RESEARCH

GATC: A Genetic Algorithm for gene Tree Construction under the Duplication-Transfer-Loss model of evolution

Emmanuel Noutahi* and Nadia El-Mabrouk

Abstract

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. However the best fit to sequence information can be lost during this process. We describe GATC, a new algorithm for reconstructing a binary gene tree topology with branch length, which 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. GATC can be used to either construct a gene tree from scratch or correct existing trees. It can thus be used independently or jointly with an existing reconstruction method, making it highly flexible to various input data types. GATC 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 phylogenies with variable rates of events (gene duplication, loss and transfer) and showed that it is able to improve gene tree reconstruction, compared with current state-of-the-art algorithms. GATC is freely available at https://github.com/UdeM-LBIT/GATC

Keywords: gene tree; genetic algorithm; phylogeny; reconciliation; gene duplication; horizontal gene transfer

Introduction

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), sequences-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 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 (events and edge rates) 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

*Correspondence: fmr.noutahi@umontreal.ca
Département d'informatique et de recherche opérationnelle, Montréal, Canada

Full list of author information is available at the end of the article

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 (hereafer DTL) model becomes NP-hard in general if time consistency is required for inferred events (unless the species tree is fully dated) [12, 13, 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, 16]. Due to this reasonable time-complexity, some recent phylogenetic software allow extending the gene family evolution model to transfers (MowgliNNI [17], eccTERA [18], TreeFix-DTL [19]). Continuous effort is also made for developing fast probabilistic frameworks capturing HGT events (see [20] for a review of these models).

Integrative methods report gene trees with better accuracy compared with sequence-only methods [17, 21, 19, 22], 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 [23] and PrIME-DLTRS [22]) compute the joint likelihood associated with a substitution model and DTL event rates, given a fixed, dated and utrametric 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 minimum.

In this paper, we present *GATC* (*G*enetic *A*lgorithm for gene *T*ree *C*onstruction), 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, we can still use it as a two-step correction method, as input trees may be the output of a given phylogenetic tool that are then corrected using GATC, 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 [24]. 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 applied to the phylogenetic problem several times, beginning with Matsuda in 1996 [25] using a maximum likelihood criterion, Lewis [26] who introduced a subtree swap crossover, and other more recent algorithms (e.g. self-adaptive GA [27], Ga-mt [28], METAPIGA [29], GARLI [30]). 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 software on a dataset of simulated cyanobacteria 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.

1 Preliminaries

1.1 Notation on trees

All considered trees are rooted unless explicitely 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, $\mathcal{L}(x) \cap \mathcal{L}(y) = \emptyset$.

A *species tree* is a tree S with $\mathcal{L}(S)$ being a set of species, and a *gene tree* is a tree G with $\mathcal{L}(G)$ being a set of genes, each gene g belonging to a genome $s(g)$. We denote by $\overline{G}$ 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, $\overline{G}$ may have several equally labeled leaves.

A *reconciliation* of G (or similarly $\overline{G}$) with S 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 $\big(p(l),\ l\big)$ from a tree T as the *deletion of l from T*.

1.2 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 theses 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* [24], 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.

2 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.

2.1 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 edges rate, branch lengths and the substitution model. Some of these parameters might not be defined or might be kept fixed during the evolutionary process. For example, the substitution model m is usually fixed for all generations, whereas duplication, loss, transfer and edge rates can vary when a probabilistic model if used to compute the reconciliation score. When parsimony is preferred, they correspond to fixed event cost, except edges rate which is undefined. Branch lengths are usually optimized during sequence likelihood computation.

In the GATC implementation, when not provided, the initial values of the hyperparameters are randomly set from a uniform distribution. Unless specified otherwise, the substitution model used is GTR + Gamma for nucleotide sequences and JTT + Gamma for proteins.

2.2 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 initial population is generated, either randomly or according to a predefined procedure. The GATC implementation allows generating the initial trees from a star tree, using PolytomySolver [31] which outputs optimal 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 prefered 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.

2.3 Computing the sequence likelihood and reconciliation score

To evaluate the fitness of each chromosome σ_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 [32] and the further computationnal enhancement described by Stamatakis et al. (2004) [33]. 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 either under 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, as induced temporal constraints may be contradictory. 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 [34, 35] and used by PrIME-DLRS [22]. It is based on a birth-death model of evolution including rates for gene duplication, transfer and loss and requires discretization of a dated species tree, and numerical resolution of ordinary differential equations. We refer the reasder to the Supplementary Material of [22] 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.

2.4 Computing the fitness score

Given a raw score vector $\vec{z_i}$ for a genome 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 reconciliation and sequences data.

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 [36]. Here we will use a technique similar to the widely known NSGA (Non-dominated Sorting Genetic Algorithm) [37, 38].

