ISSN 0026-8933, Molecular Biology, 2012, Vol. 46, No. 1, pp. 161–167. © Pleiades Publishing, Inc., 2012.
Original Russian Text © K.Y. Gorbunov, V.A. Lyubetsky, 2012, published in Molekulyarnaya Biologiya, 2012, Vol. 46, No. 1, pp. 176–183.

BIOINFORMATICS

UDC 575.852

# Fast Algorithm to Reconstruct a Species Supertree from a Set of Protein Trees

**K. Y. Gorbunov and V. A. Lyubetsky**

*Institute for Information Transmission Problems, Russian Academy of Sciences, Bol'shoi Karetnyi per. 19, Moscow, 127994 Russia;*

*e-mail: gorbunov@iitp.ru, lyubetsk@iitp.ru*

Received January 27, 2010; in final form August 16, 2011

**Abstract**—The problem of reconstructing a species supertree from a given set of protein, gene, and regulatory-site trees is the subject of this study. Under the traditional formulation, this problem is proven to be NP-hard. We propose a reformulation: to seek for a supertree, most of the clades of which contribute to the original protein trees. In such a variant, the problem seems to be biologically natural and a fast algorithm can be developed for its solution. The algorithm was tested on artificial and biological sets of protein trees, and it proved to be efficient even under the assumption of horizontal gene transfer. When horizontal transfer is not allowed, the algorithm correctness is proved mathematically; the time necessary for repeating the algorithm is assessed, and, in the worst case scenario, it is of the order $n^3 \cdot |V_0|^3$, where $n$ is the number of gene trees and $|V_0|$ is the number of tree species. Our software for supertree construction, examples of computations, and instructions can be freely accessed at http://lab6.iitp.ru/ru/super3gl/. Events associated with horizontal gene transfer are not included either in this study or in any variant of the software. A general case is described in the authors' report (journal *Problems of Information Transmission*, 2011).

**DOI:** 10.1134/S0026893312010086

*Keywords*: species tree, species supertree, new formulation of the problem of supertree reconstruction; fast algorithm to reconstruct a supertree, generation of a gene set from a supertree, modeling the gene evolution along a species tree

## INTRODUCTION

Reconstraction of a species supertree from a set of protein trees (usually of a complex of orthologous protein groups, OPG) is a problem with a long history, the fundamental and practical importance of which is widely recognized. This is one of the NP-hard problems [1]. Numerous references to the subject are given in this study; a review of the results concerning this and related problems can be found in [2]; [3] contains references to recent publications. In practical terms, the time necessary for the problem's solution depends exponentially on the volume of initial data (the number of gene trees and the number of species in them). Therefore, an algorithm for the problem's solution can only be computationally efficient (in other words, an algorithm of polynomial complexity with a low polynome degree) due to a change in the problem formulation. Such a change should be acceptable in terms of biological applications. There were no published approaches to the formulation and solution of this task. Any formulation or solution in terms of this hard task has not been reported yet. However, a great number of exponential heuristic algorithms for supertree reconstruction have been proposed, and some of them are discussed in our review [4].

We propose a new formulation of the problem and algorithms for two cases of the problem's solution, i.e., with and without taking into account horizontal gene transfer. Here we only analyze the former cases; the other one is the subject of our reports [5–7]. The repeated time of the algorithm in the presence of the worst original data is of an order of the third degree of both $n$ original gene trees and the $|V_0|$ number of the species constituting these trees. In most cases, the algorithm repeated time is much less; as a rule, it is a quadratic function of the above-mentioned two parameters (see Testing). The same is true for more general cases, which have been previously discussed [5–8].

Thus, the task is to reconstruct an $S$ tree containing $|V_0|$ species from a given set of $G_i$ gene trees, where $i$ ranges from 1 to $n$. As in many other reports (for example, [4, 9–12]), the desired $S$ species tree is constructed like a supertree for the $G_i$ set; in other words, it is constructed as a tree, which is "generally the closest" to each $G_i$. This approach requires the specification of the proximity between the given gene $G$ and species $S$ trees. The concept of proximity is traditionally based on nesting the gene tree $G$ into the species tree $S$ with the only difference that, instead of the method of Guigo et al. [12], the method described in [5] is used.

This report is a direct continuation of the study described in [5].

## BASIC CONCEPTS

In a set of $V_0$ species, each $s$ species is assumed to be a given nonempty set of $G(s)$ genes. A combination of $G(s)$ sets will be considered broken up into clusters; in other words, a cluster contains homologous components. Each cluster includes any number of genes belonging to the same species.

