИНФОРМАТИКА

УДК 575.852

БЫСТРЫЙ АЛГОРИТМ ПОСТРОЕНИЯ СУПЕРДЕРЕВА ВИДОВ ПО НАБОРУ БЕЛКОВЫХ ДЕРЕВЬЕВ

© 2012 г. К. Ю. Горбунов, В. А. Любецкий

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

Поступила в редакцию 27.01.2010 г.
Поступила после доработок 29.07.2011 г.
Принята к печати 16.08.2011 г.

Рассматривается задача построения супердерева видов по данному набору деревьев белков, генов, регуляторных сайтов и т.п. Доказано, что в рамках традиционной постановки эта задача NP-трудная. Нами предложена новая постановка задачи: ищется супердерево, большинство клад которого представлены среди клад исходных деревьев белков. В такой постановке задача кажется биологически естественной и допускает быстрый алгоритм ее решения. Предложенный алгоритм тестировали на искусственных и биологических наборах деревьев белков, и он показал свою эффективность даже при допущении горизонтальных переносов генов. Если горизонтальные переносы не допускаются, то математически доказывается корректность алгоритма и оценивается время его работы, которая в худшем случае имеет порядок $n^3 \cdot |V_0|^3$, где n – число деревьев генов, а $|V_0|$ – число видов в них. Наша программа построения супердерева, вместе с примерами вычислений и инструкцией для пользователя, свободно доступна на сайте http://lab6.iitp.ru/ru/super3gl/. В этой работе, а также в представленном варианте программы не рассматриваются события горизонтального переноса. Общий случай приведен в статье авторов (журнал "Проблемы передачи информации", 2011).

***Ключевые слова*: дерево видов, супердерево видов, новая постановка задачи построения супердерева, быстрый алгоритм построения супердерева, порождение набора генов по дереву видов, моделирование эволюции гена вдоль дерева видов.**

A FAST ALGORITHM TO BUILD A SUPERTREE WITH A SET OF GENE TREES, by *K. Y. Gorbunov*, V. A. Lyubetsky* (Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, 127994 Russia; *e-mail: gorbunov@iitp.ru). Important desired properties of an algorithm to construct a supertree (species tree) by reconciling input trees are its low complexity and applicability to large biological data. In its common statement the problem is proved to be NP-hard, i.e. to have an exponential complexity in practice. We propose a reformulation of the supertree building problem that allows a computationally effective solution. We introduce a biologically natural requirement that the supertree is sought for such that it does not contain clades incompatible with those existing in the input trees. The algorithm was tested with simulated and biological trees and was shown to possess an almost square complexity even if horizontal transfers are allowed. If HGTs are not assumed, the algorithm is mathematically correct and possesses the longest running time of $n^3 \cdot |V_0|^3$, where n is the number of input trees and $|V_0|$ is the total number of species. The authors are unaware of analogous solutions in published evidence. The corresponding inferring program, its usage examples and manual are freely available at http://lab6.iitp.ru/en/super3gl. The available program does not implement HGTs. The generalized case is described in the publication "A tree nearest in average to a set of trees" (Information Transmission Problems, 2011).

***Keywords*: species tree, fast algorithm to build supertrees, reformulation of the supertree building problem, modeling gene evolution along a species tree, modeling gene evolution along a species tree.**

ВВЕДЕНИЕ

Задача построения супердерева видов по набору деревьев белков (обычно комплексов ортологичных групп белков, КОГов) имеет давнюю историю, ее фундаментальная и практическая важность общепризнана. Эта задача относится к числу NP-трудных [1]. В списке литературы приведены работы, которые содержат многочисленные ссылки по этой теме, а обзор результатов по этой и родственным задачам приведен в книге [2], см. также [3], содержащую ссылки на публикации последнего времени. С практической точки зрения это означает, что время ее решения экспонен-

* Эл. почта: gorbunov@iitp.ru, lyubetsk@iitp.ru

циально зависит от объема исходных данных – числа деревьев генов и числа видов в них. Поэтому вычислительно эффективный (точнее сказать, полиномиальной сложности с низкой степенью полинома) алгоритм, который выдает ее точное решение, может быть найден только в случае некоего изменения в ее постановке. Важно, чтобы такое изменение было приемлемо с точки зрения биологических приложений. В литературе не приведено каких-либо результатов, способствующих постановке и решению этой задачи. Отметим, что опубликовано достаточно большое число экспоненциальной сложности эвристических алгоритмов построения супердерева, некоторые из них обсуждаются в нашем обзоре [4].

Мы предлагаем новую постановку задачи построения супердерева и алгоритмы ее решения для двух случаев, когда горизонтальные переносы генов не учитываются и когда они учитываются. Здесь анализируется только первый из этих случаев, второй из них рассмотрен в других наших работах [5–7]. Время работы алгоритма при худших исходных данных – порядка третьей степени от числа n исходных деревьев генов и от числа $|V_0|$ видов в них. В большинстве случаев время работы алгоритма много меньше: как правило, оно квадратично зависит от этих двух характеристик данных (см. раздел "Тестирование"). Это наблюдение остается верным и в более общих случаях, которые обсуждались нами ранее [5–8].