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 $\vec{z_i} \prec \vec{z_j}$, 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 :

$$PS = \{\sigma_i \mid \nexists\, \sigma_j, \vec{z_j} \prec \vec{z_i}\}$$

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',

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

procedure computeNextPop(P_k)
  Compute P'_k, the offspring population of P_k
  $P^*_k \leftarrow P_k \cup P'_k$
  Evaluate z_i for all $\sigma_i \in P^*_k$
  Compute the dominance rank d_i for each $\sigma_i \in P^*_k$
  $w \leftarrow 1$
  while $\exists\, \sigma_i \in P^*_k \mid d_i = 0$ **do**
    $Wave_w \leftarrow \{\sigma_i \mid d_i = 0\}$
    Set a shared fitness for all $\sigma_i \in Wave_w$ as w
    $P^*_k \leftarrow P^*_k \setminus Wave_w$
    Compute the dominance rank d_i for each $\sigma_i \in P^*_k$
    $w \leftarrow w + 1$
  end while
  for $\sigma_i \in P^*_k$ **do**
    set the fitness of σ_i as $w + d_i$
  end for
  $P_{k+1} \leftarrow \bigcup_w Wave_w \cup P^*_k$
  return the first $|P_k|$ of P_{k+1} according to fitness
end procedure

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}.

2.5 Selection

GATC implements multiple classical selection methods. Individuals can either be selected for crossover using the tournament selector [39] or using the roulette wheel selector which chooses individuals with probability inversely proportionnal to their fitness values (recall that the best indivuals 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 indivuals are used in the crossover operator to produce the individuals of the next generation.

2.6 Crossover

In the crossover operators implemented in GATC, two offsprings are created from two parent genomes. 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 [26], 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 Figure 1A.

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 Figure 1B).

2.7 Mutation

For a chromosome $\sigma_i = (G_i, \theta_i)$, a mutation is performed either on the tree G_i or on the rates λ, δ, τ, e 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 rerooting operations (see Figure 2A-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 Figure 2C).

2.8 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 based on the use of a reference ML tree. Under the *Population-AU criterion*, evolution is stopped when none of the individuals in the current population is statiscally worse than the known

ML tree according to the AU test [40]. This stop criterion allows for a good performance when the objective is restricted to the optimization of sequence likelihood.

3 Experimental results

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 [18], TreeFix-DTL [19], ProfileNJ [7], MowgliNNI [17] 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 [41]. 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 public simulated cynobacteria dataset of Szöllősi et al. (2013) [21] 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 [42] and multiple alignments were performed on these families with Muscle [43]. For each alignment, an MCMC sample of at least 3000 trees was obtained with PhyloBayes [44] and used to reconstruct an amalgamated tree with ALE [21]. These trees were used to simulate new multiple alignments of artifical sequences under the LG model with a gamma distribution. We refer to [21] for a more detailed description on the construction of the dataset.

From each of the 1099 simulated artificial sequence alignments, we reconstructed an inital 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 $\tau = \inf$. 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 mid-point rooting method [45]. MowgliNNI, ecceTERA and ProfileNJ were run with a threshold of 0.7 for weak edges contraction. 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 parcimony for DTL-reconciliation. The crossover and mutation rates were respectively set to 0.8 and 0.5 and we used the tournament selector as the selection operator. 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 Figure 3, 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 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. However, it should be noted that 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) [41] 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 [46]. Protein sequences from genomes not found in the species tree were removed and the remaining sequences aligned with Muscle [43]. 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 the following modfications: the DTL events cost has been changed to: $(\lambda = 1, \tau = 0, \delta = 1)$; the maximum number of generations has been set to 300 and the maximum time of evolution set to 3h per gene family. For comparison, the average time needed by RAxML to obtain the best ML trees is 2.6h.

In order to measure the accuracy of GATC, we investigated how close the reference trees were to the set of pareto optimal trees. Figure 4 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.

For the POP family, we report the precision and recall of orthologous and parologous genes inference for the solutions returned by GATC, compared to the proposed reference POP gene family 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 files 2-5). We computed precision and recall for the two types of gene relationships as follows:

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

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 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 Figure 4, 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 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 necessary hold tor 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. As implemented, 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 adressed 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 crossover and mutations rates, population size and initialization) can be performed in parallel with exchange of information through a migration operator.

