Supplementary materials for the paper of K. Gorbunov and V. Lyubetsky
«A fast algorithm to build a supertree with a set of gene trees»

1. Preparatory stage.

In task B, for each gene tree G_i and all sets V and P the set $Ed(V,G_i)$ is to be constructed. The trivial preparatory stage described below is used to construct $Ed(V,G_i)$ for a given set V and gene tree G_i.

The preparatory stage (under a fixed P) includes: 1) construction of a 0-1-vector sequence of length $|V_0|$ at each tree vertex defining a tree clade, i.e. each set in P is a vector of 0's and 1's of length $|V_0|$; the number of elements is determined for each set P; 2) computing tags for each vector pair from P to indicate if those are nested or intersecting; 3) construction of sets $Ed(V,G_i)$ for all sets $V \in P$ and trees G_i, and determining the sets cardinality, $i = 1, .., n$; 4) construction of all decompositions of each V into two sets from P.

1) Vectors are constructed with induction from the root toward the leaves. We assume that the mean number of leaves has the order of $|V_0|$, i.e. the total number of leaves to visit is $|V_0|\cdot n$, the time of visiting one vertex is $|V_0|$; the total processing time is $|V_0|^2\cdot n$.

2) The ratios of vector nesting and intersecting are computed in time $|P|^2\cdot|V_0|$.

3) Sets $Ed(V,G_i)$ under a given V for all gene trees G_i are constructed by visiting each tree edge toward the leaves. If for current edge e the expression $M_e \subseteq V$ is true, it is added to $Ed(V,G_i)$ and the next edge e_1 is visited; e_1 is not comparable with e relative to the order <. The visiting time is $|V_0|\cdot n$, because the information $M_e \subseteq V$ is already computed on step 2). The total stage computing time is then $|P|\cdot|V_0|\cdot n$.

4) Decompositions of each set V from P into two sets from P are computed in time $|P|^3$, which normally does not exceed $Cn^3\cdot|V_0|^3$. For each triad V_1, V_2, V from P it is to be determined that V_1 and V_2 are not intersecting, V_1, V_2 are nested in V and $|V_1|+|V_2| = |V|$. All decompositions from P require memory storage of $|P|^2$, because each set has $|P|$ decompositions at maximum.

Thus, the entire preparatory stage requires the time of $|V_0|^2\cdot n+|P|^2\cdot|V_0|+|P|\cdot|V_0|\cdot n+|P|^3$, which normally does not exceed $Cn^3\cdot|V_0|^3$. Storing all vectors, ratios, decompositions and sets $Ed(V,G_i)$ requires the memory of $|P|\cdot|V_0|+|P|^2$, which in average cases does not exceed $Cn^2\cdot|V_0|^2$.

2. The principal algorithm of building basic trees. A sketch of the algorithm is provided, while a detailed description is given in [7].

As a result of the preparatory stage, sets $Ed(V,G_i)$ are built for each gene tree G_i and each set V from P. For each V, all decompositions in P are constructed. As described below,

dynamic programming is used to define basic sets V from P, and each basic V is computed a score $c(V)$ and minimal decomposition into two sets from P with induction. Such decomposition of each basic set V defines a tree $S(V)$, the basic tree in set V. The vertices of $S(V)$ correspond to sets from P, the root – to set V. The set of leaves in $S(V)$ corresponds to the set V, clades of $S(V)$ are contained in P. If set V_0 is not basic, the task B cannot be solved. Otherwise, the tree $S(V_0)$ is a solution of the task. Scenarios for all trees G_i and species tree $S(V_0)$ are constructed concurrently.

Define a subtree T of the tree G_i as consistent with the set of leaves V if (1) the set of leaves in T is nested in V, and (2) for no other tree that contains T condition (1) is true. In scenarios, superroots of all subtrees in G that are consistent with V will map into the root tube $S(V)$, and $c(V)$ is the total score of these mappings. The score is computed along with other parameters that allow to reconstruct the mappings (ref. to the example below).

