Efficient gene tree correction guided by genome evolution

Emmanuel Noutahi, Magali Semeria, Manuel Lafond, Jonathan Seguin, Bastien Boussau, Laurent Guéguen, Nadia El-Mabrouk, Eric Tannier

Efficient gene tree correction guided by genome evolution

Emmanuel Noutahi[1,☯], Magali Semeria[2,☯], Manuel Lafond[1], Jonathan Seguin[1], Bastien Boussau[2], Laurent Guéguen[2], Nadia El-Mabrouk[1], Eric Tannier[2,3,*]

1 Département d'Informatique (DIRO), Université de Montréal, H3C3J7, Canada
2 LBBE, UMR CNRS 5558, Université de Lyon 1, F-69622 Villeurbanne, France
3 INRIA Grenoble Rhône-Alpes, F-38334 Montbonnot, France

☯These authors contributed equally to this work.
*** Eric.tannier@inria.fr**

Abstract

Gene trees inferred solely from multiple alignments of homologous sequences often contain weakly supported and uncertain branches. Information for their full resolution may lie in the dependency between gene families and their genomic context. Integrative methods, using species tree information in addition to sequence information, have therefore been developed. They often rely on a computationally intensive tree space search which forecloses an application to large genomic databases. We propose a new method, called ProfileNJ, that takes a gene tree with statistical supports on its branches, and corrects its weakly supported parts by using a combination of information from a species tree and a distance matrix. Its low running time enabled us to use it on the whole Ensembl Compara database, for which we propose an alternative, arguably more plausible set of gene trees. This allowed us to perform a genome-wide analysis of duplication and loss patterns on the history of 63 eukaryote species, and predict ancestral gene content and order for all ancestors along the phylogeny.
The code of ProfileNJ is available at:
`https://github.com/UdeM-LBIT/profileNJ`
and a web interface called RefineTree, including ProfileNJ as well as several published gene tree correction methods based on synteny, which we also test on the Ensembl gene families, is available at:
`http://www-ens.iro.umontreal.ca/~adbit/polytomysolver.html`

Introduction

Several gene tree databases from whole genomes are available, including Ensembl Compara [1], Hogenom [2], Phog [3], MetaPHOrs [4], PhylomeDB [5], Panther [6]. However they are known to contain many errors and uncertainties, in particular for unstable families [7]. Their use for accurate ancestral genome inference, orthology detection, or the study of genome dynamics could lead to erroneous results. For example Ensembl Compara trees, when reconciled with a species tree to annotate gene duplications and losses, systematically and unrealistically overestimate the number of genes in ancestral genomes, and lead to erroneous predictions of ancestral chromosome structures [8]. It is a known artifact, and a substantial number of nodes in the Ensembl gene trees are labeled as "dubious" [9].

Reasons for errors in gene trees are numerous. If they are constructed from multiple sequence alignments of homologous genes, they are dependent on gene annotations, gene family clustering or alignment quality, as well as on the accuracy of the models and algorithms used. But above all, gene sequences often do not

contain enough substitutions to resolve all the branches of a phylogeny, or alternatively, too many substitutions such that the substitution history is saturated. Therefore *sequence based methods*, computing gene trees from sequence information (e.g. PhyML [10], RAxML [11], MrBayes [12], PhyloBayes [13]), are usually accompanied with measures of statistical support on their branches or *a posteriori* distributions of likely trees.