Итак, рассматривается задача реконструкции дерева S, содержащего $|V_0|$ видов, по заданному набору деревьев G_i ряда генов, где i меняется от 1 до n. Как и во многих работах (например, [4, 9–12]), искомое дерево видов S строится как супердерево для набора G_i; иными словами, как дерево "в среднем наиболее близкое" к каждому G_i. Такой подход требует уточнения понятия близости между данными деревом генов G и деревом видов S. Используемое понятие близости традиционно основано на вложении дерева генов G в дерево видов S с той разницей, что вместо приведенного Гвиго (Guigo) и соавт. [12] способа вложения используется способ, предложенный в [5].

Настоящая публикация является непосредственным продолжением работы авторов [5].

ОСНОВНЫЕ ПОНЯТИЯ

Рассмотрим множество видов V_0; для каждого вида s из V_0 будем считать заданным непустое множество генов $G(s)$. Объединение множеств $G(s)$ будем считать разбитым на кластеры; неформально говоря, в кластер входят гомологичные гены. В кластер может входить любое число генов, принадлежащих одному виду.

Деревом видов называется бинарное корневое дерево, листьям которого приписаны имена видов; при этом множество рассматриваемых видов V_0 и множество листьев дерева видов находятся во взаимно однозначном соответствии. Деревом генов, соответствующим кластеру генов K, называется бинарное корневое дерево, каждому листу которого приписаны имя гена g из K, соответствие между листьями дерева генов и генами из кластера K – взаимно однозначное. Для удобства будем считать, что вместе с геном g, который приписан некоторому листу, этому же листу приписан вид s, из которого взят g; будем говорить, что такие ген g и вид s находятся в отношении "ген–вид".

Отметим, что, говоря неформально, листья дерева генов, которым приписаны пары типа $\langle g_1,s\rangle$, $\langle g_2,s\rangle$, $\langle g_3,s\rangle$, ... соответствуют паралогам g_1, g_2, ... в виде s. Гены, приписанные листьям, берутся из некоторого фиксированного семейства гомологичных генов, в основном, – из комплекса ортологичных групп белков, представленных в базах данных GenBank и NCBI. В этом смысле каждое дерево генов определяет некоторый ген в его эволюционном развитии.

Условимся, что у всех деревьев корень располагается "сверху". Обозначим e^- и e^+ верхний и нижний концы ребра e. Ребро понимается как пара вершин: начало e^- и конец e^+. Ребро, ведущее в вершину g, обозначим b_g. В упомянутой работе [5] каждое дерево рассматривается вместе с его "корневым ребром" – специально добавленным ребром, которое идет от корня вверх и соответствует времени, в котором жил общий предок всех представленных в дереве видов или генов; верхний конец корневого ребра назван "суперкорнем". Ребра дерева видов S названы *трубами*, в частности, корневое ребро называется *корневой трубой* [5].

Пусть G – дерево генов. На вершинах дерева G определим отношение порядка "ниже": $g_1 < g_2$, если $g_1 \neq g_2$, и в g_1 можно провести путь из суперкорня через g_2. На множестве всех вершин и труб в S определим единое отношение порядка $y < x$ таким образом: вершина или труба y "ниже" некоторой вершины или трубы x в S, если $y \neq x$ и в трубе y можно провести путь из суперкорня через x; соответственно "x выше y"; обозначим $y \leq x$, если $y < x$ или $y = x$.

Дадим определения понятий, используемых в этой статье. Для их понимания необходимо ознакомиться с разделом "Постановка задачи" в работе Горбунова и Любецкого [5]. В отличие от этой работы, в настоящей работе мы рассматриваем только вложения и сценарии без горизонтальных переносов, поэтому называем их просто вложениями и сценариями. Случай горизонтальных переносов рассмотрен в упомянутых публикациях [5–7].

Вложением дерева генов G в дерево видов S называется отображение f всех вершин $V(G)$ дерева G