Description of the induction process.

At the initial step, singleton sets V from P are defined, where $c(V)=0$ and the corresponding basic tree $S(V)$ has a single leaf.

Define V as a set from P; all smaller sets are already processed. Define (V_1,V_2) as a decomposition of V. If no such pair is found, the set V is tagged as non-basic and is not processed further. With induction, scores $c(V_1)$, $c(V_2)$ and trees $S(V_1)$, $S(V_2)$ are already computed.

All trees G_i are arbitrarily visited. In the current G_i, all vertices are visited to determine the number $k(V,V_1,V_2,G_i)$ of vertices, for which one descendant edge belongs to $Ed(V_1,G_i)$, and another – to $Ed(V_2,G_i)$. Define

$$(V,V_1,V_2,G_i)= Ed(V_1,G_i)|+|Ed(V_2,G_i)|–2\ k(V,V_1,V_2,G_i)$$

and

$$d(V,V_1,V_2,G_i)=Ed(V_1,G_i)|+|Ed(V_2,G_i)|–|Ed(V,G_i)|–k(V,V_1,V_2,G_i).$$

Here, powers of all sets $Ed(V,G_i)$ are computed during the preparatory stage. Find the decomposition of V into V_1^* and V_2^*, for which

$$c(V,V_1,V_2) = \sum_i [c_l \cdot l(V,V_1,V_2,G_i) + c_d \cdot d(V,V_1,V_2,G_i)] + c(V_1) + c(V_2) \qquad (2)$$

is minimal over all V_1 and V_2. Then, by definition, $c(V)$ is value (2) at *the minimal decomposition* $<V_1^*,V_2^*>$. The basic tree $S(V)$ is obtained by adding the root to basic trees $S(V_1^*)$ and $S(V_2^*)$; the root corresponds to V, and its descendent clades – to V_1^* and V_2^*.

The end of principal algorithm. An accurate and sophisticated proof of (2) is given in [7]. The computing time of $|P|^2\cdot|V_0|\cdot n$ is achieved by processing at maximum $|P|$ of different decompositions for each set from P and visiting all vertices in all trees in each decomposition.

Computing the ratios of nesting and intersecting of sets from P requires the time of $|P|^2\cdot|V_0|$, building sets $Ed(V,G_i)$ for all V from P – the time of $|V_0|^2\cdot n+|P|^2\cdot|V_0|+|P|\cdot|V_0|\cdot n$,