Another category of methods, designated here as *integrative methods*, use a species tree, in addition to a multiple sequence alignment, to model gene gains and losses inferred from the reconciliation between gene and species trees (e.g. TreeBeST [14], TreeFix [15], Notung [16], BBCA [17], PhylDog [8], ALE [18], GSR [19, 20], SPIMAP [21], Giga [22], MowgliNNI [23]). They all report gene trees with better accuracy compared with sequence based methods. But they leave a large space for improvement, both in terms of tree quality and computing time. In terms of models, they often assume unrealistic loss/retention ratios [**?**]. In terms of computation strategy, most of them use tree space exploration strategies based on small modifications or *local moves* on branches (typically NNI, SPR, TBR), usually proposed at random. Moves are accepted or rejected according to hill-climbing, Metropolis-like criteria, or other statistical or empirical arguments. Such exploration methods are computationally intensive and do not scale well as databases grow in size. Consequently, database construction pipelines such as TreeBeST (contructing the Ensembl Compara gene trees [1]) have to adopt compromises, exploring limited subsets of tree spaces. Improving local exploration of integrative methods can be done by using some *correction* techniques allowing to select directly the local moves that improve the reconciliation (e.g. [16, 23–33]). But even with such improvements, it remains that most local search strategies have no guarantee neither on running time, nor on the quality of the solution.

We propose here a new gene tree correction method, called ProfileNJ, which can be directly used as a fast integrative method, without local search. It is a deterministic approach with a guaranteed time complexity. ProfileNJ takes as input a starting tree with supports on its branches, typically constructed from a sequence based method, and outputs a rooted binary tree containing all well-supported branches of the starting tree, and minimizing the number of duplications and losses when reconciled with a species tree. Among all trees with equal reconciliation cost, a choice is made with Neighbor-Joining (NJ) principles, based on a distance matrix computed from gene sequences or from the starting tree. ProfileNJ extends a previous algorithm of our group [31] by integrating NJ principles to choose among the numerous optimal solutions, and by allowing different costs for duplications and losses, as well as unrooted trees as input. These extensions turn an algorithmic principle into a workable method suited for constructing trees from biological data.

We compare ProfileNJ with TreeFix [15]. Among correction methods, TreeFix adopts the most similar evaluation strategy, *i.e.* explores neighboring trees which are statistically equivalent according to the sequences. Moreover, TreeFix is among the best available integrative methods, according to the quality of the output trees and running time. On simulations, both algorithms achieve results of comparable quality, but ProfileNJ is several times faster, which opens the way to using ProfileNJ on big datasets.

We ran ProfileNJ on the whole set of gene families from the Ensembl database, which is out of reach for competing methods with comparable quality. The trees for the whole database were obtained in a few hours on a desktop computer (not including the starting tree construction, performed with PhyML) and compare very favorably with the trees stored in Ensembl. This set of trees and the reconstructed ancestral genomes are made accessible. We also use the reconstructed trees and ancestral genomes to study genome evolution across all the 63 eukaryotic species from the Ensembl database. A whole genome analysis of duplication patterns is provided, pointing at certain branches which seem to show acceleration of duplication or loss processes.

ProfileNJ is integrated into a modular interface called RefineTree that contains two other gene tree correction tools using information from extant and ancestral synteny [32, 33]. We can thus evaluate the results of a pipeline taking into account gene sequence, gene content and chromosome structure evolution on the Ensembl database according to several criteria: (1) likelihood ratio based on the Ensembl alignments; (2) ancestral genome sizes based on reconciliation with the species tree; (3) linearity of ancestral chromosomal segments computed with DeCo [34]. We discuss the improvements brought by each type of method and the distance of their output to "true" gene trees, in the light of incomplete lineage sorting and gene conversion.

Description of ProfileNJ

The basic vocabulary of phylogenetic trees is taken from [35], and the reconciliation method between a rooted binary gene tree G and a rooted binary species tree S is recalled in the Method section. Just note that in a reconciled gene tree G, each node (representing an extant gene if at a leaf or an ancestral gene if at an internal node) is mapped to the node of S corresponding to the genome the gene belongs to. Edges of G are subdivided, adding extra vertices and pending edges so that the extremities of an edge map either to the same node, or to two extremities of an edge of S. An internal node of G is a duplication if it maps to the same node of S as one of its child. The number of genes in a species $s \in V(S)$ induced by a reconciled gene tree G is defined as the number of nodes $x \in V(G)$ mapped to s, such that the parent of x does not map to s. The reconciliation cost is either the number of duplications and losses, or a linear combination of the two if different weights are given to the two kinds of events. Also note that when two trees G_1 and G_2 have the same genes at their leaves, we can say that a branch of G_1 is present in G_2 if the bipartition of the leaves induced by this branch in G_1 is also induced by a branch of G_2.