References

1. Guindon, S., Gascuel, O.: A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic Biology **52**, 696–704 (2003)
2. Stamatakis, A.: RAxML-VI-HPC: Maximum likelihood-based phylogenetic analysis with thousands of taxa and mixed models. Bioinformatics **22**, 2688–2690 (2006)
3. Ronquist, F., Huelsenbeck, J.P.: MrBayes3: Bayesian phylogenetic inference under mixed models. Bioinformatics **19**, 1572–1574 (2003)
4. Lartillot, N., Philippe, H.: A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Molecular Biology and Evolution **21**(6), 1095–1109 (2004). doi:10.1093/molbev/msh112
5. Schreiber, F., Patricio, M., Muffato, M., Pignatelli, M., Bateman, A.: Treefam v9: a new website, more species and orthology-on-the-fly. Nucleic Acids Research (2013). doi: 10.1093/nar/gkt1055
6. Wu, Y.C., Rasmussen, M.D., Bansal, M.S., Kellis, M.: TreeFix: Statistically informed gene tree error correction using species trees. Systematic Biology **62**(1), 110–120 (2013)
7. Noutahi, E., Semeria, M., Lafond, M., Seguin, J., Gueguen, L., El-Mabrouk, N., Tannier, E.: Efficient gene tree correction guided by genome evolution. Plos.One **11**(8) (2016)
8. Chen, K., Durand, D., Farach-Colton, M.: Notung: Dating gene duplications using gene family trees. Journal of Computational Biology **7**, 429–447 (2000)
9. Rasmussen, M.D., Kellis, M.: A bayesian approach for fast and accurate gene tree reconstruction. Mol. Biol. Evol. **28**(1), 273–290 (2011)
10. Thomas, P.D.: GIGA: a simple, efficient algorithm for gene tree inference in the genomic age. BMC Bioinformatics **11**, 312 (2010)
11. Zhang, L.: On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. Journal of Computational Biology **4**(2), 177–187 (1997)
12. Hallett, M.T., Lagergren, J.: Efficient algorithms for lateral gene transfer problems. RECOMB International Workshop on Comparative Genomics, pp. 149–156 (2001)
13. Ovadia, Y., Fielder, D., Conow, C., Libeskind-Hadas, R.: The cophylogeny reconstruction problem is NP-complete. J. Comput. Biol. **18**(1), 59–65 (2011). doi:10.1089/cmb.2009.0240
14. Doyon, J.-P., Scornavacca, C., Gorbunov, K.Y., Szöllősi, G.J., Ranwez, V., Berry, V.: An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. RECOMB International Workshop on Comparative Genomics, pp. 93–108 (2010)
15. Bansal, M.S., Eric, J.A., Kellis, M.: Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics **28**(12), 283–291 (2012). doi:10.1093/bioinformatics/bts225
16. Tofigh, A., Hallett, M., Lagergren, J.: Simultaneous identification of duplications and lateral gene transfers. IEEE/ACM Trans.Comput.BiolBioinform. **8**(2), 517–535 (2011). doi:10.1109/TCBB.2010.14
17. Nguyen, T.-H., Ranwez, V., Pointet, S., Chifolleau, A.-M.A., Doyon, J.-P., Berry., V.: Reconciliation and local gene tree rearrangement can be of mutual profit. Algorithms for Molecular Biology **8**(12) (2013)
18. Jacox, E., Chauve, C., Szöllősi, G.J., Ponty, Y., Scornavacca, C.: eccetera: comprehensive gene tree-species tree reconciliation using parsimony. Bioinformatics **32**(13), 2056–2058 (2016). doi:10.1093/bioinformatics/btw105
19. Bansal, M.S., Wu, Y., Alm, E.J., Kellis., M.: Improved gene tree error-correction in the presence of horizontal gene transfer. Bioinformatics **31**(8), 1211–1218 (2015). doi:10.1093/bioinformatics/btu806
20. Szöllősi, G.J., Tannier, E., Daubin, V., Boussau, B.: The inference of gene trees with species trees. Systematic biology **64**(1), 42–62 (2014)
21. Szöllősi, G.J., Rosikiewicz, W., Boussau, B., Tannier, E., Daubin, V.: Efficient exploration of the space of reconciled gene trees. Systematic Biology **62**(6), 901–912 (2013). doi:10.1093/sysbio/syt054
22. Sjöstrand, J., Tofigh, A., Daubin, V., Arvestad, L., Sennblad, B., Lagergren, J.: A bayesian method for analyzing lateral gene transfer. Systematic biology **63**(3), 409–420 (2014)
23. Boussau, B., Szöllősi, G.J., Duret, L., Gouy, M., Tannier., E., Daubin, V.: Genome-scale coestimation of species and gene trees. Genome Research **23**, 323–330 (2013)
24. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Inc. Boston, MA, USA (1989)
25. Matsuda, H.: Protein phylogenetic inference using maximum likelihood with a genetic algorithm. Pacific Symposium on Biocomputing, pp. 512–523. World Scientific, London (1996)
26. Lewis, P.O.: Genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. Mol. Biol. Evol. **15**(3), 277–283 (1998)
27. Skourikhine, A.: Phylogenetic tree reconstruction using self-adaptive genetic algorithm. IEEE Int. Symp. Bio-Inf. and Biomed. Eng., pp. 129–134 (2000). doi:10.1109/BIBE.2000.889599
28. Katoh, K., Kuma, K., Miyata, T.: Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny. J. Mol. Evol. **53**, 477–484 (2001)
29. Lemmon, A.R., Milinkovitch, M.C.: The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. Proc.Nat.Acad.Sci.USA **99**(16), 10516–10521 (2002)
30. Zwickl, D.J.: 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)
31. Lafond, M., Noutahi, E., El-Mabrouk, N.: Efficient non-binary gene tree resolution with weighted reconciliation cost. In: LIPIcs-Leibniz International Proceedings in Informatics, vol. 54 (2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
32. Felsenstein, J.: Evolutionary trees from dna sequences: a maximum likelihood approach. Journal of molecular evolution **17**(6), 368–376 (1981)
33. Stamatakis, A., Ludwig, T., Meier, H.: Raxml-iii: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics **21**(4), 456–463 (2004)
34. Tofigh, A.: Using trees to capture reticulate evolution: lateral gene transfers and cancer progression. PhD thesis, KTH (2009)
35. Sjöstrand, J.: Reconciling gene family evolution and species evolution. PhD thesis, Numerical Analysis and Computer Science (NADA), Stockholm University (2013)
36. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer, Secaucus, NJ, USA (2006)
37. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary computation **2**(3), 221–248 (1994)
38. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
39. Miller, B.L., Goldberg, D.E., *et al.*: Genetic algorithms, tournament selection, and the effects of noise. Complex systems **9**(3), 193–212 (1995)
40. Shimodaira, H.: An approximately unbiased test of phylogenetic tree selection. Systematic biology **51**(3), 492–508 (2002)
41. Boeckmann, B., Robinson-Rechavi, M., Xenarios, I., Dessimoz, C.: Conceptual framework and pilot study to benchmark phylogenomic databases based on reference gene trees. Briefings in bioinformatics **12**(5), 423–435 (2011)
42. Penel, S., Arigon, A.-M., Dufayard, J.-F., Sertier, A.-S., Daubin, V., Duret, L., Gouy, M., Perrière, G.: Databases of homologous gene families for comparative genomics. BMC bioinformatics **10**(6), 3 (2009)
43. Edgar, R.C.: Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research **32**(5), 1792–1797 (2004)
44. Lartillot, N., Lepage, T., Blanquart, S.: Phylobayes 3: a bayesian software package for phylogenetic reconstruction and molecular dating. Bioinformatics **25**(17), 2286–2288 (2009)
45. Swofford, D.L., Olsen, G.J., Waddell, P.J., Hillis, D.M.: Phylogenetic inference (1996)
46. Boeckmann, B., Marcet-Houben, M., Rees, J.A., Forslund, K., Huerta-Cepas, J., Muffato, M., Yilmaz, P., Xenarios, I., Bork, P.,