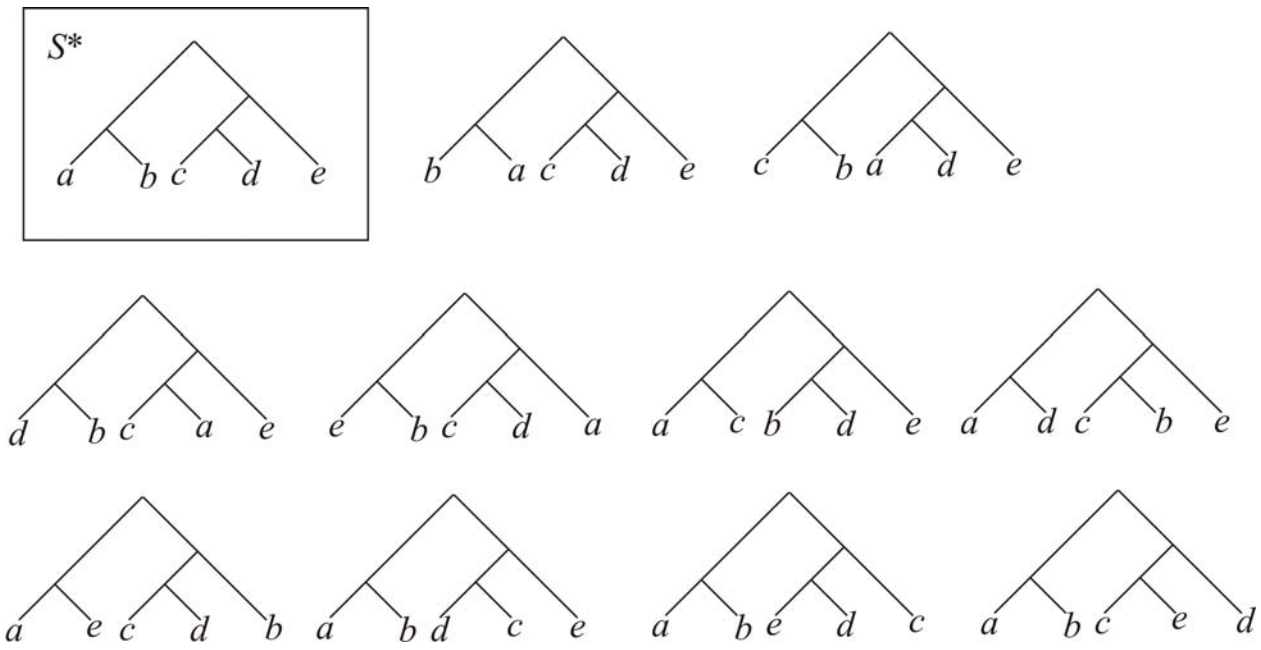
constructing all decompositions – the time of $|P|^3$. Thus, the total time complexity of the principal algorithm and the preparatory stage is $|V_0|^2 \cdot n + |P|^2 \cdot |V_0| + |P| \cdot |V_0| \cdot n + |P|^3 + |P|^2 \cdot |V_0| \cdot n$, which normally does not exceed $Cn^3 \cdot |V_0|^3$. Storing all vectors, ratios, decompositions and sets $Ed(V,G_i)$ requires the memory of $|P| \cdot |V_0| + |P|^2$, which in average cases does not exceed $Cn^2 \cdot |V_0|^2$.

Below we provide intuitive justifications of (2) that are formalized in [7]. Define fixed mappings f_i of each G_i with the set of leaves V_0 into the species tree S, and a subtree S' with the set of leaves V in S. The duplication point g from G_i or speciation point g from G_i is $f_i(g)$ if $f_i(g)$ is a tube or a vertex in the species tree, correspondingly. Call the loss point $<e,s>$ the vertex s. Lemma 2a dictates that the total score $c(V,S')$ of duplications and losses in the subtree S' starting from its root tube d over all trees depends only on S'. Indeed, $c(V,S')$ is defined only by $\bigcup_i(\mathrm{Ed}(V,G_i))$ and the topology of subtree S'. The root branching in S' defines the decomposition of set V into sets V_1 and V_2. The algorithm enumerates all decompositions of V into sets from P, therefore it is enough to prove that (2) correctly estimates the minimal score $c(V,S')$ over all S', all clades from S' belong to P and the root branching in S' produces the decomposition of V into V_1 and V_2. In induction, the second and third items already equal the minimal scores of subtrees with leaf sets V_1 and V_2, starting from the root-descendent tubes d_1 and d_2, correspondingly. It is to show that the first item equals the total score of duplications in the root tube d and losses in the root branching r of subtree S'. Let d contain n edges over all input trees, d_1 contain n_1 edges, and $d_2 - n_2$ edges. According to Lemma 2a, these equal the powers of sets $\bigcup_i(\mathrm{Ed}(V,G_i))$, $\bigcup_i(\mathrm{Ed}(V_1,G_i))$, $\bigcup_i(\mathrm{Ed}(V_2,G_i))$. As a duplication cannot occur in a tube, the number of edges in tube d can only increase due to duplications. All edges in G_i that reach the end of tube d pass the branching r and bifurcate (speciation) or continue in tubes d_1 or d_2 with truncation at the branching (a loss occurs in one of the descendent tubes). Thus, $n \le n_1+n_2$ (ref. also to Lemma 2*b*). Consider the set M of input tree vertices that map into d or r, the vertices intercalating the edges that map into d and those that map into d_1 or d_2. The set M is divided into non-intercesting subsets corresponding to edges mapped in d and defines a tree with leaves that correspond to some edges mapped in d_1 or d_2. The number of inner vertices in a tree is the number of leaves minus one. Therefore M contains the exactly $k=n_1+n_2-n$ of vertices, each representing a duplication in tube d or speciation at branching r. Then, the total number of duplications and speciations equals k. The number of speciations is obtained by visiting all vertices in all trees and finding vertices with one descendent edge contained in d_1, and the other – in d_2; such vertices correspond exactly to speciations in r according to the minimal mapping in Lemma 1. The number of duplications in tube d is obtained by subtracting this number from k. The number of losses in r is n_1+n_2-2k, because each n_1+n_2 edges mapped in d_1 or d_2 either speciated or truncated in r. Each speciation event involves two edges, one