ProfileNJ is a gene tree correction algorithm that takes as input a gene tree (rooted or unrooted) G for a given gene family with supports on its branches, and improves it according to the available information, taken from a species tree, a distance matrix and a threshold number for statistical support. It can be viewed as a generalization of three different standard algorithms designed for evolutionary studies:

- The Wagner parsimony method applied to the inference of ancestral gene contents from the extant gene contents by minimizing a duplication and loss cost [36];
- The Neighbor-Joining [37] (NJ) method which constructs a tree from a distance matrix D between taxa;
- The reconciliation of a gene tree G with a species tree S [38].

Whereas these three methods do not have much in common *a priori*, they are all bricks of our solution and each of them reduces to some particular case of our problem. ProfileNJ outputs a rooted binary gene tree G_c on the same gene family as the input gene tree G, where all branches of G with a support above the threshold are present in G_c. Among all such trees, ProfileNJ outputs those minimizing a duplication and loss cost when reconciled with the species tree with respect to the NJ criterion.

ProfileNJ is an extension of PolytomySolver, a previous algorithm developed by our group [31]. We first describe the principle of the latter and then describe the additions.

PolytomySolver: It takes as input a multifurcated rooted gene tree G (with non-binary nodes) and a binary rooted species tree S. It outputs a binary rooted gene tree containing all branches of G, that minimizes the number of duplications and losses when reconciled with S. It has been shown by [31] that each polytomy (multifurcated node) of G can be considered independently. Therefore, in the following, we restrict the presentation to a single polytomy P (s.f. polytomy P in Figure 1).

The algorithm, based on dynamic programming, computes a table M where, for each node (including leaves) s of S and each integer k (limits on k are discussed in [31]), $M(s,k)$ is the reconciliation cost of a gene tree with k genes in species s before any duplication in s. For example, in Figure 1, P has three genes belonging to genome b, and thus $M(b,1) = 2$ as any solution having one gene in b before any duplication in b means that two duplications must have occurred in b, while $M(b,4) = 1$ as having four genes induces one gene loss on b.

The final cost of a minimum refinement of the polytomy is given by $M(r,1)$, where r is the root of S. Using a backtracking approach, PolytomySolver then outputs a *count vector* V containing the number of genes per node of S. Notice that, by construction, two brother nodes of S (nodes with the same parent) have the same count. Then a gene tree G of minimum cost $M(r,1)$ is found, such that in the reconciliation of G with S there are exactly $V[s]$ genes in each $s \in V(S)$. For example, the final binary tree in Figure 1 has two maximal trees rooted at b, as required by the count vector V.