Lewis, S.E., *et al.*: Quest for orthologs entails quest for tree of life: in search of the gene stream. Genome biology and evolution **7**(7), 1988–1999 (2015)

Figures

Figure 1 Crossover operator. A. Subtree swap. A subtree $G_1[x]$ (in red) is pruned from G_1 then regrafted to a random branch of G_2 after deleting from G_2 its leaves that also appear in $G_1[x]$ (shown in dotted lines). To obtain the second child, a similar operation is performed from G_2 to G_1 **B. Subtree swap preserving reconciliation**. Two subtrees $G_1[x]$ and $G_2[y]$, respectively from G_1 and G_2, such that $\overline{G_1[x]} = \overline{G_2[y]}$ are swapped and the remaining leaves are corrected to conserve the same leafset as the parent.

Figure 2 Mutation operator. A. Re-rooting. The tree is rerooted at a random edge. **B. SPR move**. A subtree is pruned from the tree and regrafted to another edge. **C. Mutation preserving reconciliation cost**. Two leaves l_1 and l_2 such that $s(l_1) = s(l_2)$ are swapped. This mutation only alter the sequence likelihood.

Figure 3 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 tree for all 1099 gene families. GATC achieves the best accuracy on the simulated dataset, followed by Treefix-DTL.

Figure 4 Distribution of individuals' raw scores during evolution on three "gold standard" gene families. The scores of the ML tree obtained with RAxML, the MPR tree for the DL score, and the reference gene tree of [41] are also shown. Note that for fair compairson, the RAxML tree reconciliation score correspond to the best rooting score, whereas the MPR tree sequence likelihood correspond to the tree with the minimum negative log likelihood in the set of equivalent MPR trees. For the sake of visibility, we increased the size of each data point. The "best tree" is expected to be located in the lower left corner. For the ATPase and Nox families, the reference tree was present in the set of pareto optimal trees returned by GATC. For the Popeye gene families, the reference tree was located in the proximity of a cluster of pareto optimal solutions.

Tables

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

Additional Files 1 to 5

Additional file 1 — Reference tree for the Popeye family
Topology of the reference tree for the Popeye family. Branch lengths are not shown. The gene tree was reconcilied with the species tree. Duplication nodes (leading to paralogs) are indicated by a red square, while speciation nodes(leading to orthologs) are indicated by a green circle. The total number of duplications and losses are shown at the bottom. Lost branches were not shown for clarity.