entering d_1, and another – d_2; losses correspond to truncated edges. According to the algorithm, the first item in (2) indeed equals to the total score of duplications in tube d and losses at the root branching r.

3. An case example of principal algorithm. Consider ten trees G_i in Fig. 3.

Figure 3. An artificial case. The supertree S^* and ten input gene trees.



The input trees were selected for the ease of inferring the supertree S^*. Here $V_0 = \{a,b,c,d,e\}$. Define P as a set of species containing all clades in all input trees. Define the loss score as 2, the duplication score as 3. With (2) recursively estimate the scores of all sets in P. The scores of singleton sets are null by definition. Estimate the scores of ten two-element sets V from P. Estimations for the set $V = \{x,y\}$ obtained with its single decomposition into $V_1 = \{x\}$ and $V_2 = \{y\}$ are given in Table 1.

Table 1. Scores of two-element sets. The first and forth columns contain the values of x and y, the third and sixth – the costs of $\{x,y\}$.

$V=\{x,y\}$	t	$c(V)$		$V=\{x,y\}$	t	$c(V)$
a,b	6	24		*b,d*	8	32
c,d	6	24		*a,e*	9	36
a,c	8	32		*b,e*	9	36
a,d	8	32		*c,e*	9	36
b,c	8	32		*d,e*	9	36

Here we detail the calculations. Consider the two cases when the current tree Gi does and does not contain the set V. In both cases $|Ed(V_1,G_i)|=|Ed(V_2,G_i)|=1$. In the first case $|Ed(V,G_i)| =2$, in the second case $|Ed(V,G_i)| =1$; it follows from the definition of set $Ed(V,G_i)$ or the algorithm of its construction described above. In the first case, the tree G_i does not contain vertices with one descendent edge belonging to $Ed(V_1,G_i)$ and another – to $Ed(V_2,G_i)$, while in the second case one such vertex exists. Thus, $l(V,V_1,V_2,G_i)$ and $d(V,V_1,V_2,G_i)$ equal 2 and 0 in the first case, and are null in the second case. Substitute $l(V,V_1,V_2,G_i)$ and $d(V,V_1,V_2,G_i)$ into (2). The column t of Table 1 contains the amount of gene trees without clade V for each set V. Then we obtain the value $c(V)$ indicated in the Table; the sets $\{a,b\}$ and $\{c,d\}$ have the minimal score of 24.

Define a decomposition of V that coincides with that in S^* as a standard decomposition, and others as non-standard. The algorithm does not rely on the tree S^* and enumerates all decompositions.

