

RESEARCH

Open Access



GATC: a genetic algorithm for gene tree construction under the Duplication-Transfer-Loss model of evolution

Emmanuel Noutahi* and Nadia El-Mabrouk

From The Sixteenth Asia Pacific Bioinformatics Conference 2018
Yokohama, Japan. 15-17 January 2018

Abstract

Background: Several methods have been developed for the accurate reconstruction of gene trees. Some of them use reconciliation with a species tree to correct, *a posteriori*, errors in gene trees inferred from multiple sequence alignments. Unfortunately the best fit to sequence information can be lost during this process.

Results: We describe GATC, a new algorithm for reconstructing a binary gene tree with branch length. GATC returns optimal solutions according to a measure combining both tree likelihood (according to sequence evolution) and a reconciliation score under the Duplication-Transfer-Loss (DTL) model. It can either be used to construct a gene tree from scratch or to correct trees inferred by existing reconstruction method, making it highly flexible to various input data types. The method is based on a genetic algorithm acting on a population of trees at each step. It substantially increases the efficiency of the phylogeny space exploration, reducing the risk of falling into local minima, at a reasonable computational time. We have applied GATC to a dataset of simulated cyanobacterial phylogenies, as well as to an empirical dataset of three reference gene families, and showed that it is able to improve gene tree reconstructions compared with current state-of-the-art algorithms.

Conclusion: The proposed algorithm is able to accurately reconstruct gene trees and is highly suitable for the construction of reference trees. Our results also highlight the efficiency of multi-objective optimization algorithms for the gene tree reconstruction problem. GATC is available on Github at: <https://github.com/UdeM-LBIT/GATC>.

Keywords: Gene tree, Genetic algorithm, Phylogeny, Reconciliation, Gene duplication, Horizontal gene transfer

Background

Most biological discoveries can only be made in the light of evolution. In particular, functional annotation of genes is usually deduced from the orthology and paralogy relation between genes, which is inferred from the comparison of a gene tree with a species tree. Therefore, phylogenetic tree reconstruction is an important component of most bioinformatic pipelines. In this paper, we focus on the reconstruction of gene trees.

Standard phylogenetic tools are based on maximum likelihood (ML) or bayesian methods reconstructing a

tree from gene sequences (e.g. PhyML [1], RAxML [2], MrBayes [3], PhyloBayes [4]). However, for a variety of reasons due, not only to technical limitations but also to the data itself (sequences too close to each other or conversely too divergent), sequence-only methods often do not allow to confidently discriminate one tree from another.

To address this limitation, more recent gene tree reconstruction methods, designated here as *integrative methods*, include information from the species tree. The idea is to consider, in addition to a maximum likelihood value measuring the fitness of a tree to a sequence alignment (according to a model of sequence evolution), a measure reflecting the evolution of a whole gene family

*Correspondence: fmr.noutahi@umontreal.ca

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal, Canada

through gene gain and loss. A standard measure of fitness between a gene tree and a species tree is computed in terms of a “reconciliation” score. In a probabilistic framework, the reconciliation score corresponds to the probability density of the gene tree given some parameters (rates of events and branch lengths) under a birth-death and gain model of evolution. For the Most Parsimonious Reconciliation model (MPR), this score corresponds to the optimal number of gene gain and loss events, weighted by their costs, explaining the incongruence between a gene tree and a species tree.

Most integrative methods for gene tree reconstruction assume a simplified model of gene family evolution where gene gain events are reduced to gene duplication (e.g. TreeBeST [5], TreeFix [6], ProfileNJ [7], NOTUNG [8], SPIMAP [9], Giga [10]). In fact, the MPR problem for the Duplication-Loss (hereafter denoted DL) model of gene family evolution is linear-time solvable [11]. By introducing horizontal gene transfer (HGT) events, the Duplication-Transfer-Loss (DTL) model becomes NP-hard in general if time consistency is required for inferred events (unless the species tree is fully dated) [12–14]. However the MPR problem for the DTL model, with an undated species tree, can still be computed in polynomial time if the time consistency requirement is relaxed [15–17]. Due to this reasonable time-complexity, some recent phylogenetic softwares have extended the gene family evolution model to transfers (MowgliNNI [18], ecceTERA [19], TreeFix-DTL [20]). Continuous effort is also made for developing fast probabilistic frameworks capturing HGT events (see [21] for a review of these models).

Integrative methods report gene trees with better accuracy compared with sequence-only methods [18, 20, 22, 23], but they still leave space for improvement, both on tree quality and on computation time. In fact, most of them rely on a two-steps approach, first computing a tree with the best fit to the sequences, and then exploring a tree space surrounding the initial tree to select one minimizing the considered reconciliation distance. From an accuracy point of view, this two step methodology does not guarantee that the output tree optimizes both the likelihood given the sequence alignment, and the reconciliation measure, as the best fit to the sequences may be lost at the second step. In addition, by considering a single tree at a time, the risk of ignoring a large part of the tree space and falling into a local minimum is high. Other integrative methods (see for example PhylDog [24] and PrIME-DLTRS [23]) compute the joint likelihood associated with a substitution model and DTL event rates, given a fixed, dated and ultrametric species tree and a gene family alignment. They use tree exploration heuristics similar to those found in sequence-only programs for phylogenetic tree reconstruction to explore

the solution space, often in a bayesian-MCMC framework. These methods come at a high computational cost, especially when HGT events are considered, and they are still subject to the risk of being stuck in a local optimum.

In this paper, we present *GATC* (Genetic Algorithm for gene Tree Construction), a new software for gene tree reconstruction under the DTL model that can take as input completely unresolved, partially unresolved or fully resolved trees, and outputs a tree minimizing a measure combining both tree likelihood (according to sequence evolution) and a reconciliation score. In other words, it can either be used as a two-step correction method, when input trees are the output of other phylogenetic methods, or as a one-step method resolving a full polytomy (star tree) in a way optimizing fit to both the species tree and the sequences.

With *GATC*, we explore a new methodological framework based on a Genetic Algorithm (GA), a global search metaheuristic that mimics biological evolution [25]. The ability of GAs to find near-optimal solutions quickly, even for complex models and data makes them suitable for the problem of phylogenetic inference. In fact, the GA methodology has been previously applied to the phylogenetic inference problem, starting with Matsuda in 1996 [26] using a maximum likelihood criterion, Lewis [27] who introduced a subtree swap crossover operator, and other more recent algorithms (e.g. self-adaptive GA [28], Ga-mt [29], METAPIGA [30], GARLI [31]). However, all these algorithms are solely based on sequence information and, as discussed above, are often error-prone in the case of gene tree reconstruction. To our best knowledge, GAs have never been applied to species tree-aware gene tree reconstruction, although the technique is suitable to the resolution of Multi-Objective Optimization Problems (MOOP).