Additional file 2 — Tree 1 return by GATC
Additional file 3 — Tree 2 return by GATC
Additional file 4 — Tree 3 return by GATC
Additional file 5 — Tree 4 return by GATC

Figure 1

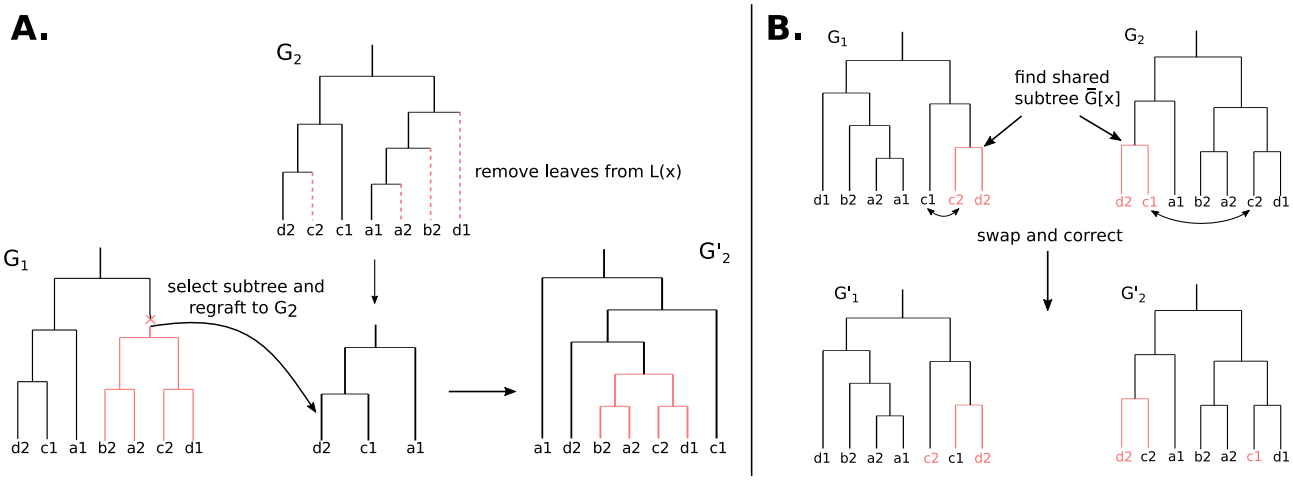

A.
G2
d2 c2 c1 a1 a2 b2 d1
remove leaves from L(x)
G1
select subtree and
regraft to G2
d2 c1 a1 b2 a2 c2 d1
d2 c1 a1
G'2
a1 d2 b2 a2 c2 d1 c1
B.
G1
G2
find shared
subtree G̅[x]
d1 b2 a2 a1 c1 c2 d2
d2 c1 a1 b2 a2 c2 d1
swap and correct
G'1
G'2
d1 b2 a2 a1 c2 c1 d2
d2 c2 a1 b2 a2 c1 d1