A species tree is a binary rooted tree and the names of the species are assigned to tree leaves; the set of $V_0$ species and the set of tree leaves are in one-to-one correspondence. The gene tree corresponding to the gene cluster $K$ is a binary rooted tree and the name of each gene $g$ of $K$ is assigned to each leaf of the gene tree; the leaves of the gene tree and the genes of the $K$ cluster are also in one-to-one correspondence. For convenience, we assume that along with the $g$ gene, which is assigned to some leaf, the $s$ species, from which the $g$ gene originates, is assigned to the same leaf; we will say that these $g$ genes and $s$ species are in a "gene–species" relationship.

Note that, in other words, the gene tree leaves, which represent pairs, such as $\langle g_1,s\rangle$, $\langle g_2,s\rangle$, $\langle g_3,s\rangle$, etc., correspond to the paralogues $g_1$, $g_2$, etc., in the form of $s$. The genes assigned to leaves originate from a certain family of homologous genes mostly encoding a complex of orthologous groups of proteins represented in the GenBank and NCBI databases. In this sense, every gene tree defines a gene in its evolutionary development.

Let us assume that the root of a tree is "from above". We denote by $e^-$ and $e^+$ the upper and lower endpoints of edge $e$, respectively. An edge is understood as a pair of vertices with a $e^-$ starting point and $e^+$ ending point. The incoming edge of a vertex $g$ is denoted as $b_g$. As mentioned in [5], each tree is considered together with its "root edge", which is a specially added edge that comes up from the root and corresponds to the time when there existed a common ancestor of all species or genes constituting the tree: the upper endpoint of the root edge is referred to as a "superroot". The edges of a species tree $S$ are called *tubes*; in particular, a root edge is referred to as a *root tube* [5].

Suppose that $G$ is a gene tree. At the vertices of the $G$ tree, the ordering of the relationship "lower" will be defined as $g_1 < g_2$ if $g_1 \neq g_2$, and the path from the superroot to $g_1$ can be made through $g_2$. In the presence of a set of all the vertices and tubes in $S$, we will define the ordering relationship $y < x$ as follows: vertex or tube $y$ is "lower" than a certain vertex or tube in $S$ if $y \neq x$, and, in the $y$ tube, the path from the superroot can be made through $x$; respectively, "$x$ is above $y$"; we denote $y \leq x$ if $y < x$ or $y = x$.

Let us define the terms used in this article. It would be helpful to read the section Formulation of the Problem of the report of Gorbunov and Lyubetsky [5]. Unlike their report, in this study, we only discuss nesting and scenarios without horizontal transfers and, therefore, we refer to them as simply nesting and scenarios. Horizontal transfers are discussed in [5–7].

*Nesting* of the $G$ gene tree into species tree $S$ implies the reflection $f$ of all $V(G)$ vertices of the $G$ tree in the $V(S)$ vertices and $E(S)$ tubes of the $S$ tree, when the following conditions are fulfilled:

(1) The superroot of $G$ is reflected in the root tube of $S$; each leaf $g$ in $G$ is reflected in leaf $s$ of $S$ according to the relationship gene–species;

(2) If $g_1$ —descendant of $g$ and $f(g)$ —is the vertex, then $f(g_1) < f(g)$; whereas if $f(g)$ is a tube, then $f(g_1) \leq f(g)$.

(3) Let us agree that $g_1$ and $g_2$ are the descendants of the $g$ vertex: if $f(g)$ is a vertex, then $f(g_1)$ and $f(g_2)$ are located in different subtrees that are rooted in the descendants of the $f(g)$ vertex.

In other words, condition (3) means the following: the species, to which the $g$ gene belongs, is the last common ancestor of the species to which $g_1$ and $g_2$ belong.

For the given $f$ nesting, the following formal definitions of evolutionary events should be noted [5]. Gene *duplication* is determined by the nonsuperroot $g$ vertex in $G$, for which $f(g)$ is a tube from $S$. A *loss* of the gene is determined by the pair $\langle e, s\rangle$, where $e$ is an edge in $G$, $s$ is a vertex in $S$, and $f(e^+) < s < f(e^-)$. The speciation (with respect to the gene under consideration) is determined by the $g$ vertex from $G$, where $f(g)$ is a vertex in $S$ and both $g$ and $f(g)$ vertices are not leaves. We only consider speciation associated with bifurcations in the $G$ tree; since their cost is assumed to be lower than zero, speciation is not discussed as a separate event.

Figures 1a and 1b demonstrate an example of nesting a gene tree within a species tree

Let us explain the informal definitions. The internal vertices of a species tree correspond to (hypothetic) ancestral species; the internal vertices of a gene tree correspond to (hypothetic) ancestral genes. Ideally, nesting a gene tree into species tree gives information about the ancestral species to which the given ancestral gene belongs (for the leaves, i.e., modern species and genes, this is satisfied by definition).