To measure the performance of *GATC*, we compared it to current state-of-the-art softwares on a dataset of simulated cyanobacterial phylogenies. Our results show that *GATC* is more accurate than existing methods, suggesting that it substantially increases the efficiency of the phylogeny space exploration. We also evaluated *GATC*'s ability to infer accurate homology relationships between genes on a standardized, manually curated, dataset of real trees. The predicted relationships were mostly in agreement with the ones inferred from a reference tree, highlighting the efficiency of the framework.

Methods

Notation on trees

All considered trees are rooted unless explicitly stated. A tree is *binary* if all its internal nodes have exactly two children, and *non-binary* otherwise. Unless stated differently, all trees are considered binary.

We denote by $V(T)$ the nodeset, by $E(T)$ the edgeset, by $\mathcal{L}(T)$ the leafset and by $r(T)$ the root of a tree T . An edge e of $E(T)$ is written as a pair (x, y) of two adjacent nodes where e is an outgoing edge of x . For $e = (x, y)$, x is the parent $p(y)$ of y , while y is a child of x . A node x is an *ancestor* of y , which is denoted $x <_T y$, if it is on the path from y to the root (excluding y). In this case, y is called a *descendant* of x . Similarly, an edge $e' = (x', y')$ is an ancestor of an edge $e = (x, y)$ if it is on the path from y to the root. Given a node x , $T[x]$ is the subtree of T rooted at x and $\mathcal{L}(x)$ the leafset of $T[x]$. Two subtrees $T[x]$ and $T[y]$ are *separated* in T if $x \neq y$, $x \not<_T y$ and $x \not>_T y$. In this case, $\mathcal{L}(x) \cap \mathcal{L}(y) = \emptyset$, if the leaves of T are uniquely labeled by the elements of $\mathcal{L}(T)$.

A *species tree* is a tree S with leaves uniquely labeled and $\mathcal{L}(S)$ being a set of species. Likewise, a *gene tree* is a tree G with leaves uniquely labeled and $\mathcal{L}(G)$ corresponding to a set of genes where each gene g belong to a genome $s(g)$. We denote by \bar{G} the tree obtained from G by replacing each leaf label g_i by its genome $s(g_i)$. Notice that the mapping $s : \mathcal{L}(G) \rightarrow \mathcal{L}(S)$ does not have to be injective nor surjective. In particular, \bar{G} may have several equally labeled leaves.

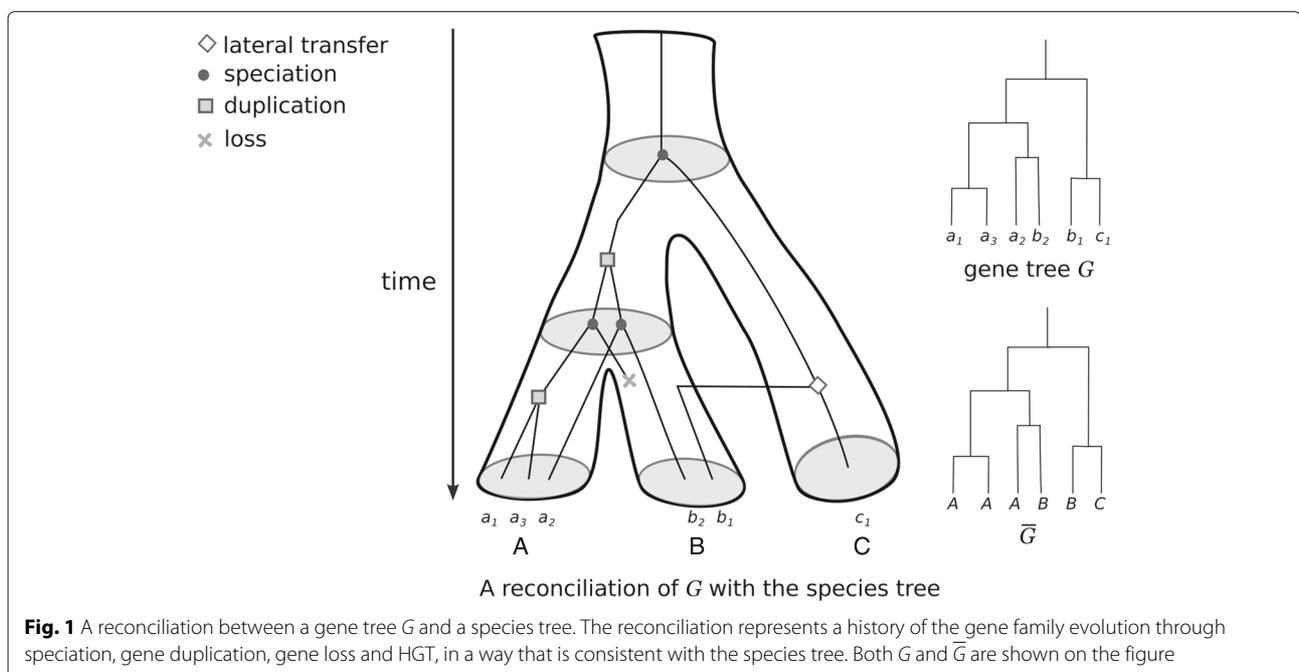
A *reconciliation* of G (or similarly \bar{G}) with S (see Fig. 1) is an extension of s from $V(G)$ to $V(S)$ with additional labels on each internal node x of G , describing the type of evolutionary event that has led to $G[x]$ (duplication, speciation or transfer). G can be expanded to include lost genes.

Finally, we refer to the process of removing a leaf l and its associated edge $(p(l), l)$ from a tree T as the *deletion of l from T* .

Vocabulary of Genetic Algorithms

A Genetic Algorithm (GA) is an algorithmic framework mimicking biological evolution. The vocabulary of a GA is filled with biological metaphors. It begins with a *population* of *individuals* whose *chromosomes* or *genomes* encode specific solutions to the problem of interest. Performance of these individuals in solving the problem is measured by their *fitness score*. To avoid confusion, throughout this paper the word “chromosome” will be used solely to designate the data structure of a genetic algorithm, and the word “genome” will be used in its biological meaning to designate the macromolecules containing the genes under study.

At each step, starting from an initial population, a new population is generated using three operators: *selection*, *crossover* and *mutation* [25], which are defined according to the nature of the problem and the encoding of the solution. During selection, the fitness score is used to select individuals for reproduction. Selected candidates are combined using the crossover operator to create new individuals that are then modified by the mutation operator in order to introduce diversity and avoid local optima. With *elitism*, the less fit individuals of the newly obtained population are replaced by the best fit of the previous *generation*, in order to conserve the best solutions found so far. The process described above is repeated through multiple



generations, until an optimal solution (chromosome of the best individual) is obtained or a stop criterion is reached.

This natural selection process generally leads to the improvement of the average population fitness over generations. While GAs often converge to an optimal or near optimal solution, their performance mainly depends on the mechanism for balancing two potentially contradictory objectives: keeping the best solutions found so far, and at the same time efficiently exploring the search space for promising solutions.

The GATC algorithm description

In the rest of the manuscript, we will loosely refer to the tree likelihood given a multiple sequence alignment as *sequence likelihood*.