Now consider all three-element sets from P. Let us exemplify the procedure with $V = \{c,d,e\}$. Here the standard decomposition is $V_1 = \{c,d\}$ and $V_2 = \{e\}$. Compute the item $c(V,V_1,V_2)$ of function (2). For each gene tree G_i there can be: 1) $\{c,d\}$ is a clade in G_i, $\{c,d,e\}$ is not (two such trees); 2) $\{c,d,e\}$ is a clade in G_i, $\{c,d\}$ is not (two such trees); 3) $\{c,d\}$ and $\{c,d,e\}$ are not clades in G_i (four such trees); 4) $\{c,d\}$ and $\{c,d,e\}$ are clades in G_i (two such trees).

In cases 1) and 4) $|Ed(V_1,G_i)| = |Ed(V_2,G_i)| =1$, in cases 2) and 3) $|Ed(V_1,G_i)| =2$, $|Ed(V_2,G_i)| =1$. In case 1) $|Ed(V,G_i)| =2$, in cases 2) and 4) $|Ed(V,G_i)| =1$, in case 3) $|Ed(V,G_i)| =3$. In cases 1) and 3) the tree G_i does not contain vertices with one descendant belonging to $Ed(V_1,G_i)$, and another – to $Ed(V_2,G_i)$), in cases 2) and 4) one such vertex exists. Therefore, $l(V,V_1,V_2,G_i)$ and $d(V,V_1,V_2,G_i)$ equal 2 and 0 in case 1), 1 and 1 in case 2), 3 и 0 in case 3), and 0 and 0 in case 4). With equation (2), for a standard decomposition we obtain: $c(V,V_1,V_2) = 8 + 10 + 24 + 0 + c(\{c,d\}) + c(\{e\}) = 42 + 24 + 0 = 66$, because with induction $c(\{c,d\}) = 24$ and $c(\{e\}) = 0$.

Now consider a non-standard decomposition of the same set $V=\{c,d,e\}$ into $V_1 = \{c,e\}$ and $V_2 = \{d\}$, one of the two symmetric decompositions. Compute the value of $c(V,V_1,V_2)$. For each G_i there are possibilities: 1) $\{c,d\}$ is a clade in G_i, $\{c,d,e\}$ is not (two such trees); 2) $\{c,d,e\}$ is a clade in G_i, $\{c,e\}$ is not (three such trees); 3) none of the sets $\{c,d\}$, $\{c,e\}$, $\{d,e\}$ is not a clade in G_i (four such trees); 4) $\{c,e\}$ and $\{c,d,e\}$ are clades in G_i (one such tree).

In cases 1), 2) and 3) $Ed(V_1,G_i)| =2$, $|Ed(V_2,G_i)| =1$, in case 4) $|Ed(V_1,G_i)| =|Ed(V_2,G_i)| =1$. In case 1) $|Ed(V,G_i)| =2$, in cases 2) and 4) $|Ed(V,G_i)| =1$, in case 3) $|Ed(V,G_i)| =3$. In cases 1), 2) and 4) the tree G_i possesses one vertex with one descendant belonging to $Ed(V_1,G_i)$, and

another – to $Ed(V_2,G_i)$), in case 3) such vertices lack. Thus, $l(V,V_1,V_2,G_i)$ and $d(V,V_1,V_2,G_i)$ equal 1 and 0 in case 1), 1 and 1 in case 2), 3 и 0 in case 3), and 0 and 0 in case 4).

With equation (2), for this non-standard decomposition we obtain: $c(V,V_1,V_2) = 4 + 15 +24 +0 +c(\{c, e\}) +c(\{d\}) = 43 + 36 + 0 = 79$, because with induction $c(\{c, e\}) = 36$ and $c(\{d\}) = 0$. As the second non-standard decomposition has the same score, we conclude that all decompositions of V into two subsets from P are considered, and choose the minimal decomposition according to equation (2) (here – 66, a standard decomposition). Thus, the tree $(\{c, d, e\})$ coincides with the subtree in S*.

Analogously, $c(V_0,V_1,V_2)$ is computed for $V_0 = \{a,b,c,d,e\}$ and its standard decomposition into $V_1 = \{a,b\}$ and $V_2 = \{c,d,e\}$ (equals 128) and non-standard decompositions (the minimal value is 143). The output tree $S(V_0)$ coincides with the tree S* and has the score of 128.