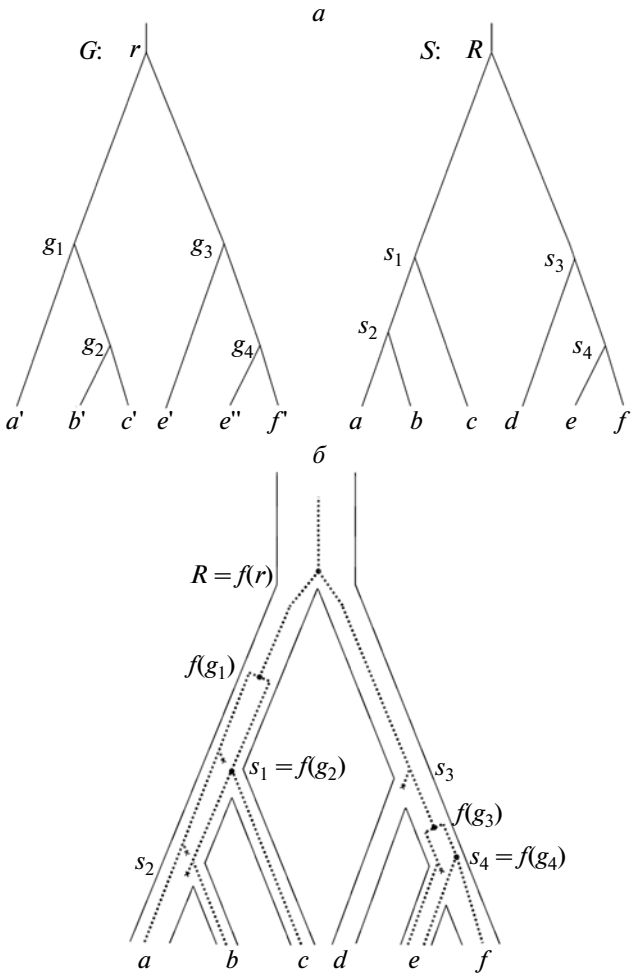

Рис. 1. Иллюстрация понятий дупликации, потери гена и видообразования. *а* – Пример дерева генов G и дерева видов S, у которых в листьях: ген a' взят из вида a и т.д., паралоги e' и e'' взяты из вида e. Вид d не представлен в дереве G. *б* – Наглядно показаны значения вложения f дерева G в дерево S, значения на листьях в G совпадают с соответствующими листьями в S. Значения отображения f на внутренних вершинах дерева G показаны жирными точками. Значение $f(g_1)$ показано в трубе (хотя формально оно равно этой трубе) и, по определению дупликации, вершина g_1 соответствует событию дупликации, то же самое – для вершины g_3. Значения f на всех других внутренних вершинах дерева G совпадают с соответствующими внутренними вершинами дерева видов и, по определению видообразования, соответствуют событиям видообразования. Для ребра $h = (g_1, a')$ вершины s_1 и s_2 лежат между значениями на его концах и, по определению потери пары (h, s_1) и (h, s_2), соответствуют событиям потери; на рисунке они показаны отростками с крестиком на конце. Аналогично, потерями являются пары $((g_2, b'), s_2)$, $((g_3, e'), s_4)$, $((r, g_3), s_3)$.

в вершины $V(S)$ и трубы $E(S)$ дерева S, для которого выполнены следующие условия.

1) Суперкорень в G отображается в корневой трубе в S; каждый лист g в G отображается в листе s в S согласно отношению ген–вид.

2) Если g_1 – сын g и $f(g)$ – вершина, то $f(g_1) < f(g)$, а если $f(g)$ – труба, то $f(g_1) \le f(g)$.

3) Пусть g_1 и g_2 – сыновья вершины g: если $f(g)$ – вершина, то $f(g_1)$ и $f(g_2)$ лежат в разных поддеревьях с корнями в сыновьях вершины $f(g)$.

Говоря неформально, условие (3) означает следующее: вид, которому принадлежит ген g, – последний общий предок видов, которым принадлежат виды g_1 и g_2.

Для данного вложения f напомним на формальном уровне следующие определения эволюционных событий [5]. *Дупликация* гена – несуперкорневая вершина g в G, для которой $f(g)$ – труба из S. *Потеря* гена – пара $\langle e, s\rangle$, для которой e – ребро в G, s – вершина в S и $f(e^+) < s < f(e^-)$. *Видообразование* (в отношении рассматриваемого гена) – вершина g из G, для которой $f(g)$ – вершина в S и обе вершины g и $f(g)$ не являются листьями. Рассматриваются только видообразования, которым соответствуют развилки в дереве G; поскольку их цена ниже полагается нулевой, то видообразование, как отдельное событие, по существу не рассматривается.

Пример вложения дерева генов в дерево видов приведен на рис. 1*а*, *б*.

Поясним неформально введенные определения. Внутренние вершины дерева видов соответствуют (гипотетическим) предковым видам; внутренние вершины дерева генов соответствуют (гипотетическим) предковым генам. Вложение дерева генов в дерево видов (в идеале) показывает, какому предковому виду принадлежит данный предковый ген (для листьев, т.е. современных видов и генов, это выполнено по определению).

Отметим, что в дереве видов вершинами обозначены не все предковые виды, а только те, с которыми связано разветвление на два вида (“видообразование”). Между событиями видообразования виды могли развиваться, некоторые гены могли удвоиться (“дупликация”). В таком случае (вид эволюционирует, но не раздваивается) новый вид будет (неявно) соответствовать некоторой внутренней точке на ребре (“трубе”) дерева видов. Вершина в дереве генов, соответствующая дупликации гена, таким образом, отображается в трубу, а не в вершину.

Аналогично и в дереве генов, гипотетическому предковому гену g соответствует вершина в дереве генов, только если из гена g эволюционно произошли два новых гена – с видообразованием (такому гену соответствует вершина в дереве видов) или без него (см. выше). Если же при видообразовании (чему соответствует некоторая вершина s дерева видов) в одном из образовавшихся видов аналог гена g отсутствует, то при “раздвоении” видов не происходит “раздвоения генов”. Поэтому в этом случае (“потеря гена”) уже вершина де-

рева видов s не соответствует никакой вершине дерева генов. Можно сказать, что s соответствует точке на некотором ребре e дерева генов. Очевидно, что при этом выполнено условие $f(e^+) < s < f(e^-)$. Отметим следующее. Во-первых, одному ребру e дерева генов G может соответствовать несколько потерь генов, т. е. несколько вершин s дерева видов S таких, что выполняется условие $f(e^+) < s < f(e^-)$. Аналогично, одной вершине s дерева видов может соответствовать несколько потерь генов в дереве генов, т.е. несколько ребер e, для которых выполнено условие $f(e^+) < s < f(e^-)$. Это связано с дупликациями генов (см. пример на рис. 2*а*, *б*).

Для любых деревьев генов G и видов S и любого вложения f *обозначим*: $l(f, G, S)$ – число потерь, $d(f, G, S)$ – число дупликаций генов при вложении f. *Обозначим* c_l – цену одной потери гена, c_d – цену одной дупликации. Цену за одно видообразование мы полагаем равной нулю; однако если эта цена меньше, чем $c_d + 2c_l$, то алгоритм и приведенные ниже утверждения сохраняются.