Note that the vertices of a species tree do not indicate all ancestral species, but only those involved in speciation (divergence into two species). Between speciation events, some species could evolve and some genes could double ("duplication"). In such a case (a species evolves but does not diverge), a new species will (implicitly) correspond to a certain internal point on the edge ("tube") of the species tree. The gene–tree vertex corresponding to gene duplication is therefore reflected in a tube rather than in a vertex.

Similarly, in a gene tree, the vertex corresponds to a hypothetic ancestral gene $g$ only if two new genes evolved from the $g$ gene, irrespective of whether this

was accompanied by speciation or not (in the former case, a vertex of the species tree corresponds to that gene) (see above). If after speciation (in this case, a certain vertex $s$ is present in the species tree) none of the resultant species contains an analogue of the $g$ gene, then no "gene duplication" occurred during species divergence. Therefore, in this case ("a loss of gene"), even a vertex $s$ of the species tree does not correspond to any vertex of the gene tree. We can say that $s$ corresponds to a point of some $e$ edge of the gene tree. The condition $f(e^+) < s < f(e^-)$ is clearly satisfied. Note the following: First, several gene losses, i.e., several vertices $s$ of the $S$ species tree, might correspond to a single edge $e$ of the $G$ gene tree if the condition $f(e^+) < s < f(e^-)$ is satisfied. Similarly, several gene losses in the gene tree (i.e., several edges $e$ for which the condition $f(e^+) < s < f(e^-)$ is satisfied) might correspond to a single vertex $s$ of the species tree. This is due to gene duplication (see, for example, Figs. 2 a, 2b).

For any gene tree $G$ and specie tree $S$, as well as for any nesting $f$, the following designations are used: $l(f, G, S)$ is the number of losses; $d(f, G, S)$ is the number of gene duplications for the nesting $f$. The costs of a single gene loss and of a single duplication are denoted as $c_l$ and $c_d$, respectively. The cost of a single speciation is believed to be zero; however, if this cost is lower than $c_d + 2c_l$, the algorithm and following assertions are preserved.

## FORMULATION OF THE PROBLEM

A set $\{G_i\}$ is given, which consists of $n$ rooted binary gene trees. The species tree $S$ is consistent with the $\{G_i\}$ set if the set of leaves in $S$ coincides with the $V_0$ set of all species presented by the leaves of all $G_i$ trees. Our goal is to formalize the problem of constructing a species tree $S$, which is the "closest" to the whole set of $\{G_i\}$ trees. We begin with an auxiliary task.

**Gene evolution scenario along a species tree: supertree.** Let us agree that we have a given set of gene trees $\{G_i\}$ and a coordinated species tree $S$. The nesting $f$ of the $\{G_i\}$ set into the $S$ tree indicates a set of nesting $\{f_i\}$, where each $f_i$ is nesting of $G_i$ into $S$.

**Task A1.** Having a $\{G_i\}$ set and an $S$ species tree, it is required to find the nesting $f$ for the $\{G_i\}$ set and $S$ tree, which reaches a minimum functional

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

Note that $c_l \cdot \Sigma_i l(f_i, G_i, S)$ is the total cost for all losses in all $G_i$, whereas $c_d \cdot \Sigma_i d(f_i, G_i, S)$ is the total cost for all duplications in all $G_i$.

The nesting that reaches a minimum in (1) will be called a scenario.

We have proved that when $\{G_i\}$, $S$, and the coefficients $c_l$ and $c_d$ are fixed, the scenario is unique and does not even depend on the choice of any nonnegative values of these coefficients.