Given a sequence alignment D and a species tree S , our objective is to find the gene tree G or a set \mathcal{G} of gene trees, with branch length, that are (near) optimal for both the sequence likelihood and the reconciliation score. To solve this problem, our fitness function should reflect both objectives. We will present different ways for computing the fitness score, either by a linear combination of the two scores, or by trying to reach a *pareto optimality*. We start by presenting the general framework of the GA.

Solution encoding

A chromosome σ is defined as (G, θ) where G is a rooted binary gene tree and θ is the set of hyperparameters underlying the evolutionary model. Namely, $\theta = (\lambda, \delta, \tau, e, l, m)$, representing respectively the duplication rate, the loss rate, the transfer rate, the substitution rates across the gene tree edges, branch lengths and the substitution model. Some of these parameters might be kept fixed during the evolutionary process. For example, the substitution model m is usually fixed for all generations, whereas duplication, loss and transfer rates can vary when a probabilistic model is used to compute the reconciliation score. When parsimony is preferred, they correspond to fixed event cost. Branch lengths and edge substitution rates are usually optimized during sequence likelihood computation.

For the probabilistic model of reconciliation, the initial values of the hyperparameters λ, δ, τ and e are randomly drawn from a uniform distribution, unless explicitly provided. The default substitution model used for nucleotide sequences is the GTR model [32] with a gamma distribution to account for rate variation [33], whereas for proteins, the JTT model [34] with gamma-distributed rates is used.

Gene trees in the initial population

When starting trees are available (from any other tree construction method, integrative or not), they can be

used as the population of the first generation in our GA. Otherwise the trees of the initial population are generated, either randomly or according to a predefined procedure. GATC implementation allows generating the initial trees from a star tree, using PolytoMySolver [35] which outputs the most parsimonious trees for the DL-reconciliation score (but not necessary optimal for the DTL-reconciliation score or the sequence likelihood), or using bootstrapped trees obtained with RAxML [2]. These two methods should be preferred to the initialization with random trees, which can affect the algorithm's convergence.

Notice that two trees G_1, G_2 such that $\overline{G_1} = \overline{G_2}$ have the same reconciliation score, and thus if G_1 is a solution of PolytoMySolver, minimizing the DL-reconciliation score, then G_2 is also a solution. Therefore, in this case, to increase the initial population of the GA, additional trees can be obtained by permutation of the genes at the leaves of G_1 in a way respecting the mapping function s .

Computing the sequence likelihood and reconciliation score

To evaluate the fitness of each chromosome $\sigma_i, 1 < i < n$, in a population of size n , we first compute a vector \vec{z}_i of two components, called the *raw score vector*, containing the sequence likelihood and the reconciliation score. Note that when the objective is to optimize only sequence likelihood, the second component corresponding to the reconciliation score is set to zero.

The sequence likelihood scores $p(D|G, l, m)$ can be computed using the Felsenstein algorithm [36] and the further computational enhancement described by Stamatakis et al. (2004) [37]. In fact, GATC use subroutines from RAxML to compute the sequence likelihood, thus benefiting from both its high computational speed and its large set of substitution models.

As for the reconciliation score, it can be computed under either the probabilistic or MPR model. For the MPR scoring model, we implemented the Bansal algorithm [15] which computes the DTL reconciliation cost between a binary gene tree G and a binary species tree S in time $O(|G||S|)$. Notice that, as explicit transfer pathways are not specified, a DTL scenario is not necessarily possible as it may violate temporal constraints [14]. In fact, a donating and a receiving species must have co-existed at the time of the transfer. Moreover, in contrast to duplications and losses, HGT are inter-dependent, and can induce contradictory temporal constraints on ancestral species. However, as the reconciliation problem for DTL using undated species tree with the constraint of respecting temporal constraints is NP-hard, the Bansal algorithm remains a good alternative for computing a reasonable DTL reconciliation score. In absence of HGTs, we compute the DL reconciliation score using a linear-time algorithm [11], to speed up calculations.

For the probabilistic scoring model, we have implemented the DTL model first described by Tofigh [38, 39] and used by PrIME-DLRS [23]. It is based on a birth-death model of evolution including rates for gene duplication, transfer and loss that requires discretization of a dated species tree and numerical resolution of ordinary differential equations. We refer the readers to the Supplementary Material of [23] for a thorough description of how the probability density of the reconciliation is computed.

Rather than minimizing the reconciliation score and maximizing likelihood, it is easier to simultaneously minimize both measures. For this reason, we take the negative log value when likelihood is used for any of the two scores. Therefore, it has to be understood that the best adapted individuals will be those with the lowest fitness.

Computing the fitness score

Given a raw score vector \vec{z}_i for a chromosome σ_i , a weight vector \vec{w} and a scaling function ϕ , we define the fitness score f_i of σ_i as $f_i = \vec{w} \cdot \phi(\vec{z}_i)$. In other words, f_i corresponds to the weighted sum of the two components of the raw score vector, scaled by a function ϕ . The simplest definition of ϕ is the identity function $\phi(\vec{z}) = \vec{z}$. An alternative is to standardize each score to a zero-minimum resulting in the following formulation: $\phi(z_i^k) = z_i^k - \min_i(z_i^k)$ for $1 \leq k \leq 2$ and $1 \leq i \leq n$. However, for this latter scaling function, fitness is not comparable between individuals of different generations.

Using the method described above for computing f_i transforms our problem into a single objective minimization problem and is suitable when both components of z_i are log likelihood values, since it is related to the joint weighted probability density for sequences data and reconciliation to the species tree.

When the reconciliation score is computed using parsimony, combining the two scores this way might not be optimal. Instead, we compute a set of *pareto optimal solutions* for this multi-objective optimization problem (MOOP). Several evolutionary based techniques have been developed for MOOP [40]. Here we will use a technique similar to the widely known NSGA (Non-dominated Sorting Genetic Algorithm) [41].

A raw score vector $\vec{z}_i = (z_i^1, z_i^2)$ is said to *dominate* another vector $\vec{z}_j = (z_j^1, z_j^2)$, denoted as $\vec{z}_i < \vec{z}_j$, iff $\vec{z}_i \neq \vec{z}_j$ and $z_i^1 < z_j^1, z_i^2 < z_j^2$. We are interested in finding the set of non-dominated solutions called *pareto set* (PS) and denoted as:

$$PS = \{\sigma_i \mid \nexists \sigma_j, \vec{z}_i < \vec{z}_j\}$$

At the end of the GA's evolutionary process, the pareto set represents the set of pareto optimal solutions. In contrast with classical genetic algorithms, computing the

pareto set requires to consider simultaneously a parent population P_i and its offspring P'_i , as optimal solutions from P_i can be lost if we use P'_i as the population P_{i+1} of the next generation.