ПОСТАНОВКА ЗАДАЧИ

Дан набор $\{G_i\}$, состоящий из n укорененных бинарных деревьев генов. Дерево видов S назовем *согласованным* с набором $\{G_i\}$, если множество листьев в S совпадает с множеством V_0 – всех видов, представленных в листьях всех деревьев G_i. Нашей целью будет формализация задачи построения дерева видов S "наиболее близкого" в целом к набору деревьев $\{G_i\}$. Начнем со вспомогательной задачи.

Сценарий эволюции гена вдоль дерева видов. Супердерево. Пусть дан набор деревьев генов $\{G_i\}$ и согласованное с ним дерево видов S. *Вложением* f набора $\{G_i\}$ в дерево S назовем набор вложений $\{f_i\}$, где каждое f_i является вложением G_i в S.

Задача А1. Дан набор $\{G_i\}$ и дерево видов S. Требуется найти вложение f для набора $\{G_i\}$ и дерева S, на котором достигает минимума функционал

$$c(\{G_i\}, f, S) = \Sigma_i(c_l \cdot l(f_i, G_i, S) + c_d \cdot d(f_i, G_i, S)). \quad (1)$$

Отметим, что $c_l \cdot \Sigma_i\, l(f_i, G_i, S)$ – суммарная цена за все потери во всех G_i, а $c_d \cdot \Sigma_i\, d(f_i, G_i, S)$ – суммарная цена за все дупликации во всех G_i.

Вложение, на котором достигается минимум в (1), назовем *сценарием*.

Нами доказано, что при фиксированных $\{G_i\}$, S и коэффициентах c_l, c_d сценарий единствен и даже не зависит от выбора любых неотрицательных значений этих коэффициентов.

Вершины g в дереве генов и s в дереве видов назовем *согласованными*, если они не суперкорни, и выполняется одно из следующих условий:

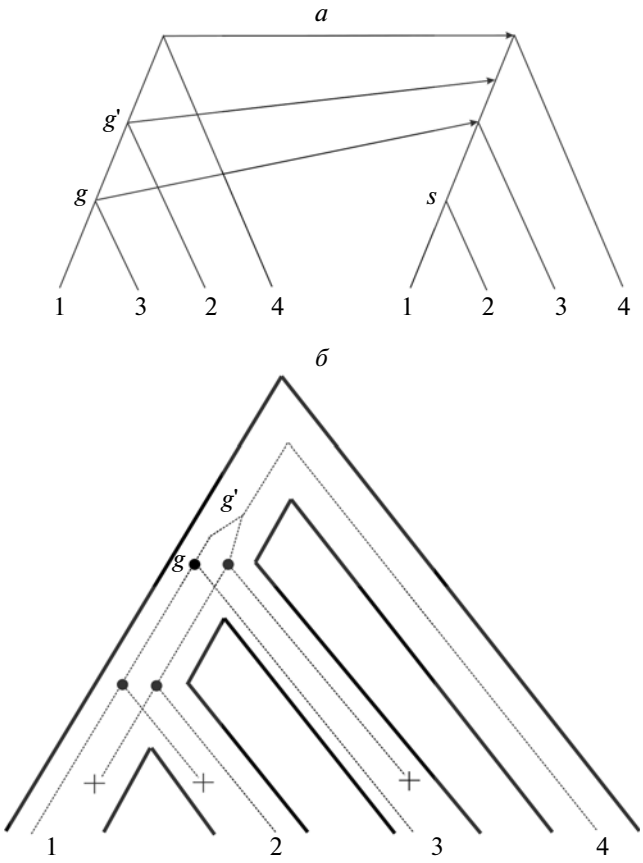

Рис. 2. *а* – Вложение дерева генов (слева) в дерево видов (справа). В примере четыре вида, они обозначены цифрами от 1 до 4. В листьях дерева генов каждому виду соответствует один ген, поэтому в его листьях показаны только номера видов. Эволюционные события: дупликация в вершине g' и три потери, которые соответствуют парам ребро–вершина $(2, g')$ и s, $(2, g')$ и s', $(1, g)$ и s, где s' – отец s. *б* – То же вложение, показанное путем "вписывания" дерева генов в трубы дерева видов. Вложение имеет одну дупликацию в вершине g' и три потери, показанные крестиками. Точки указывают на дивергенцию гена в связи с видообразованием.

1) g и s – листья, находящиеся в отношении ген–вид; или

2) g имеет сыновей g_1, g_2 и s имеет сыновей s_1, s_2 и

$$[(M_{g1} \subseteq M_{s1} \text{ и } M_{g2} \subseteq M_{s2}) \text{ или } (M_{g1} \subseteq M_{s2} \text{ и } M_{g2} \subseteq M_{s1})].$$

Запись $X \subseteq Y$ означает, что множество X является частью (подмножеством) множества Y. Напомним, что b_s обозначает трубу в дереве S с концом в вершине s.

Лемма 1. Пусть G – дерево генов, S – дерево видов. Тогда для любого сценария h, соответствующего G и S, и любых фиксированных неотрицательных значений цен за одну дупликацию и одну потерю, выполняется следующее. Пусть g – любая вершина в G, отличная от суперкорня, а s – послед-

ний общий предок множества M_g в дереве S. Тогда $h(g) = s$, если вершины g и s согласованные, и $h(g) = b_s$, если эти вершины не согласованные.