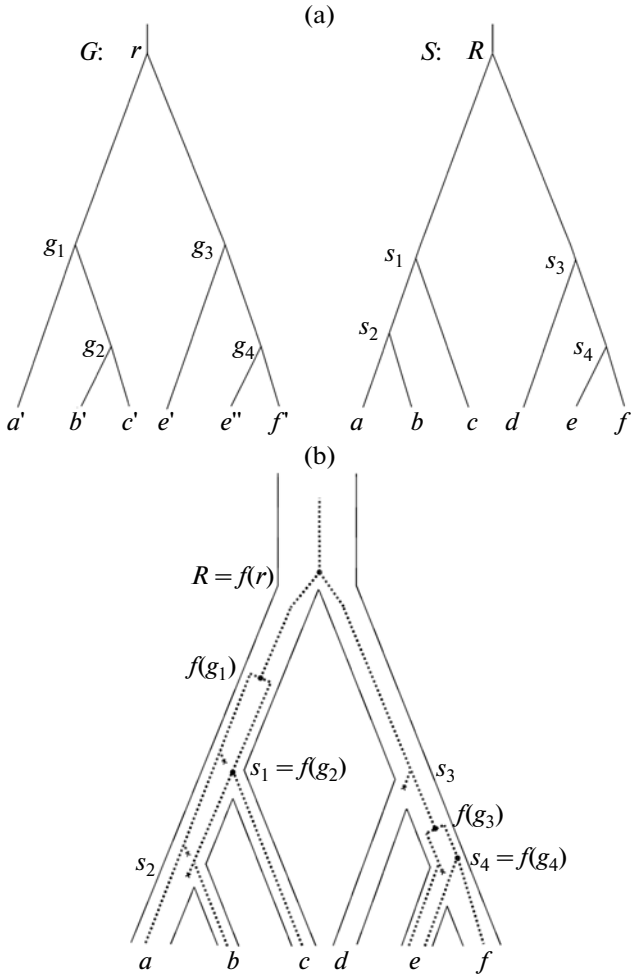


**Fig. 1.** Illustration of the concepts of duplication, gene loss, and speciation. (a) An example of gene tree $G$ and species tree $S$, in the leaves of which gene $a'$ is taken from species $a$, etc., and paralogs $e'$ and $e''$ are taken from species $e$. There is no species $d$ in the $G$ tree. (b) The values of nesting $f$ the $G$ tree into the $S$ tree are clearly demonstrated; the values on the leaves of $G$ coincides with appropriate leaves in $S$. The values of $f$ reflection on the internal vertices of the $G$ tree are marked with bullet points. The $f(g_1)$ value is indicated in the tube (though it is formally equal to this tube), and, according to the definition of duplication, the $g_1$ vertex corresponds to a duplication event; the same is true for the $g_3$ vertex. The values of $f$ on all other internal vertices of the $G$ tree coincide with the corresponding internal vertices of the species tree and, according to the definition of speciation, correspond to speciation events. For the edge $h = (g_1, a')$, the $s_1$ and $s_2$ vertices lie between the values at the ends and, according to the definition of loss, the $(h, s_1)$ and $(h, s_2)$ pairs correspond to the events of losses; in the figure, they are the processes with a cross at the end. Similarly, the $(g_2, b'), s_2)$, $((g_3, e'), s_4)$, and $(r, g_3), s_3$ pairs are losses.

Vertices $g$ in the gene tree and $s$ in the species tree will be considered compatible if they are not superroots and one of the following conditions is satisfied: (1) $g$ and $s$ are leaves that are in the gene–species relationship or (2) $g$ has descendants $g_1$ and $g_2$, whereas $s$ has descendants $s_1$ and $s_2$ and

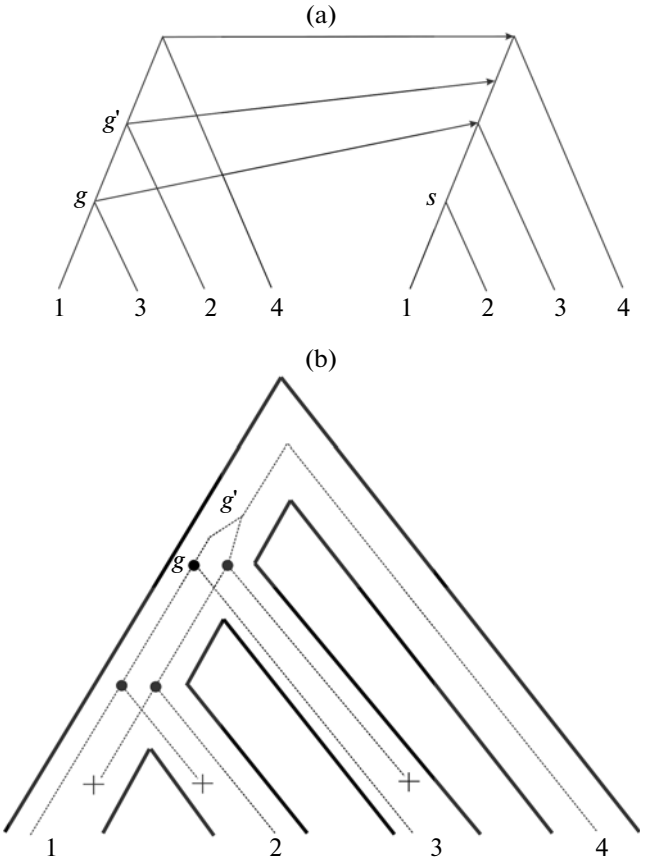


**Fig. 2.** (a) Nesting a gene tree (on the left) into a species tree (on the right). In the example, four species are indicated with figures from 1 to 4. In the leaves of the gene tree, each gene corresponds to each species; therefore, only the species numbers are indicated on the leaves of the tree.