Algorithm 1 illustrates the way fitness is computed for all individuals of a generation. It proceeds in a *wave* fashion, selecting the non-dominated individuals from the population $P^* = P_i \cup P'_i$, assigning them a shared fitness score, and then removing them from P^* . This process is repeated while increasing the fitness score for the individuals in the new waves, until the expected population size per generation is met or there is no non-dominated individuals anymore. In the latter case, the fitness of the remaining individuals is computed as the sum of their *dominance rank* (number of individuals that dominates an individual) and the fitness of the last wave. This process ensures that individuals belonging to the same wave have the same fitness and as such the same probability to reproduce. The n individuals with the best fitness constitutes P_{i+1} . Selection, crossover and mutation operators can be applied to P_{i+1} resulting in offspring P'_{i+1} .

Algorithm 1 Compute next generation population P_{k+1} from P_k

```

1: procedure COMPUTENEXTPOP( $P_k$ )
2:   Compute  $P'_k$ , the offspring population of  $P_k$ 
3:    $P_k^* \leftarrow P_k \cup P'_k$ 
4:   Evaluate  $z_i$  for all  $\sigma_i \in P_k^*$ 
5:   Compute the dominance rank  $d_i$  for each  $\sigma_i \in P_k^*$ 
6:    $w \leftarrow 1$ 
7:   while  $\exists \sigma_i \in P_k^* \mid d_i = 0$  do
8:      $Wave_w \leftarrow \{\sigma_i \mid d_i = 0\}$ 
9:     Set a shared fitness for all  $\sigma_i \in Wave_w$  as  $w$ 
10:     $P_k^* \leftarrow P_k^* \setminus Wave_w$ 
11:    Compute the dominance rank  $d_i$  for each  $\sigma_i \in P_k^*$ 
12:     $w \leftarrow w + 1$ 
13:  end while
14:  for  $\sigma_i \in P_k^*$  do
15:    set the fitness of  $\sigma_i$  as  $w + d_i$ 
16:  end for
17:   $P_{k+1} \leftarrow \bigcup_w Wave_w \cup P_k^*$ 
18:  return the first  $|P_k|$  of  $P_{k+1}$  according to fitness
19: end procedure

```

Selection

GATC implements multiple classical selection methods. Individuals can either be selected for crossover using the tournament selector [42] or using the roulette wheel selector which chooses individuals with probability inversely proportional to their fitness values (recall that the best individuals have the smallest fitness value). Alternatively,

the random uniform selector can be used, which gives equal reproduction probability to all individuals regardless of their fitness. Selected individuals are used in the crossover operator to produce the individuals of the next generation.

Crossover

In the crossover operators implemented in GATC, two offsprings are created from two parent chromosomes. Each offspring inherits its hyperparameter θ from one of its parents, while its gene tree is obtained from the combination of the two parental trees.

Given trees G_i and G_j respectively from parent σ_i and σ_j , the first offspring is obtained with the subtree swap crossover operator [27], achieved by the following actions:

- 1 Select a subtree $G_i[x]$ from G_i (the root is excluded)
- 2 Delete all leaves from G_j that are also in $\mathcal{L}(x)$;
- 3 Regraft $G_i[x]$ to a random edge of G_j to obtain the offspring tree G'_j .

The second offspring tree G'_i , is obtained in a similar way by selecting a subtree from G_j and regrafting it in G_i . The crossover operator is illustrated on Fig. 2a.

In the special case where the objective is to only optimize the sequence likelihood, under the hypothesis that the reconciliation score is already optimal, this crossover operator is not applicable as it does not preserve the reconciliation score. Instead, the offspring trees are created by exchanging two subtrees $G_1[x]$ and $G_2[y]$ such that $\overline{G_1[x]}$ and $\overline{G_2[y]}$ are isomorphic with respect to the labels at their leaves (see Fig. 2b).

Mutation

For a chromosome $\sigma_i = (G_i, \theta_i)$, a mutation is performed either on the tree G_i or on the rates $\lambda, \delta, \tau, \epsilon$ unless their values are fixed. Mutations on the rate parameters

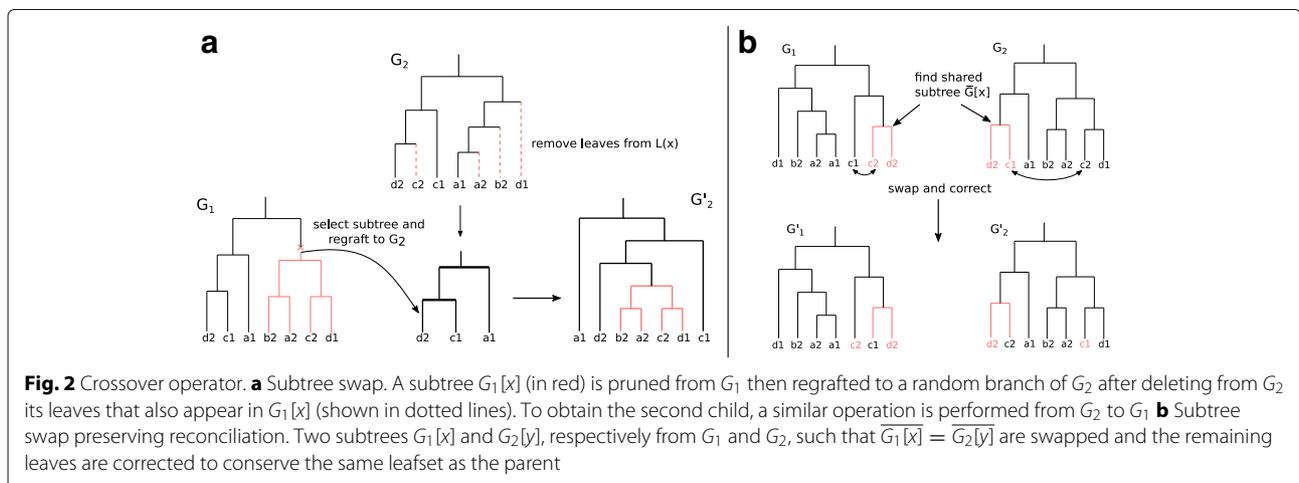
consist in drawing a new value from their distribution. On the other hand, a mutation operates on G_i by applying a topological modification. GATC uses SPR (Subtree Pruning and Regrafting) and re-rooting operations (see Fig. 3a-b) to generate a new tree topology. As with the crossover operator, when only sequence likelihood has to be optimized, reconciliation score should be preserved after mutations. For this purpose, mutation are performed by permuting the genes assignment to the leaves of G_i in such a way that only genes belonging to the same species are allowed to switch places (see Fig. 3c).