Figure 2

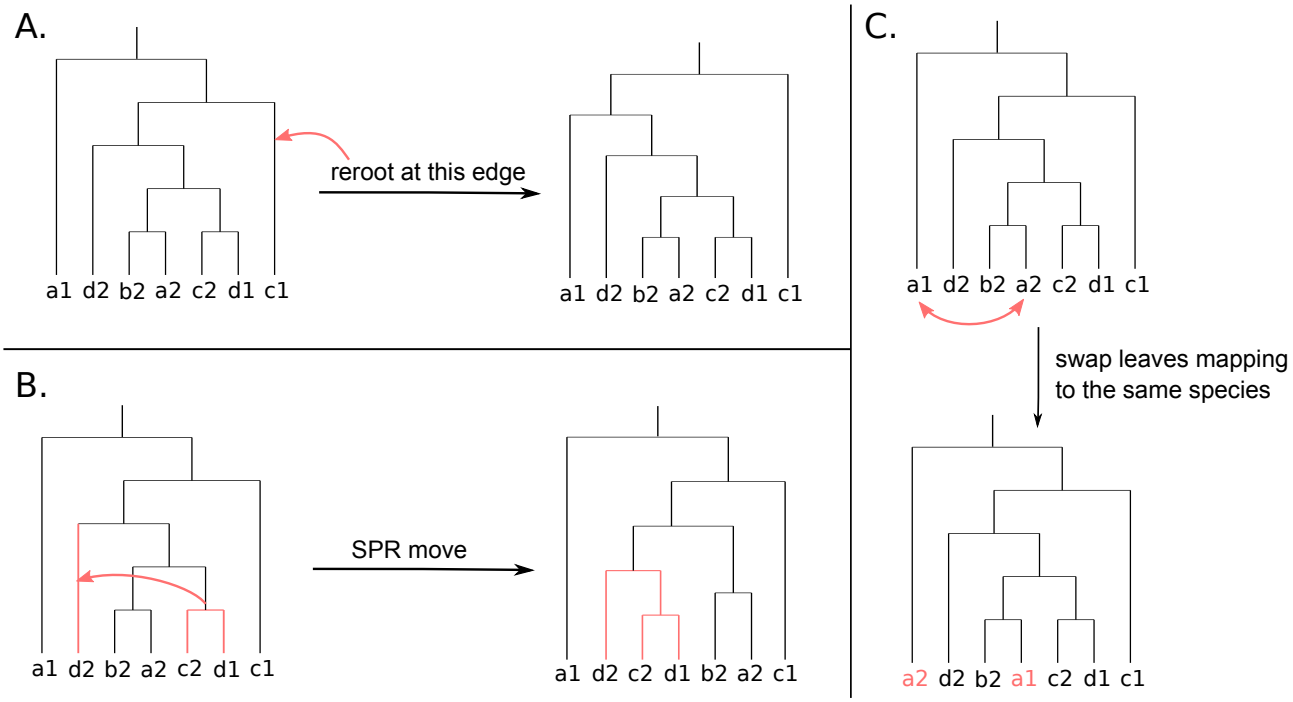

A.
reroot at this edge
a1 d2 b2 a2 c2 d1 c1
a1 d2 b2 a2 c2 d1 c1
B.
SPR move
a1 d2 b2 a2 c2 d1 c1
a1 d2 c2 d1 b2 a2 c1
C.
a1 d2 b2 a2 c2 d1 c1
swap leaves mapping
to the same species
a2 d2 b2 a1 c2 d1 c1