Доказательство приведено ранее [7].

Вложение, описанное в лемме 1, назовем *сценарием* эволюции генов, представленных деревьями G_i, вдоль дерева видов S и обозначим $h(\{G_i\}, S) = \{h_i\}$. Значение $c(\{G_i\}, S) = c(\{G_i\}, h(\{G_i\}, S), S)$ будем называть *ценой* сценария.

Супердеревом S^* для набора деревьев генов $\{G_i\}$ будем называть дерево видов, для которого величина

$$c(\{G_i\}, S) = c(\{G_i\}, h(\{G_i\}, S), S) \qquad (1^*)$$

принимает минимально возможное значение.

Задача А2. По данному набору деревьев генов построить супердерево.

Приведенную постановку задачи построения дерева видов, основанную на минимизации функционала (1*) можно назвать традиционной. Различные варианты алгоритма построения супердерева приведены ранее [4, 9–12]. Все подобные алгоритмы решают эту задачу за экспоненциальное время; при этом алгоритмы находят лишь эвристические приближения к дереву S^*. Само дерево S^*, как правило, неизвестно, кроме случаев искусственно подобранных данных.

Новая постановка задачи. Кладой M_v в дереве видов S назовем множество видов, приписанных листьям, которые расположены ниже некоторой вершины v в дереве S, саму вершину v назовем корнем клады. Само это множество листьев назовем "множеством листьев клады". Аналогично, *кладой M_g в дереве генов G* назовем множество видов, приписанных листьям, расположенным ниже некоторой вершины g в G; эту вершину назовем корнем клады.

Предлагаем рассмотреть следующую задачу.

Задача Б. По данному набору деревьев генов $\{G_i\}$ найти дерево видов S^* такое, что (i) все клады S^* принадлежат некоторому заранее фиксированному набору множеств P; таким образом, набор P является *параметром* задачи; (ii) значение функционала (1*) для супердерева S^* не превосходит значения этого функционала для других деревьев видов, удовлетворяющих условию (i).

Множество клад искомого дерева видов заведомо включает множество всех видов V_0 и все его одноэлементные множества, и не включает пустое множество. Поэтому будем рассматривать, *только такие* наборы P, которые содержат указанные множества. Алгоритм, излагаемый ниже, применим к любому набору P, но типичный пример такого набора – набор P_0 всех клад во всех исходных деревьях генов, пополненный множеством всех видов V_0. Этот набор будем называть *стандартным* для заданного множества деревьев генов. Обозначим $|X|$ – число элементов в множестве X. Число элементов в стандартном наборе P_0 оценивается сверху: $|P_0| \le 2|V_1|$, где V_1 – множество всех листьев во всех деревьях генов. При естественном предположении, что *среднее число листьев в деревьях генов порядка* $|V_0|$ получаем: $|P_0| \le K|V_0| \cdot n$, где K – некоторая константа, в наших данных не превышающая 2; напомним, что n – число исходных деревьев генов, а V_0 – множество всех видов в них. Если набор P не включает все множества из стандартного P_0, то его можно расширить множествами из P_0. Поэтому *предполагаем*, что рассматриваемые ниже наборы P включают все множества из стандартного набора P_0.

Далее будем считать фиксированным набор деревьев генов $\{G_i\}$ и набор множеств P. В нашем подходе существенную роль играют следующие два определения. Первое определение. Пусть e – ребро в дереве генов G. Определим M_e как множество видов, приписанных всем листьям ниже ребра e в дереве G. Множество M_d для трубы d дерева видов S определяется аналогично. Для дерева генов G и множества видов M *определим* множество $Ed(M, G)$ ребер e в G, в которых $M_e \subseteq M$ и не существует ребра $e' > e$ с этим свойством. Таких ребер e может быть несколько, но все они несравнимы в G.

Пусть f – вложение дерева генов G в дерево видов S. Тогда утверждение "ребро e из G *входит в трубу* d из S" означает, что $f(e^+) \le d < f(e^-)$ ("входит" в геометрическом смысле).

Лемма 2. Для сценария $h(G, S)$ из Леммы 1 выполняется:

a) в трубу b в S входят в точности ребра из $Ed(M_b, G)$;

b) если труба b_1 – сын трубы b, то для любого ребра e в G, входящего в b_1, существует ровно одно ребро $e' \ge e$, входящее в b.

Доказательство приведено ранее [7].

Второе определение. Множество V из P назовем *базисным*, если его можно разбить на какие-то две части из P, каждую часть в свою очередь можно разбить на две части из P и так далее – до достижения одноэлементных множеств, представляющих виды. Очевидно, задача Б имеет решение тогда и только тогда, когда множество всех видов V_0 – базисное.

КОМПЬЮТЕРНАЯ ПРОГРАММА ПОСТРОЕНИЯ СУПЕРДЕРЕВА

Алгоритм и его обоснование изложены нами ранее [7], сама программа свободно доступна на сайте http://lab6.iitp.ru/ru/super3gl/. Приведем несколько обозначений и пояснений, необходимых при работе с программой.

Эвристическое решение задачи А2 состоит из двух шагов. На первом шаге строятся так называемые “базисные деревья” $S(V)$ для всех базисных множеств V из данного набора P множеств. На втором шаге по набору деревьев $S(V)$ строится аппроксимация S' искомого супердерева S^* для $\{G_i\}$ (см. дополнительные материалы, пункты 1–3 на сайте www.molecbio.com/downloads/2012/1/supp_gorbunov_rus.pdf).