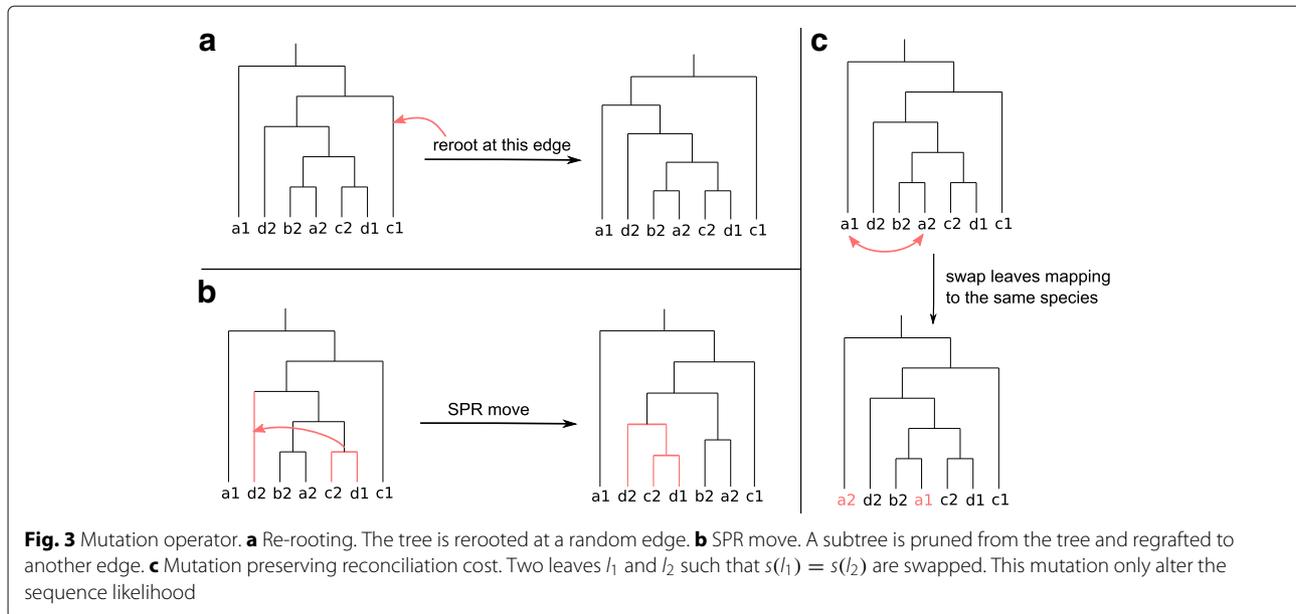
Stop criteria

We proposed several criteria to stop the GA evolution. The simplest ones are to terminate when a maximum number of generations or a time limit are met, or when all individuals converge to a single fitness value. Aside from these criteria, we propose another simple termination criterion called *population-AU criterion* that is based on the use of a reference ML tree. Under this criterion, evolution is stopped when all the individuals in the current population are statistically equivalent to the known ML tree, according to the AU test [43]. This stop criterion allows for a good performance when the objective is restricted to the optimization of sequence likelihood.

Results and discussion

To measure the efficiency of GATC in reconstructing accurate gene trees, we compared its performance, on a simulated dataset, to four different gene tree reconstruction methods: ecceTERA [19], TreeFix-DTL [20], ProfileNJ [7], MowgliNNI [18] and RAxML [2]. In contrast to RAxML which is a sequence-only method, the former four methods use both sequences and species tree information. We also used GATC to reconstruct the gene trees of three gene families for which reference trees have been proposed [44]. We will entirely focus on evaluating





GATC's performance under the MPR model, as it is our main contribution and also because DTL-reconciliation scores can be computed significantly faster under this model.

Evaluation on a simulated Cyanobacteria histories dataset

We used the simulated cyanobacteria dataset of Szöllösi et al. (2013) [22] publicly available at <http://datadryad.org/resource/doi:10.5061/dryad.pv6df>. This dataset consists of 1099 gene families from 39 cyanobacteria species along with a well-resolved dated species tree. To construct the dataset, the gene families were retrieved from HOGENOM [45] and multiple alignments were performed on these families with Muscle [46]. For each alignment, an MCMC sample of at least 3000 trees was obtained with PhyloBayes [47] and used to reconstruct an amalgamated tree with ALE [22]. These trees were used to simulate new multiple alignments of artificial sequences under the LG model with a gamma distribution. We refer to [22] for a more detailed description on the construction of the dataset.

From each of the 1099 simulated artificial sequence alignments, we reconstructed an initial tree using RAxML (LG + Gamma, 100 bootstraps). The RAxML trees (with bootstrap values) were used as input for all programs being compared against GATC.

For all programs, we used fixed DTL rates ($\lambda = 2$, $\tau = 3$, $\delta = 1$) except for ProfileNJ which supports only a DL model of reconciliation and for which we took $\tau = \infty$. We ran TreeFix-DTL with default parameters and LG + Gamma as model of evolution. As it requires rooted trees, the input RAxML trees were rooted using the midpoint rooting method [48]. MowgliNNI, ecceTERA and

ProfileNJ were run with a threshold of 0.7 for the contraction of for weak edges. Note that ProfileNJ and ecceTERA can output several solutions from which users can later select a tree according to some other measure (sequence likelihood for example). In our comparison, we only consider the first solution returned by both methods, as this selection process is not part of the methods, and also removes their running time advantage. We ran GATC with the following parameters : a maximum of 50 generations, a time limit of 90 minutes per gene family, LG + Gamma as the model of evolution and parsimony for DTL-reconciliation. We used the tournament selector as the selection operator and set the crossover and mutation rates to 0.8 and 0.5 respectively (see Additional file 1: Section 1 and Figure S1, for a discussion on the effect of crossover and mutation rates on accuracy). To construct the initial population of the GA, we used PolytoMySolver's resolutions of RAxML trees after contraction of edges with support less than 0.7. In order to keep the GA population size fixed at 30, we randomly removed or duplicated chromosomes from the initial population until its size became 30. We also used the population-AU as additional stopping criterion with the RAxML tree being the known best ML tree and a significance level $\alpha = 0.05$. When there were more than one tree in the pareto optimal set, the tree with the lowest DTL-reconciliation score was returned as GATC final solution.

We measured the accuracy, defined as the normalized Robinson-Foulds distance between each reconstructed tree and the true tree. As shown in Fig. 4, trees reconstructed with species tree-aware algorithms were more accurate than RAxML's trees. This result was expected, since it has been shown several times that integration

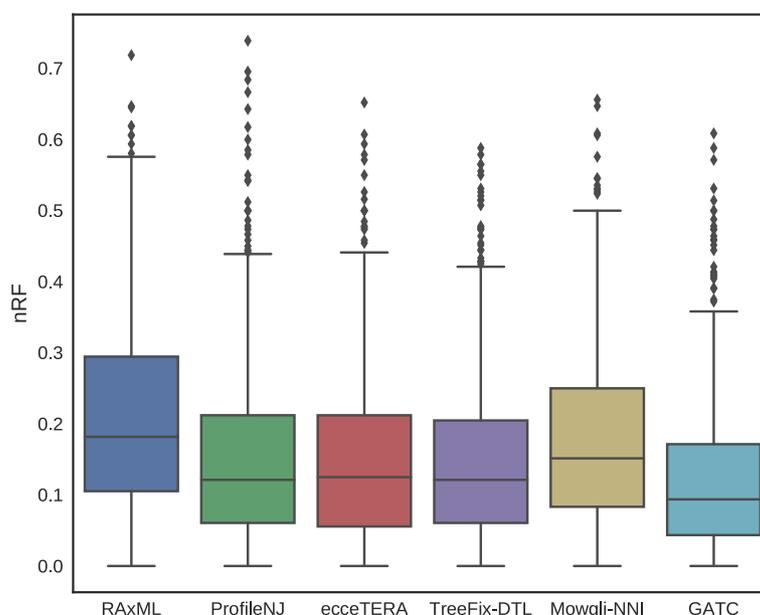


Fig. 4 Accuracy of RAxML, ProfileNJ, ecceTERA, TreeFix-DTL, Mowgli and GATC on a dataset of simulated Cyanobacteria histories: we measure the normalized Robinson-Foulds distance of the reconstructed trees to the true gene trees for all 1099 gene families. GATC achieves the best accuracy on the simulated dataset, followed by TreeFix-DTL