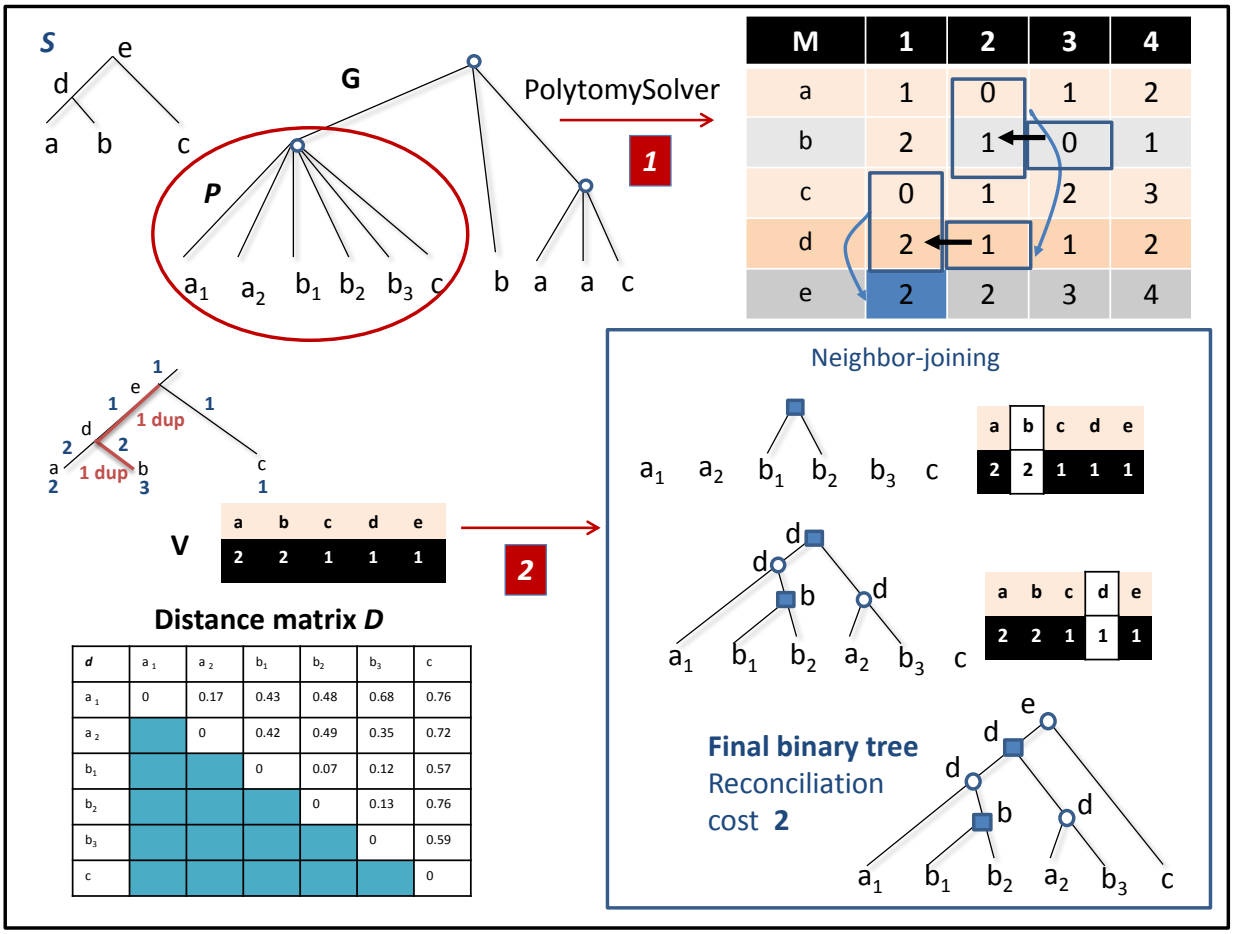

Figure 1. A species tree S and a multifurcated gene tree G. Each leaf x_i or x of G represents a gene belonging to genome x present as a leaf in S. Step (1) of ProfileNJ is PolytomySolver, which resolves each polytomy P of G independently. A dynamic programming table M is constructed. Step (2) of ProfileNJ takes as input a count vector V, here resulting from the backtracking path related by rectangles and arrows in table M, and a distance matrix d for the considered genes. A Neighbor joining (NJ) based procedure computes the gene tree in agreement with V that best reflects the distance matrix. The final completely refined tree is given bottom right. Duplication nodes are indicated by squares.

If the reconciliation cost is the number of duplications and losses (i.e. the same unit cost is attributed to each duplication or loss), as it was initially published [31], the table M can be constructed in time linear in the size of S [31], leading to a linear-time algorithm for finding one optimal refinement of the polytomy. Moreover, we showed recently [39] that linearity can be extended to a whole gene tree involving multiple polytomies. For weighted operations, i.e. different costs for duplications and losses, the algorithm runs in quadratic time [39].

Extensions: ProfileNJ consists in contracting branches with low support in G, which leads to polytomies, and applying Polytomysolver. If G is not rooted, one node is chosen as the root. All nodes can be tried and one minimizing the cost can be chosen. The first phases of Polytomysolver are applied, until the construction of a count vector V.