Figure 3

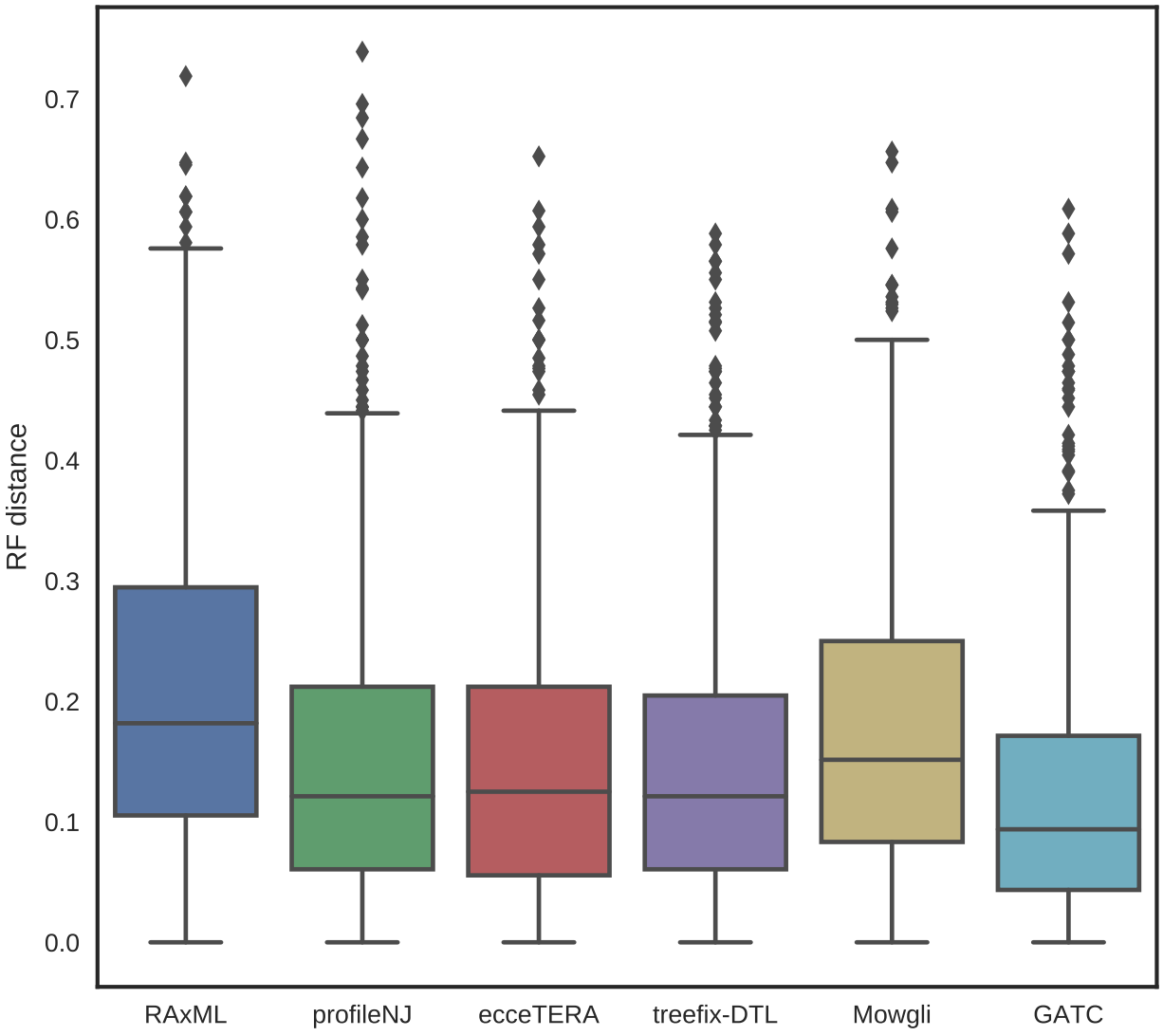
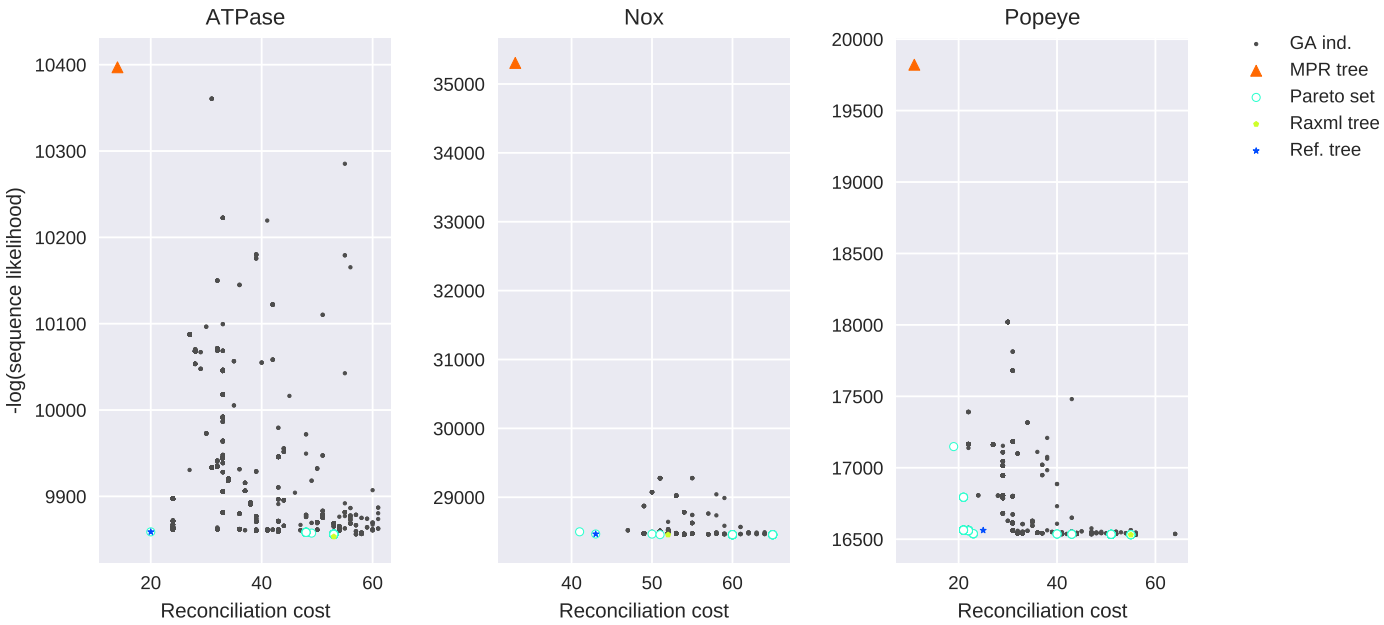
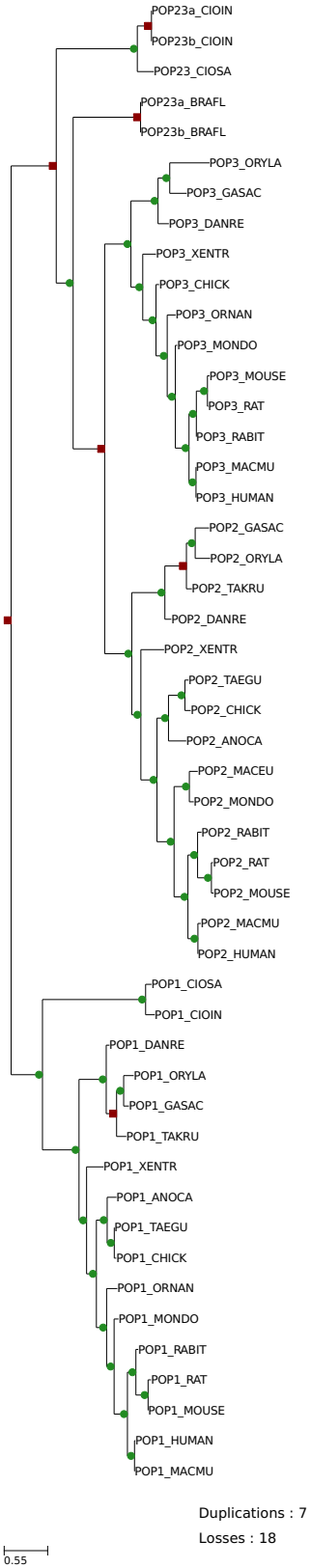

RF distance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
RAxML
profileNJ
ecceTERA
treefix-DTL
Mowgli
GATC