Компьютерная программа основана на следующей теореме.

Теорема 1. Пусть P – набор клад.

a) Если множество V_0 – базисное, то дерево $S(V_0)$ есть решение задачи Б. В противном случае задача Б не имеет решений.

b) Если P – стандартный набор и среднее число листьев в наборе деревьев генов $\{G_i\}$ порядка $|V_0|$, то алгоритм определяет множество $\{S(V)\}$, где переменная V пробегает все базисные множества, за число шагов порядка $|V_0|^2n + |P|^2|V_0| + |P||V_0|n + |P|^3 + |P|^2|V_0|n \le Cn^3|V_0|^3$. За это время алгоритм выдает решение задачи Б или сообщает, что оно не существует. При этом достаточно памяти порядка $n^2 \cdot |V_0|^2$.

Доказательства теоремы 1 приведено нами ранее [7].

Для решения задачи А2 к полученному таким образом набору базисных деревьев нужно применить вспомогательный алгоритм (см. дополнительные материалы, пункт 4).

Компьютерная программа построения набора деревьев $\{S(V)\}$, где V пробегает все базисные множества и дерева S' разработана Л.И. Рубановым и вместе с примерами вычислений и руководством по использованию свободно доступна на сайте http://lab6.iitp.ru/ru/super3gl/.

Программа super3GL способна обрабатывать большие наборы исходных деревьев, включая небинарные, и предназначена в первую очередь для расчетов на мультипроцессорной системе с поддержкой MPI-1.2, но допускает работу и на обычном ПК. Она написана на языке программирования C++ и имеет интерфейс командной строки. Исходный код можно переносить, и после перекомпиляции он может использоваться в среде OS Windows 32/64-bit, Linux, Unix, MacOS. Исполняемые модули программы для Windows 32/64bit (однопроцессорный и параллельный варианты) можно свободно загрузить по ссылкам на указанном сайте, исходный код предоставляется по бесплатной лицензии для некоммерческого использования в научных и учебных организациях. Параллельные модули для Windows рассчитаны на работу в среде MPICH2 (разработчик Argonne National Laboratory) версии 1.3.2 или выше, соответствующий (32/64-bit) вариант которой необходимо установить на используемой мультипроцессорной установке. Поскольку эффективность распараллеливания алгоритма при построении базисных деревьев и супердерева неодинакова и соотношение трудоемкости этих этапов существенно зависит от решаемой задачи, программа позволяет выполнять их как вместе, так и отдельно, в том числе, – на разных компьютерах. Подробные сведения о быстродействии программы на различных вычислительных установках приводятся в руководстве.

Файлы для загрузки приведены в дополнительных материалах (пункт 5): Описание программы (PDF), Однопроцессорный вариант программы (Windows 32bit), Однопроцессорный вариант программы (Windows 64bit), Вариант для MPICH2 v.1.3.2 (Windows 32bit), Вариант для MPICH2 v.1.3.2 (Windows 64bit), Утилита для расшифровки сокращенных наименований в дереве видов, Скрипт для укоренения деревьев.

Ниже приведены результаты тестирования программы на искусственных исходных данных, для которых правильный ответ был заранее получен. В дополнительных материалах (пункт 6) приведены результаты тестирования случаев биологических исходных данных из базы Hodgenom. Дан пример с 276 видами в двух вариантах, Пример с 814 видами и т.д. В этих случаях ответ не был известен, но он хорошо согласуется с известными деревьями видов [11, 13].

ТЕСТИРОВАНИЕ АЛГОРИТМА

Тестирование состояло в том, что по дереву видов S (случайно выбранному или биологическому) сначала строился набор деревьев генов $\{G_i\}$, для которого S наверняка является супердеревом, так как перебором деревьев в окрестности дерева S проверялось, что функционал (1*) имеет в S минимум. Тогда аргументом в пользу нашего или любого другого алгоритма реконструкции дерева видов будет то, что алгоритм полностью или частично восстанавливает S по $\{G_i\}$.

Задача построения такого набора $\{G_i\}$ по данному S сама по себе представляет большой интерес и совершенно нетривиальна. Для ее решения мы использовали такой подход: строили набор $\{G_i\}$, моделируя “реальный” процесс эволюции гена вдоль дерева S, как описано ранее [5]. Удобно рассматривать деревья S, которые включают и вершины с одним сыном. А именно, для каждой трубы дерева S были заданы вероятности $p_d(x)$ и $p_l(x)$ соответственно событий дупликации и потери, которые могли произойти в этой трубе. Пусть уже построена часть G' будущего дерева G_i, начиная с корневого ребра; в G' каждой концевой вершине v приписана труба $x(v)$ из S или пара $\langle i$, лист в $S\rangle$. Во втором случае эта концевая вершина является листом в G_i. Перебираем все концевые вер-

шины v в G', которым приписана труба. Для каждого v разыгрываем событие дупликации с вероятностью $p_d(x(v))$. Если событие произошло, то определяем в x развилку и двум вновь возникшим концевым вершинам приписываем ту же трубу x. Если это событие не произошло, то рассматриваем три случая.