The evolutionary events are the following: duplication in the $g'$ vertex and three losses, which correspond to the edge–vertex pairs $(2, g')$ and $s$; $(2, g')$ and $s'$; and $(1, g)$ and $s$, where $s'$ is the ancestor of $s$. (b) The same nesting is shown through "inscribing" the gene tree into the tubes of the species tree. The nesting has one duplication in the $g'$ vertex and three losses marked with crosses. The gene divergence due to speciation is indicated with points.

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

$X \subseteq Y$ indicates that the $X$ set is a part (a subset) of the $Y$ set. Remember that $b_s$ indicates a tube of the $S$ tree with the end point in the $s$ vertex.

**Lemma 1.** Suppose that $G$ is a gene tree and $S$ is a species tree. For any scenario $h$ that corresponds to $G$ and $S$ and for any fixed nonnegative values of the costs for a single duplication and single loss, the following is true. Let $g$ be vertex $G$ distinct from a superroot, whereas $s$ is the last common ancestor of the $M_g$ set in the $S$ tree. Then, $h(g) = s$, if the $g$ and $s$ vertices are compatible, and $h(g) = b_s$ if these vertices are incompatible.

The proof was given earlier in [7].

Let the nesting described in lemma 1 is a *scenario* $h(\{G_i\}, S) = \{h_i\}$ for evolving the genes of the $G_i$ tree along the $S$ species tree. The value $c(\{G_i\}, S) = c(\{G_i\}, h(\{G_i\}, S), S)$ is the *cost of the scenario.*

The $S^*$ supertree for a set of gene trees $\{G_i\}$ is called a species tree for which

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

takes the minimum possible value.

**Task A2.** Reconstruction of a supertree from a given gene tree. This seems to be a traditional formulation of the task to reconstruct a specie tree on the basis of minimizing functional (1*). Different versions of the algorithm for constructing a supertree have been presented earlier [4, 9–12]. All these algorithms ensure the solution of the above-mentioned task during exponential time and allow for only heuristic approximations of the $S^*$ tree. The $S^*$ tree itself, as a rule, remains unknown, except for the cases of artificially selected data.

**New formulation of the task.** The set of species assigned to the leaves below a certain vertex $v$ of the S tree is called a clade $M_v$ of the $S$ species tree; the vertex itself, $v$, is the clade root. This set of leaves is denoted as "a set of clade leaves". Similarly, a set of species assigned to the leaves below a certain vertex $g$ of $G$ is denoted "clade $M_g$ in the $G$ gene tree; this vertex is the clade root.

We suggest considering the following task.

**Task B.** From a given set of gene trees $\{G_i\}$, the $S^*$ species tree should be found so that (i) all clades of $S^*$ belonged to a predetermined sets $P$. Thus, the $P$ set is a parameter of the problem; (ii) the value of function (1*) for the $S^*$ supertree does not exceed the values of the functions for other species trees that satisfy condition (i).

The set of clades of the desired species tree is known to include a set of all species $V_0$ and all its elemental sets but does not include an empty set. Therefore, we will only consider $P$ sets containing the aforementioned sets. The algorithm described below is applicable to any $P$ set, but a typical example of such a set is $P_0$ that includes all clades of all original gene trees and is augmented by the set of all $V_0$ species. This is a *standard* set for a given set of gene trees. We denote by $|X|$ the number of elements of the $X$ set. The number of elements in a standard $P_0$ set is assessed from above: $|P_0| \le 2|V_1|$, where $V_1$ is the set of all leaves in all gene trees. Under the normal assumption that the average number of leaves in gene trees is of an order of $|V_0|$, we obtain $|P_0| \le K|V_0| \cdot n$, where $K$ is a certain constant, which does not exceeds 2 in our data; remember that $n$ is the number of original gene trees and $V_0$ is the set of all species in them. If the $P$ set does not include all the sets of standard $P_0$, it can be expanded with the sets from $P_0$. Therefore, we assume that the $P$ sets discussed below include all sets from standard $P_0$.

The gene tree $\{G_i\}$ and $P$ sets are considered fixed. The following two definitions play a significant role in our approach.

**First definition.** Suppose that $e$ is an edge in gene tree $G$. Let us define $M_e$ as a set of species assigned to all leaves below the $e$ edge in the $G$ tree. The $M_d$ set for a tube $d$ of the $S$ species tree is defined similarly. For the $G$ gene tree and a set of species $M$, the set of $e$ edges in $G$ is $Ed(M, G)$, where $M_e \subseteq M$ and there is no $e' > e$ with this property. There might be several edges $e$ of this kind, but they are incomparable in $G$.

Suppose that $f$ is the nesting of the $G$ gene tree into the $S$ species tree. Then the assertion that "the edge $e$ of $G$ enters the tube $d$ of $S$" suggests that $f(e^+) \le d < f(e^-)$ ("enters" is in geometrical sense).