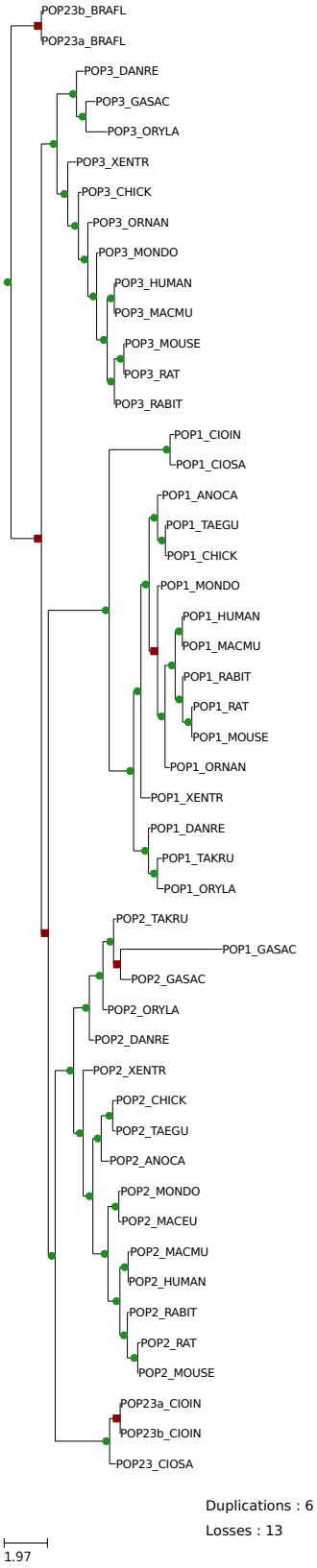
Figure 4


ATPase
Nox
Popeye
-log(sequence likelihood)
Reconciliation cost
Reconciliation cost
Reconciliation cost
GA ind.
MPR tree
Pareto set
Raxml tree
Ref. tree

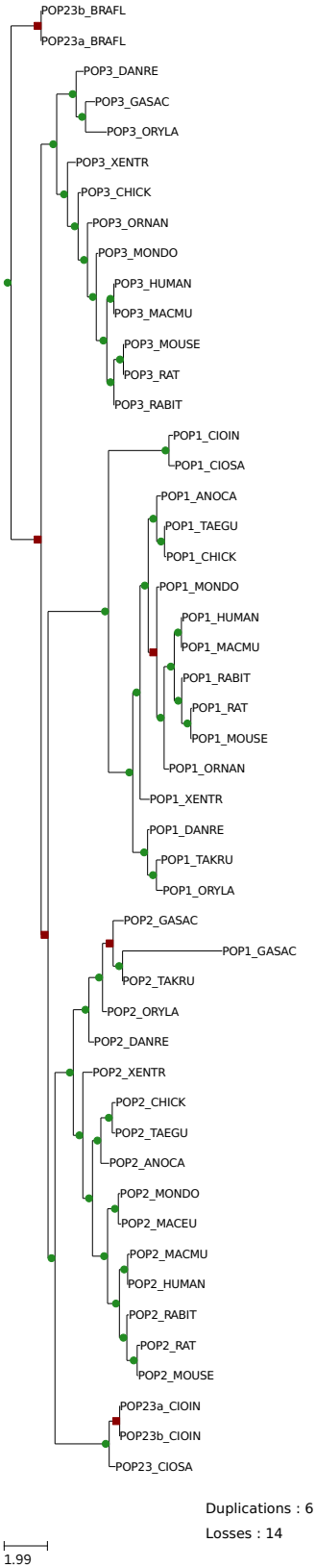
Additional file 1



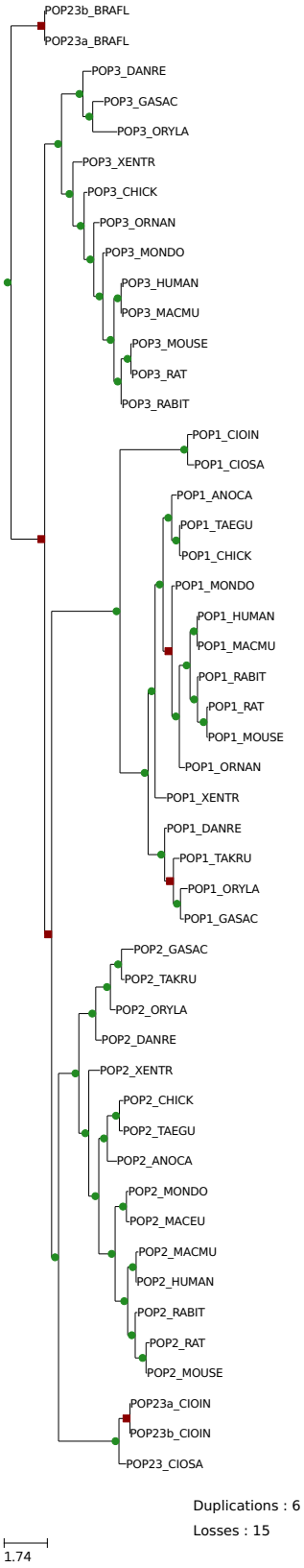
Additional file 2



Additional file 3



Additional file 4



Additional file 5