1) Труба x ведет в развилку дерева S. Тогда дважды разыгрываем событие потери, каждый раз с вероятностью $p_l(y)$, где y – любой из сыновей трубы x. Если хотя бы один раз потеря произошла, то разыгрываем с равными вероятностями, в какой из труб, выходящих из этой развилки, произошла эта потеря. Приписываем вершине v трубу, смежную с этой трубой. Если потеря не произошла, то определяем в x развилку и двум вновь возникшим концевым вершинам приписываем каждой своего сына трубы x. 2) Труба x ведет в вершину с одним сыном. Приписываем вершине v трубу, являющуюся сыном трубы x. 3) Труба x ведет в лист дерева S. Приписываем вершине v пару с именем вида в этом листе в S.

Нами выполнен подбор зависимостей $p_d(x)$ и $p_l(x)$ так, чтобы получаемые в процессе эволюции количества разных типов событий и их распределение по трубам были близки к наблюдавшимся при вложениях биологических деревьев КОГов в естественные деревья видов. Соответствующие данные были взяты из опубликованных работ [2, 5, 11] и из наших неопубликованных данных. Конечно, для более адекватного построения таких распределений полезно анализировать более обширные данные и учитывать искажения, которые могут возникать из-за особенностей метода определения эволюционного сценария.

Итак, при всех тестированиях ожидалось, что дерево S', полученное склейкой базисных деревьев, совпадет или будет близко к известному в этих специальных условиях супердереву S, которое, конечно, не было дано алгоритму; оно использовалось только при сравнении S' и S. Алгоритм получал на вход только смоделированный по дереву S набор деревьев генов $\{G_i\}$.

1. Реконструкция искусственного сбалансированного бинарного дерева с 64 листьями. В этом примере S – сбалансированное бинарное дерево видов с 64 листьями. Значения упомянутых выше вероятностей брались следующими: p_d плавно убывало от корневой трубы к листовым трубам от 0.3 до 0.01 (при этом в листьях не возникали многочисленные паралоги); p_l плавно убывало от прикорневых труб к листовым трубам от 0.5 до 0.25 (вблизи корня вероятность гену не оставить дошедших до листьев потомков больше). Для сбалансированных деревьев, у которых длины всех путей из корня в листья одинаковы (в этом пункте и далее в пункте 3), величины p_d и p_l задавали аналитически. А именно: пусть x – труба и $\rho(x)$ – число труб до x (включительно), начиная с корневой трубы. Напомним, что $|V_0|$ – число листьев в искомом дереве S, обозначим $b = (\lg_2|V_0|) + 1)$. Тогда p_d и p_l задавалась линейной функцией, определенной на отрезке [1, b] или [2, b] значениями в его концах, которые указаны выше (второе число – значение в b).

Таким образом порождался набор из 1000 искусственных деревьев генов, у которых число листьев колебалось от 32 до 96, в среднем одно дерево содержало 70 листьев и 39 видов. На одно дерево в среднем приходилось 16 дупликаций и 37 потерь. Порожденный набор деревьев подавался на вход нашей программы. Она *точно* восстанавливала исходное дерево видов. Такое тестирование повторялось 20 раз.

2. Реконструкция естественного дерева бактерий с 40 листьями. Это супердерево S было взято из предыдущей работы [5, рис. 4]. Порождение по нему набора деревьев генов происходило так же, как в предыдущем примере. При этом брались следующие значения вероятностей событий: p_d плавно убывало от корня к листьям от 0.1 до 0.01; p_l плавно убывало от корня к листьям с 0.5 до 0.25. Это дерево по своей топологии ближе к “гребенке”, чем к сбалансированному дереву, так что вероятность дупликации уменьшена по сравнению с примером 1, чтобы число дупликаций было не слишком большим по сравнению с известными эволюционными сценариями. Для несбалансированных деревьев (в этом пункте и, далее, в пункте 4) величины p_d и p_l задавались несколько сложнее. Сначала для указанных в этих пунктах деревьев S строились сбалансированные деревья $+S$ путем разбиения исходных труб новыми вершинами с одним сыном, смысл такого разбиения обсуждали ранее [5] (раздел “Алгоритмы построения внутреннего дерева и временных слоев”, пункт c). Это делалось алгоритмом, описанным в том разделе. Затем для дерева $+S$ вычисляли величины p_d и p_l с помощью указанной выше линейной функции. Заметим, что дерево S', полученное по набору $\{G_i\}$, который был построен по $+S$, сравнивалось именно с деревом S, так как наш алгоритм построения супердерева определяет дерево с точностью до развилок.

Таким образом был порожден набор из 1000 искусственных деревьев генов, у которого число листьев колебалось от 30 до 60 (в среднем, одно дерево содержало 51 лист и 31 вид). На одно дерево, в среднем, приходилось 15 дупликаций и 29 потерь. Порожденный набор деревьев подавался на вход программы. Она точно восстанавливала исходное дерево видов. Это тестирование повторялось 20 раз.

3. Реконструкция искусственного сбалансированного бинарного дерева со 128 листьями. Здесь порождение деревьев генов происходило аналогичным образом по данному супердереву S с теми же значениями вероятностей, что в примере 1. Та-

ким образом были порождены 1000 искусственных деревьев генов, у которых число листьев колебалось от 64 до 192 (в среднем, одно дерево содержало 146 листьев и 77 видов). На одно дерево, в среднем приходилось 33 дупликации и 79 потерь. Порожденный набор деревьев подавали на вход программы. Она выдала дерево, совпадающее с исходным. Это тестирование повторялось 20 раз.