of species tree information usually improves gene trees reconstruction. GATC, in particular, achieves a better accuracy than other methods, due to its improved tree space search efficiency. The algorithm also appears to be robust, in some extent, to errors in the species tree topology (see Additional file 1: Section 2 and Figure S2). However, it should be noted that in order to obtain accurate results, there is a need to allocate a considerable time for the evolution of the GA. As such, the algorithm is much slower, in comparison to ProfileNJ and ecceTERA which can output solutions in a few seconds. To our surprise, ProfileNJ was almost as accurate as the second best method (TreeFix-DTL), although it only supports a DL model of reconciliation and HGT were present in the dataset. It is possible that most edges with weak support were not involved in HGT events, which can explain the observed performance of ProfileNJ.

Evaluation on an empirical dataset

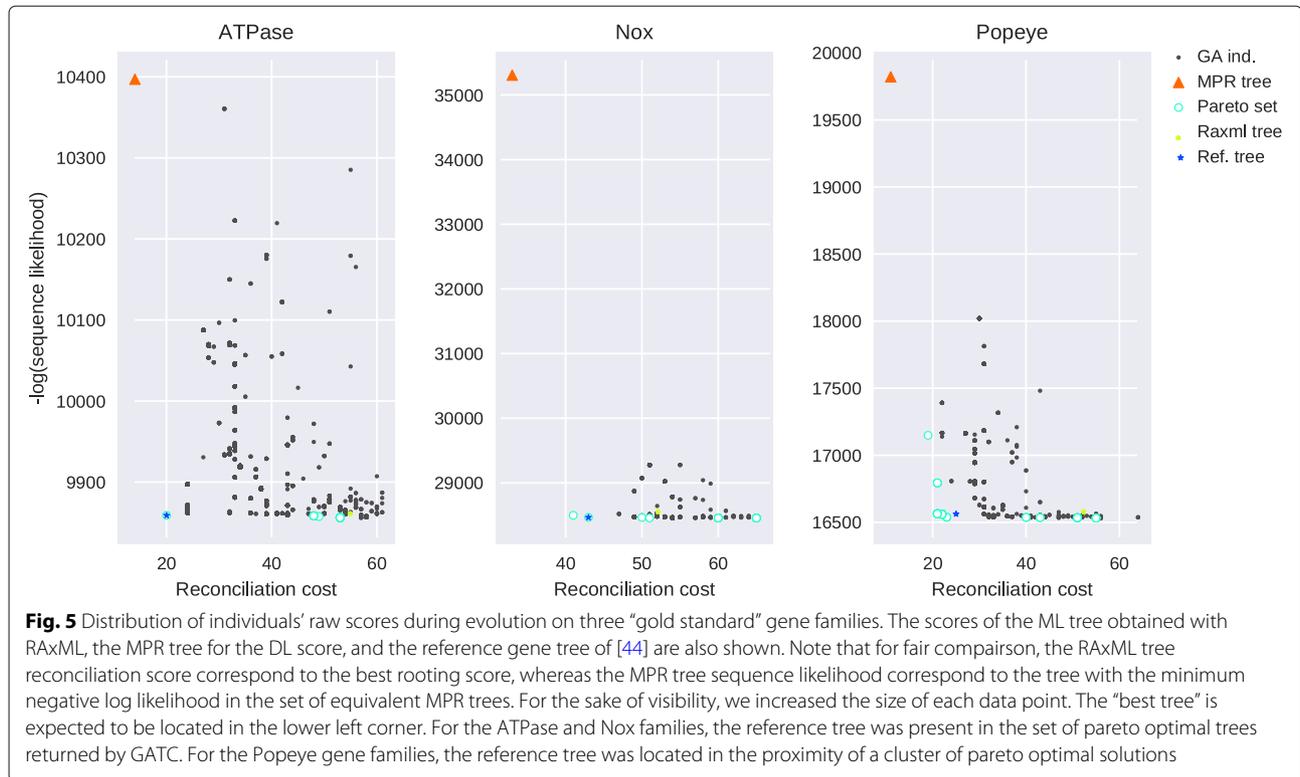
In an attempt to establish a benchmark for comparing orthology prediction methods, Boeckmann et al. (2011) [44] proposed manually curated “gold standard” gene trees for three well-conserved gene families: the Popeye-domain containing family (POP), the NOX ‘ancestral-type’ subfamily of NADPH oxidases (NOX) and the V-type ATPase beta subunit (VATP).

These gene families have been re-analyzed here to assess the performance of GATC on an empirical dataset. The reference species tree used was obtained from SwissTree

[49]. Protein sequences from genomes not found in the species tree were removed and the remaining sequences aligned with Muscle [46]. GATC was used to reconstruct the corresponding trees for each gene family with initial population of trees obtained from bootstrap replicates. We used the same parameters as above except for the DTL events cost, which was changed to: ($\lambda = 1$, $\tau = \infty$, $\delta = 1$). Here, we prohibit HGT events since they are not expected in the dataset. We also set the maximum number of generations to 300 and the maximum time of evolution to 3h per gene family. For comparison, the average time needed by RAxML to obtain the best ML trees is 2.4h.

In order to measure the accuracy of GATC, we investigated how close the reference trees were to the set of pareto optimal trees. Figure 5 shows the distribution of individuals’ scores, over generations, during the GA evolution for each gene family. We were able to retrieve the reference tree for the NOX and VATP gene families, whereas the reference tree for the POP family was located close to a cluster of pareto optimal trees. From the same figure, it can also be seen that even though the ML and MPR trees theoretically belong to the pareto optimal set of the complete tree space, they are often located far from the desired optimal result.

Since the reference gene tree was not obtained for the POP family, we report the precision and recall of orthologous and paralogous genes inference from the solutions returned by GATC, compared to the proposed reference



tree (Table 1). Note that GATC only outputs ten trees from the 30 individuals of the final population resulting in four unique trees (see Additional file 1: Figures S4-S7). We computed precision and recall for the two types of gene relationships as follows:

$$Precision = \frac{TP}{TP + FP} , Recall = \frac{TP}{TP + FN}$$

where TP corresponds to the number of shared pairs of orthologs/paralogs with the reference tree, FP corresponds to the number of predicted pairs of orthologs/paralogs not present in the reference tree, and FN to the number of missed orthologs/paralogs. As shown on Table 1, the precision and recall for the inferred gene relationships were high for all four solutions. Difference between GATC's solutions and the reference POP gene family tree (Additional file 1: Figure S3) can mostly be explained by the fact that duplication nodes were often

placed lower in the solutions, resulting in fewer number of losses and consequently lower reconciliation scores.