**Lemma 2.** For the $h(G, S)$ scenario from lemma 1, the following is satisfied:

(a) The edges from $Ed(M_b, G)$ enter accurately the tube $b$ of $S$.

(b) If the $b_1$ tube is the descendant of the $b$ tube, then the only edge $e' \ge e$ entering $b$ exists for any edge $e$ of $G$ that enters $b_1$.

The proof was given earlier in [7].

**Second definition.** Set $V$ of $P$ is called basic if it can be divided into two parts of $P$ and each part, in turn, can be divided into two parts of $P$ and so on until one-element species-representing sets are reached.

Obviously, the B task has a solution if and only if the all-species $V_0$ set is a basic one.

## COMPUTER PROGRAM FOR SUPERTREE RECONSTRUCTION

The algorithm and its rationale have been previously reported [7]; the software itself is freely available at the website http://lab6.iitp.ru/ru/super3gl/. Several notations and explanations required for working with the program are presented below.

The heuristic solution of task A2 consists of two steps. The first step involves the construction of the so-called "basic trees" $S(V)$ for all basic sets $V$ from the given $P$ sets. At the second stage, the $S(V)$ set is used to obtain an approximation $S'$ of the desired supertree $S^*$ for $\{G_i\}$ (see additional materials, items 1–3, www.molecbio.com/downloads/2012/1/supp_gorbunov_en.pdf).

The computer program is based on the following theorem.

**Theorem 1.** Suppose that $P$ is a set of clades.

(a) If the $V_0$ set is a basic one, the $S(V_0)$ tree is a solution of task B. Otherwise, task B has no solutions.

(b) If $P$ is a standard set and the average number of leaves in a set of gene trees $\{G_i\}$ is of an order of $|V_0|$, then the algorithm determines the $\{S(V)\}$ set, where the $V$ variable runs over all basic sets for a number of steps of an order $|V_0|^2 n + |P|^2|V_0| + |P||V_0|n + |P|^3 + |P|^2|V_0|n \le Cn^3|V_0|^3$. During this process, the algorithm produces a solution to task B or reports that no solution exists. Memory of the order of $n^2 \cdot |V_0|^2$ is sufficient.

The proof of theorem 1 was given earlier [7].

To solve task A2, an auxiliary algorithm should be applied for the set of basic trees (see supplementary materials, item 4).

A computer program for constructing a set of trees $\{S(V)\}$, where $V$ runs over all the basic sets of the $S'$ tree, has been developed by L.I. Rubanov; it is freely available at the website http://lab6.iitp.ru/ru/super3gl/ together with examples of computation and operating instructions.

The super3GL program is capable of handling large sets of original trees, including nonbinary ones; it is primarily intended for computing on a multiprocessor system with MPI-1.2 support, although working on a standard PC is also possible. The program is written in the C++ programming language and has a command line interface. The source code can be moved, and, after recompilation, it can be used in the OS Windows 32/64-bit, Linux, Unix, and MacOS systems. The program's executable modules for Windows 32/64 bit (uniprocessor and parallel versions) can be freely downloaded from this website; the source code is available under a free license for noncommercial use in scientific and educational organizations. Parallel modules for Windows are designed to work with the MPICH2 system developed by the Argonne National Laboratory (versions 1.3.2 or higher); the appropriate version(32/64 bit) should be installed on the multiprocessor used. Since the efficiency of algorithm paralleling varies during the construction of the basic trees and supertree, the complexity of these steps depends on the problem and the program allows them to be run together and separately, including on different computers. The performance of the program on a variety of computing facilities is discussed in detail in the manual.

The downloadable files are listed in the supplementary materials (item 5): program description (PDF); uniprocessor version of the program (Windows 32 bit); uniprocessor version of the program (Windows 64 bit); option for MPICH2 v.1.3.2 (Windows 32 bit); option for MPICH2 v.1.3.2 (Windows 64 bit); utility to decrypt the abbreviations in the species tree; and a script for rooting trees.

The results of testing the program using artificial sources of data are given below; the correct answer was obtained in advance. The results of testing the biological source of data from the Hodgenom database are in the supplementary materials (item 6). An example with 276 species is given in two versions, as well as an example with 814 species, etc. In these cases, the answer was unknown, but the results are consistent with the known species trees [11, 13].

## ALGORITHM TESTING

Algorithm testing was conducted as follows: first, from the $S$ species tree (randomly selected or biological), a set of gene trees $\{G_i\}$ was constructed, for which $S$ is definitely a supertree, because the verification of the trees in the vicinity of the $S$ tree showed that function (1*) had an $S$ minimum. The argument in favor of our algorithm or any other one is that this algorithm reconstructs $S$ from $\{G_i\}$ completely or partially.