4. **The auxiliary algorithm**. In task A2, our algorithm implements only a heuristic solution and infers the supertree as S' on the set $\{S(V)$: V is a basic set$\}$ (described below). It reconciles the set $S(V)$ into S'. No all clades from S' belong to P, therefore S' is not necessarily a solution in task B, as well as in task A2. Computer modeling suggests both $S' = S(V_0)$ and $S' \neq S(V_0)$ are possible, with the tree S' being biologically more relevant in the latter case.

The auxiliary algorithm: reconciliation of basic trees $S(V)$ into the tree S'. A variety of pertinent algorithms exist (ref., e.g., to [14], also for further references). However the algorithm complexity is not assessed in [14], it is obviously exponential. The efficiency strongly depends on the condition of consistency among the reconciled trees, e.g., in terms of Theorem 1b, which holds for basic trees.

Define the score of an arbitrary species tree S with the leaf set V against an arbitrary gene tree G as a score of mapping the tree G' obtained from G by pruning all leaves not contained in V; more precisely, by pruning subtrees with all leaves not contained in V. The score of the arbitrary species tree S is defined as a sum of scores of S against all trees in $\{S(V)$: V is a basic set$\}$.

At the initial step, by enumeration we find a three-leaf tree with a minimal score. It is used as a seed tree S.

At the induction step, all possible pairs $<s,e>$ are selected, where s is a species not present in the seed S, and e is an edge in S, including the root. For each pair $<s,e>$, a new edge is added to S connecting the midpoint in e with the leaf species s. Compute the score of the resulting tree $S(<s,e>)$ and choose a pair $<s',e'>$ with the minimal score. The seed S is extended into $S(<s',e'>)$, i.e. the new S becomes $S(<s',e'>)$. The induction continues over all eaves and constructs the final S'. The algorithm generally has a cubic complexity, which

becomes square in many cases, as computer modeling suggests. The end of the auxiliary algorithm.

The maximal number of minimal decompositions is trivially estimated as |P|. The output of the above described principal algorithm is based on one of the minimal decompositions for each V from P, and it, in principal, outputs not all basic trees $S(V)$. To obtain an exhaustive set of basic trees, the algorithm must enumerate all minimal decompositions at each induction step to produce $S(V)$. Such algorithm can be exponential, because its maximal complexity is trivially estimated as $|P|^{|V0|}$. In our study, several minimal decompositions occurred rarely (at maximum three), and this version of the algorithm required the same computing time as the principal algorithm.

To characterize trees from $\{S(V)$: V is a basic set$\}$, i.e. to provide an axiomatic definition of the basic tree, function 1* in task B is to be generalized by summing over all subtrees G' with root edges belonging to $Ed(V,G_i)$, and substituting V_0 with V.

We obtain the functional

$$C(V,S) = \Sigma_i \Sigma_{G'} [c_l \cdot l(f_{G'},G',S) + c_d \cdot d(f_{G'},G',S)], \qquad (3)$$

where V is a basic set from P.

Task B is then extended into task C. If $V = V_0$, then all G' from G_i coincide with G_i and function (3) is reduced to function (1), and task C – to task B. For any G' and S, variable $f_{G'}$ can be substituted in function (3) by the only (according to Lemma 1) scenario $h(G',S)$ for these G' and S.

Theorem 1A. Let P be a set of clades.
For any basic set V from P, the constructed basic tree $S(V)$ is a solution of task C. Vice versa, any solution of task B is $S(V)$, where V is a basic set from P, under the corresponding succession of minimal decompositions.

The proof of Theorem 1A is given in [7].

5. A computer program of building supertrees is available at http://lab6.iitp.ru/ru/super3gl/.

6. Case examples with biological data are available at http://lab6.iitp.ru/ru/super3gl/.