It is hard to argue whether the proposed reference tree represents the true evolutionary history of the gene family over our pareto optimal solutions. In fact, from Fig. 5, it can be seen that some pareto optimal solutions were better than the reference POP gene tree for both scores, suggesting that they could be of higher quality. As the true evolutionary histories of gene families are hardly known, relying on high-quality phylogenetic gene trees for biological analyses is preferable.

In summary, our results on the empirical dataset demonstrate how a GA framework can be used for the inference of gene trees with high accuracy.

Conclusion

Algorithms for constructing gene trees from multiple sequence alignments are widely used. However when a

Table 1 Comparison between the reference tree of the Popeye family and the pareto optimal trees returned by GATC

	NormRF distance	Orthologs		Paralogs	
		Prec.	Rec.	Prec.	Rec.
Tree 1	0.260	0.763	0.942	0.971	0.871
Tree 2	0.260	0.765	0.941	0.971	0.873
Tree 3	0.087	0.902	0.983	0.992	0.894
Tree 4	0.109	0.829	0.866	0.940	0.922

reliable species tree is available, it is preferable to use species tree-aware methods which are often more accurate. In this work, we have presented a GA framework for the reconstruction of gene trees using both sequences and species tree information. From the comparison with existing methods, we have shown that this framework, implemented in a software called GATC, outputs more accurate gene trees.

As the true evolutionary history of a gene family does not always correspond to the most parsimonious one, GATC assumes instead that the true gene tree can most likely be found in the pareto optimal set of the search space. Therefore, given enough time, the algorithm will converge to a set of candidates containing that tree. Although this hypothesis was supported by our results on the empirical dataset, it does not necessarily hold for all gene families. For example, since our reconciliation model does not consider Incomplete Lineage Sorting (ILS), the efficiency of GATC is expected to decrease in presence of ILS. Indeed, signals of deep coalescence leading to incongruence between species and gene tree would be explained by DTL events, possibly resulting in incorrect trees. Moreover, another problem still persists when there are several trees in the final pareto set, as alternative criteria for discriminating between these equivalent candidates are required. In its current implementation, GATC outputs solutions sorted by either the sequence likelihood or the reconciliation score.

Despite the good results we obtained by using GATC, one fundamental aspect that should be addressed in order to improve efficiency is the required evolution time. Indeed, running time cannot be accurately estimated especially when the starting trees have poor quality. When ML or bayesian trees have been inferred beforehand, it may be appropriate to set the maximum evolution time to the time required to find the best ML tree. As the underlying idea behind GAs allows for easy parallelism, running time can be dramatically reduced. Balance between scalability to large datasets and search efficiency would likely be achieved by carefully selecting the different genetic operators and the stopping criteria. Finally, to avoid being trapped in local optima, multiple replicate searches, using different settings (such as DTL, crossover and mutations rates, population size and initialization) can be performed in parallel with exchange of information through a migration operator.

Additional file

Additional file 1: Contains supplementary information on the effect of operator rates (**Figure S1**) and errors in the species tree (**Figure S2**) on reconstruction accuracy. It also contains the original reference tree of the Poyeye family (**Figure S3**) and the four alternative trees obtained by GATC (**Figure S4-S7**). (PDF 315 kb)

Abbreviations

AU: Approximately unbiased test; DL: Duplication-loss; DTL: Duplication-Transfer-Loss; FN: False negative; FP: False positive; GA: Genetic algorithm; GATC: Genetic algorithm for gene tree construction; HGT: Horizontal gene transfer; ILS: Incomplete lineage sorting; JTT: Jones, Taylor and Thornton substitution matrix; LG: Le and Gascuel substitution matrix; ML: Maximum likelihood; MOOP: Multi-objective optimization problem; MPR: Most parsimonious reconciliation; NSGA: Non-dominated sorting genetic algorithm; SPR: Subtree pruning and regrafting; TP: True positive

Acknowledgments

Computations were made on the supercomputer Briarée from Université de Montréal managed by Calcul Québec and Compute Canada. The authors also thank the three anonymous reviewers for their insightful comments and constructive suggestions that helped improve the manuscript.

Funding

Publication charges for this work was funded by The Natural Sciences and Engineering Research Council of Canada (NSERC).

Availability of data and materials

GATC is available for download at <https://github.com/UdeM-LBIT/GATC>. The simulated cyanobacteria dataset analysed during the current study is publicly available at <http://datadryad.org/resource/doi:10.5061/dryad.pv6df>. The empirical dataset of reference gene trees is available at https://swisstrree.vital-it.ch/gold_standard. Full outputs of all programs are available from the corresponding author on reasonable request.

About this supplement

This article has been published as part of *BMC Genomics* Volume 19 Supplement 2, 2018: Selected articles from the 16th Asia Pacific Bioinformatics Conference (APBC 2018): genomics. The full contents of the supplement are available online at <https://bmcgenomics.biomedcentral.com/articles/supplements/volume-19-supplement-2>.

Authors' contributions

EN and NEM designed the research and wrote the manuscript. EN implemented GATC, performed the experiments and analysed the results. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Published: 9 May 2018

References

- Guindon S, Gascuel O. A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol.* 2003;52:696–704.
- Stamatakis A. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analysis with thousands of taxa and mixed models. *Bioinformatics.* 2006;22:2688–90.
- Ronquist F, Huelsenbeck JP. MrBayes3: Bayesian phylogenetic inference under mixed models. *Bioinformatics.* 2003;19:1572–4.
- Lartillot N, Philippe H. A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Mol Biol Evol.* 2004;21(6):1095–109. <https://doi.org/10.1093/molbev/msh112>.
- Schreiber F, Patricio M, Muffato M, Pignatelli M, Bateman A. Treefam v9: a new website, more species and orthology-on-the-fly. *Nucleic Acids Res.* 2013. <https://doi.org/10.1093/nar/gkt1055>.
- Wu YC, Rasmussen MD, Bansal MS, Kellis M. TreeFix: Statistically informed gene tree error correction using species trees. *Syst Biol.* 2013;62(1):110–20.