The task of $\{G_i\}$ reconstruction from a given $S$ set is nontrivial and of great interest. To solve this problem, we used the following approach: a $\{G_i\}$ set was reconstructed by modeling the "real" process of gene evolution along the $S$ tree, as it has been described earlier [5]. It is convenient to consider S trees, which also include vertices with one descendant. Namely, the $p_d(x)$ and $p_l(x)$ probabilities were given in accordance with duplications and losses, which might occur in each tube of an $S$ tree. Suppose we have already constructed, starting from the root edge, a $G'$ part of the expected $G_i$ tree; the $x(v)$ tube from $S$ or $\langle i$ (a leaf of $S$) is assigned to $G'$ of each terminal vertex $v$. In the second case, the terminal vertex is a leaf of $S\rangle$. Let us verify all terminal vertices $v$ in $G'$, to which the tube is assigned. Duplication is simulated in each $v$ with a probability of $p_d(x(v))$. If this event does occur, then bifurcation appears in $x$ and the two new terminal vertices belong to the same tube $x$. If the event does not occur, the following three cases are possible.

(1) Tube $x$ leads to the bifurcation of the $S$ tree. Then the loss is simulated twice with a probability of $p_l(y)$, where $y$ is any of the descendants of tube $x$. If a loss occurred at least once, then such a loss is simulated with an equal probability in any of the two tubes outgoing from the bifurcation. The tube adjacent to the tube, where the loss did occur, is assigned to vertex $v$. If a loss has not occurred, the bifurcation is determined in $x$, and the descendant of tube $x$ is assigned to each of the two newly emerged terminal vertices.

(2) Tube $x$ leads to a vertex with an only descendant. Then the tube, which is the descendant of tube $x$, is assigned to vertex $v$.

(3) Tube $x$ leads to a leaf of the $S$ tree. Then a pair with the name of the species attributed to this leaf is assigned to vertex $v$.

We have selected the $p_d(x)$ and $p_l(x)$ dependencies so that the number of different events during evolution, as well as their distribution along the tubes, were close to those observed after nesting of biological OPG trees into natural species trees. The appropriate data were taken from reports [4, 5, 11] and our own unpublished data. Of course, more extensive data are required for the adequate reconstruction of these distributions, and distortions caused by the evolution scenario should be taken into account.

Thus, in all tests, the $S'$ tree obtained by merging the basis trees was expected to match or be close to the $S$ supertree known for these special conditions. Of course, the $S$ supertree was not specified for the algorithm; it was only used when comparing $S'$ and $S$. As input data, the algorithm received only a set of gene trees $\{G_i\}$, which was modeled from the $S$ tree.

**1. Reconstruction of an artificial balanced binary tree with 64 leaves.** In this example, $S$ is a balanced binary species tree with 64 leaves. The values of the aforementioned probabilities were the following: $p_d$ decreased gradually from the root tube to the leaf tubes from 0.3 to 0.01 (this was not accompanied by an increase in numerous paralogs); $p_l$ decreased gradually from the root tubes to the leaf tubes from 0.5 to 0.25 (near the root, the probability is higher for the gene to leave no descendants capable of reaching the leaves). For balanced trees, in which the lengths of all paths from their roots to leaves are equal (in this point and in point 3 thereafter), both $p_d$ and $p_l$ values were analytically given. Namely, suppose that $x$ is a tube and $\rho(x)$ is the number of tubes before $x$ (inclusively), beginning from the root tube. Note that $|V_0|$, which is the number of leaves in the desired $S$ tree, is designated $b = (\log_2|V_0|) + 1$. Then $p_d$ and $p_l$ represent a linear function; on the segment $[1, b]$ or $[2, b]$, it is determined by the values at the ends of the same segment, which are specified above (the second figure is the value of $b$).

Thus, a set of 1000 artificial trees has been generated, in which the number of leaves ranged from 32 to 96; on average, 70 leaves and 39 species occur on one tree. On average, 16 duplications and 37 losses fall on one tree. After inputting the generated trees into our program, the algorithm restored accurately the original species tree. Each test was repeated 20 times.

**2. Reconstruction of a natural bacterial tree with 40 leaves.** This supertree $S$ was taken from a previous study (Fig. 4) [5] and used for the generation of a set of gene trees as in the previous example. The following values of the probabilities of the events were taken: $p_d$ and $p_l$ decreased gradually from the root to the leaves within the ranges 0.1 to 0.01 and 0.5 to 0.25, respectively. In its topology, this tree is closer to a "comb" than to a balanced tree; therefore, the probability of duplication is reduced as compared to example 1, and the number of duplications is not too large in comparison with known evolutionary scenarios. For unbalanced trees (in this point and in point 4 thereafter), the more complex $p_d$ and $p_l$ values were taken. First, balanced $+S$ trees were constructed for the $S$ trees indicated in these points by means of dividing the original tubes with new vertices with one descendant each; the meaning of such a partition has been discussed previously [5] (section Algorithms of Constructing the Internal Tree and Temporal Layers, item c). The algorithm described in the section has been used for this purpose. Then the $p_d$ and $p_l$ values were calculated using the above linear function for the $+S$ tree. Note that the $S'$ tree obtained from the $\{G_i\}$ set, which was reconstructed from $+S$, was compared with the $S$ tree, because our algorithm of supertree reconstruction determines a tree just within bifurcations.