4. Реконструкция естественного дерева видов со 169 листьями. Здесь набор деревьев генов строили так же как, в предыдущих примерах, – вдоль естественного супердерева *S* со 169 листьями, взятого из статьи Пизани (Pisani) и соавт. [11, рис. 1]. Значения вероятностей для моделирования брались такими же, как и в примере 2. Таким образом были порождены 1500 искусственных деревьев генов, у которых число листьев колеблется от 120 до 200 (в среднем, одно дерево содержит 170 листьев и 130 видов). На одно дерево, в среднем, приходится 70 дупликаций и 107 потерь. Программа восстановила исходное 169-листное дерево. Это тестирование повторялось 20 раз.

ЗАКЛЮЧЕНИЕ

Предложена новая постановка задачи построения супердерева видов по набору деревьев белков (генов), которая основана на следующем *предположении*: большинство клад искомого дерева видов представлено кладой хотя бы в одном дереве белков из исходного набора; соответственно супердерево ищется среди деревьев видов, большинство клад которых представлено хотя бы в одном из исходных деревьев белков. Для этой постановки предложен эвристический алгоритм практически квадратичной сложности, который строит супердерево. В случае эволюционных сценариев без горизонтальных переносов доказано, что этот алгоритм точно решает поставленную задачу и имеет в случае худших исходных данных кубическую сложность [7]. Случай горизонтальных переносов рассматривался нами ранее [7]. Эти утверждения следуют из доказанных в этой статье [7] утверждений и тестирования, результаты которого приведены выше. Для тестирования строили набор деревьев белков по тому или иному дереву видов с помощью моделирования процесса эволюции гена вдоль него. По полученному таким образом набору деревьев белков наш алгоритм реконструировал исходное супердерево; реконструкция проходила очень быстро и точно восстанавливала супердерево. Результаты тестирования с использованием биологических данных приведены в дополнительных материалах (пункт 6). Само предположение кажется естественным при рассмотрении многих биологических задач.

СПИСОК ЛИТЕРАТУРЫ

1. Ma B., Li M., Zhang L., et al. 1998. On reconcstructing species trees from gene trees in term of duplications and losses. In *Proc. Second Annu. Internat. Conf. Res. Computat. Mol. Biol.* N.Y., USA: ACM, pp. 182–191.
2. *Phylogenetic supertrees. Combining information to reveal the Tree of Life*. 2004. Edited by Olaf R.P. Bininda-Emonds. Kluwer Academic Publishers, Dordrecht/Boston/London.
3. Bansal M.S., Burleigh J.G., Eulenstein O., Fernández-Baca D. 2010. Robinson-Foulds Supertrees. *Algorithms Mol. Biol.* **5**, 18.
4. Lyubetsky V.A., Gorbunov K.Yu., Rusin L.Y., V'yugin V.V. 2006. Algorithms to reconstruct evolutionary events at molecular level and infer species phylogeny. In: *Bioinformatics of Genome Regulation and Structure II*, Springer Sci. & Business Media, Inc., 189–204.
5. Горбунов К.Ю., Любецкий В.А. 2009. Реконструкция эволюции генов вдоль дерева видов. *Молекуляр. биология*. **43**, 946–958.
6. Горбунов К.Ю., Любецкий В.А. 2010. Об одном алгоритме согласования деревьев генов и видов с учетом дупликаций, потерь и горизонтальных переносов генов. *Информационные процессы*. **10**, 140–144.
7. Горбунов К.Ю., Любецкий В.А. 2011. Дерево ближайшее в среднем к данному набору деревьев. *Проблемы передачи информации*. **47**, 64–79.
8. Doyon J., Scornavacca C., Gorbunov K.Yu., Szeollosi G.J., Ranwez V., Berry V. 2010. An Efficient Algorithm for Gene/Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers. *Lecture Notes in Bioinformatics* (Subseries of Lecture Notes in Computer Science*). Springer-Verlag Berlin Heidelberg*. **6398**, 93–108.
9. Вьюгин В.В., Гельфанд М.С., Любецкий В.А. 2002. Согласование деревьев: реконструкция эволюции видов по филогенетическим деревьям генов. *Молекуляр. биология*. **36**, 807–816.
10. Bansal M.S., Burleigh J.G., Eulenstein O., Wehe A. 2007. Heuristics for the gene-duplication problem: a $\theta(n)$ speed-up for the local search. *Lecture Notes Comp. Sci*. **4453**, 238–252.
11. Pisani D., Cotton J.A., McInerney J.O. 2007. Supertrees disentangle the chimerical origin of eukaryotic genomes. *Mol. Biol. Evol.* **24**, 1752–1760.
12. Guigo R., Muchnik I., Smith T.F. 1996. Reconstruction of ancient molecular phylogeny. *Mol. Phylogenet. Evol.* **6**, 189–213.
13. Wu D., Hugenholtz P., Mavromatis K., et al. 2009. A phylogeny-driven genomic encyclopaedia of bacteria and archaea nature, Vol 462|24/31 December 2009| doi:10.1038/nature08656.
14. Wehe A., Bansal M.S., Burleigh J.G., Eulenstein O. 2008. DupTree: A program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics*. **24**, 1540–1541.
15. Горбунов К.Ю., Любецкий В.А. 2005. Поиск предковых генов, нарушающих согласованность деревьев белков и видов. *Молекуляр. биология*. **39**, 847–858.