7. Noutahi E, Semeria M, Lafond M, Seguin J, Boussau B, Guéguen L, El-Mabrouk N, Tannier E. Efficient gene tree correction guided by genome evolution. *PLoS One*. 2016;11(8):e0159559. <https://doi.org/10.1371/journal.pone.0159559>. eCollection 2016.
8. Chen K, Durand D, Farach-Colton M. Notung: Dating gene duplications using gene family trees. *J Comput Biol*. 2000;7:429–47.
9. Rasmussen MD, Kellis M. A bayesian approach for fast and accurate gene tree reconstruction. *Mol Biol Evol*. 2011;28(1):273–90.
10. Thomas PD. GIGA: a simple, efficient algorithm for gene tree inference in the genomic age. *BMC Bioinformatics*. 2010;11:312.
11. Zhang L. On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. *J Comput Biol*. 1997;4(2):177–87.
12. Hallett MT, Lagergren J. Efficient algorithms for lateral gene transfer problems. In: *Proceedings of the fifth annual international conference on computational biology (RECOMB 2001)*. New York: ACM; 2001. p. 149–56.
13. Ovadia Y, Fielder D, Conow C, Libeskind-Hadas R. The cophylogeny reconstruction problem is NP-complete. *J Comput Biol*. 2011;18(1):59–65. <https://doi.org/10.1089/cmb.2009.0240>.
14. Doyon JP, Scornavacca C, Gorbunov KY, Szöllösi GJ, Ranwez V, Berry V. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In: *Proceedings of the 14th International Conference on Research in Computational Molecular Biology (RECOMB)*. Berlin/Heidelberg: LNCS, Springer; 2010. p. 93–108.
15. Bansal MS, Eric JA, Kellis M. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*. 2012;28(12):283–91. <https://doi.org/10.1093/bioinformatics/bts225>.
16. Tofigh A, Hallett M, Lagergren J. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Trans Comput Biol Bioinform*. 2011;8(2):517–35. <https://doi.org/10.1109/TCBB.2010.14>.
17. Chen ZZ, Deng F, Wang L. *IEEE/ACM Trans Comput Biol Bioinform (TCBB)*. 2012;9(5):1515–28.
18. Nguyen TH, Ranwez V, Pointet S, Chifolleau AMA, Doyon JP, Berry V. Reconciliation and local gene tree rearrangement can be of mutual profit. *Algorithms Mol Biol*. 2013;8(12). <https://doi.org/10.1186/1748-7188-8-12>.
19. Jacox E, Chauve C, Szöllösi GJ, Ponty Y, Scornavacca C. eccetera: comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics*. 2016;32(13):2056–8. <https://doi.org/10.1093/bioinformatics/btw105>.
20. Bansal MS, Wu Y, Alm EJ, Kellis M. Improved gene tree error-correction in the presence of horizontal gene transfer. *Bioinformatics*. 2015;31(8):1211–8. <https://doi.org/10.1093/bioinformatics/btu806>.
21. Szöllösi GJ, Tannier E, Daubin V, Boussau B. The inference of gene trees with species trees. *Syst Biol*. 2014;64(1):42–62.
22. Szöllösi GJ, Rosikiewicz W, Boussau B, Tannier E, Daubin V. Efficient exploration of the space of reconciled gene trees. *Syst Biol*. 2013;62(6):901–12. <https://doi.org/10.1093/sysbio/syt054>.
23. Sjöstrand J, Tofigh A, Daubin V, Arvestad L, Sennblad B, Lagergren J. A bayesian method for analyzing lateral gene transfer. *Syst Biol*. 2014;63(3):409–20.
24. Boussau B, Szöllösi GJ, Duret L, Gouy M, Tannier E, Daubin V. Genome-scale coestimation of species and gene trees. *Genome Res*. 2013;23:323–30.
25. Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn. Inc. Boston: Addison-Wesley Longman Publishing Co., Inc.; 1989.
26. Matsuda H. Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In: *Pacific Symposium on Biocomputing*. London: World Scientific. 1996. p. 512–23.
27. Lewis PO. Genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Mol Biol Evol*. 1998;15(3):277–83.
28. Skourikhine A. Phylogenetic tree reconstruction using self-adaptive genetic algorithm. In: *Proceedings of the 1st IEEE International Symposium on Bioinformatics and Biomedical Engineering (BIBE '00)*. Washington, DC: IEEE Computer Society; 2000. p. 129–34. <https://doi.org/10.1109/BIBE.2000.889599>.
29. Katoh K, Kuma K, Miyata T. Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny. *J Mol Evol*. 2001;53:477–84.
30. Lemmon AR, Milinkovitch MC. The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. *Proc Nat Acad Sci USA*. 2002;99(16):10516–21.
31. Zwickl DJ. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. PhD thesis, The University of Texas at Austin. 2006.
32. Lanave C, Preparata G, Saccone C, Serio G. A new method for calculating evolutionary substitution rates. *J Mol Evol*. 1984;20(1):86–93.
33. Yang Z. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods. *J Mol Evol*. 1994;39(3):306–14.
34. Jones DT, Taylor WR, Thornton JM. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*. 1992;8(3):275–82.
35. Lafond M, Noutahi E, El-Mabrouk N. Efficient non-binary gene tree resolution with weighted reconciliation cost. In: Grossi R, Lewenstein M, editors. *27th Annual Symposium on Combinatorial Pattern Matching (CPM 2016)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 54. Dagstuhl: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; 2016. p. 14:1–14:12.
36. Felsenstein J. Evolutionary trees from dna sequences: a maximum likelihood approach. *J Mol Evol*. 1981;17(6):368–76.
37. Stamatakis A, Ludwig T, Meier H. Raxml-iii: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics*. 2004;21(4):456–63.
38. Tofigh A. Using trees to capture reticulate evolution: lateral gene transfers and cancer progression. PhD thesis, KTH. 2009.
39. Sjöstrand J. Reconciling gene family evolution and species evolution. PhD thesis, Numerical Analysis and Computer Science (NADA), Stockholm University. 2013.
40. Coello CAC, Lamont GB, Veldhuizen DAV. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Secaucus: Springer; 2006.
41. Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput*. 1994;2(3):221–48.
42. Miller BL, Goldberg DE, et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst*. 1995;9(3):193–212.
43. Shimodaira H. An approximately unbiased test of phylogenetic tree selection. *Syst Biol*. 2002;51(3):492–508.
44. Boeckmann B, Robinson-Rechavi M, Xenarios I, Dessimoz C. Conceptual framework and pilot study to benchmark phylogenomic databases based on reference gene trees. *Brief Bioinform*. 2011;12(5):423–35.
45. Penel S, Arigon AM, Dufayard JF, Sertier AS, Daubin V, Duret L, Gouy M, Perrière G. Databases of homologous gene families for comparative genomics. *BMC Bioinformatics*. 2009;10(6):3.
46. Edgar RC. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. 2004;32(5):1792–7.
47. Lartillot N, Lepage T, Blanquart S. Phylobayes 3: a bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics*. 2009;25(17):2286–8.
48. Swofford DL, Olsen GJ, Waddell PJ, Hillis DM. Phylogenetic inference. In: Hillis DM, Moritz C, Mable BK, editors. *Molecular systematics*. 2nd edn. Sunderland: Sinauer Associates, Inc.; 1996. p. 407–514.
49. Boeckmann B, Marcet-Houben M, Rees JA, Forslund K, Huerta-Cepas J, Muffato M, Yilmaz P, Xenarios I, Bork P, Lewis SE, et al. Quest for orthologs entails quest for tree of life: in search of the gene stream. *Genome Biol Evol*. 2015;7(7):1988–99.