In this manner, a set of 1000 artificial gene trees was generated, which contained from 30 to 60 leaves (on average, 51 leaves and 31 species per tree). On average, 15 duplications and 29 losses occur on one tree. After inputting the generated set of trees into the program, the algorithm accurately restored the original species tree. This test was repeated 20 times.

**3. Reconstruction of an artificial balanced binary tree with 128 leaves.** In this case, generation of gene trees from the given *S* supertree occurred similarly and with the same probability as in example 1. In this manner, 1000 artificial gene trees were generated, in which the number of leaves ranged from 64 to 192 (on average, a tree contained 146 leaves and 77 species). On average, 33 duplications and 79 losses occur on one tree. After inputting the generated set of trees into the program, the algorithm produced a tree, which coincided with the original one. This test was repeated 20 times.

**4. Reconstruction of a natural species tree with 169 leaves.** In this case, the set of trees was constructed as in the preceding examples, i.e., along the natural *S* supertree with 169 leaves (see the report of Pisani et al. [1], Fig. 1). The probability values were the same as in example 2. In this manner, 1500 artificial gene trees were generated, in which the number of leaves ranged from 120 to 200 (on average, one tree contained 170 leaves and 130 species). On average, 70 duplications and 107 losses occur on one tree. The algorithm restored the original tree with 169 leaves. This test was repeated 20 times.

## CONCLUSIONS

We propose a new formulation of the problem of reconstruction of a species supertree on the basis of a set of protein (gene) trees; namely, we assumed that most clades of the desired species tree are represented in at least one protein tree from the original set; therefore, the search for the supertree is conducted among the species trees, most clades of which are represented in at least one of the original protein trees. For the purpose of supertree reconstruction, we developed a new heuristic algorithm of an almost quadratic complexity. In the case of the evolutionary scenario without horizontal transfers, this algorithm proved to solve accurately the task and has a cubic complexity when the initial data are the worst [7]. The case of horizontal transfer has been previously discussed by us [7]. In our tests, a set of protein trees has been reconstructed from some or other species tree by modeling the process of gene evolution along this tree. Using a set of protein trees obtained by this method, our algorithm reconstructed the original supertree; the reconstruction process was rapid, and it restored accurately the supertree. The results of tests, where biological data were used, are described in the supplementary materials (item 6). The assumption itself seems to be natural when considering various biological problems.

## REFERENCES


1. Ma B., Li M., Zhang L., et al. 1998. On reconcstructing species trees from gene trees in term of duplications and losses. *Proc. Second Annu. Int. Conf. Res. Computat. Mol. Biol.* NY: ACM, pp. 182–191.
2. *Phylogenetic Supertrees. Combining Information to Reveal the Tree of Life*. 2004. Ed. Bininda-Emonds O.R.P. Dordrecht: Kluwer.
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*. Eds. Kolchanov N., Hofestaedt R., Milanesi L. Springer, pp. 189–204.
5. Gorbunov K.Yu.,Lyubetsky V.A. 2009. Reconstructing the evolution of genes along the species tree. *Mol. Biol.* (Moscow). **43**, 881–893.
6. Gorbunov K.Yu., Lyubetsky V.A. 2010. An algorithm of reconciliation of gene and species trees and inferring gene duplications, losses and horizontal transfers. *Inform. Protsesses*. **10**, 140–144.
7. Gorbunov K.Yu., Lyubetsky V. A. 2011. The tree nearest on average to a given set of trees. *Probl. Inform. Trans*. **47**, 274–288.
8. Doyon J.-P., Scornavacca C., Gorbunov K.Yu., Szöllösi G.J., Ranwez V., Berry V. 2010. *Lecture Notes Comp. Sci*. **6398**, 93–108.
9. V'yugin V.V., Gelfand M.S., Lyubetsky V.A. 2002. Identification of horizontal gene transfer from phylogenetic gene trees. *Mol. Biol.* (Moscow). **37**, 650–658.
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*. **462**, 1056–1060. 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. Gorbunov K.Yu. and Lyubetsky V.A. 2005. Identification of ancestral genes that introduce incongruence between protein- and species trees. *Mol. Biol.* (Moscow). **39**, 741–751.