Our main extension concerns a treatment of the multiplicity of solutions, as it can be exponential. Indeed, the backtracking procedure mentioned above may lead to many optimal count vectors, and for each count vector, there are possibly several gene trees in agreement with it. Therefore, exploring the set of optimal trees requires exploring the set of all gene trees in agreement with each count vector. For example, the count vector of Figure 1 induces two duplications in b. However this vector involves no information on which of the three genes b_1, b_2, b_3 should be joined first. Such information can be deduced from the pairwise alignment distance between gene sequences.

Suppose that a pairwise distance matrix D is available for the gene family. Then the problem can be seen as selecting, among all optimal solutions possibly output by PolytomySolver given a vector V, the one best reflecting the distance matrix D. The problem of constructing a solution such that its induced distance is close to D according to a standard measure of metric spaces comparison is NP-complete. But it is also known to be empirically and, to a certain extent, theoretically, well approximated by Neighbor-Joining (NJ) [37, 40]. In ProfileNJ, we use such an NJ approach for choosing neighboring genes.

As in the NJ algorithm, a metric space E induced by D on the leaves of P, is progressively augmented with newly created genes. The algorithm proceeds by successively joining pairs of nodes (points of E), eventually leading to a full binary tree. For example, in Figure 1, the initial metric space E contains the nodes $\{a_1, a_2, b_1, b_2, b_3, c\}$. Joining the nodes b_1 and b_2 leads to the new set of nodes $\{a_1, a_2, b_3, b_4, c\}$. Nodes to be joined are selected according to the NJ criterion, namely we select from a node set of size n, the couple of genes x and y minimizing:

$$Q(x, y) = (n-2)D(x, y) - \sum_{t \neq x} D(x, t) - \sum_{t \neq y} D(y, t). \quad (1)$$

The metric space E is updated after each join $r = (x, y)$ by removing x and y, adding a new node r, and computing the distance between the newly created node r with each element t of E. When x and y are not created artificially (i.e. they are not loss nodes created with the last instruction of Algorithm 1), this is done using the NJ formula:

$$D(r, t) = \frac{1}{2}(D(x, t) + D(y, t) - D(x, y)) \quad (2)$$

Otherwise, if x is a loss, we set $D(r, t) = D(y, t)$ and if y is a loss, $D(r, t) = D(x, t)$.

A full pseudo-code of the extension part of ProfileNJ is written as Algorithm 1. It works on one polytomy P, assuming that all polytomies below have been resolved. It takes as input a count vector V, the species tree S and a distance matrix D defining the metric space E. It outputs a refinement of P in agreement with V, resulting from the performed joins on the nodes of E. Given a node s of S, denote by $E(s)$ the subset of E restricted to the genes belonging to s, and by $m(s) = |E(s)|$ the multiplicity of s in E. The tree S is processed bottom-up. For each internal node s, speciations are considered first by clustering, using the NJ criterion, the genes from $E(s^l)$ with the genes from $E(s^r)$, where s^l and s^r are the two children of s. If the obtained multiplicity $m(s)$ of s is greater than the desired count $V[s]$, then duplications are performed, again

using the NJ criterion for choosing the gene pairs in $E(s)$ to be joined. Otherwise, if $m(s)$ is lower than the desired count $V[s]$ of gene copies, then losses are predicted.

Algorithm 1 ProfileNJ (S,P,V,D)

Let E be the metric space with nodes corresponding to the leaves of P;
For each node s of S in a bottom-up traversal of S *Do*
  If s is an internal node of S with chidren s^l, s^r *Do*
    {By construction, $V[s^l] = V[s^r] = n$}
    For $i = 1$ *to* n *Do*
      Choose in $E(s^l) \times E(s^r)$ the gene pair (g^l, g^r) minimizing equation (1) and create the node $g = (g^l, g^r)$;
      Remove g^l and g^r from E and add g;
      Compute $D(g, g')$ for all $g' \in E$ using equation (2);
    End For
  End If
  If $m(s) > V[s]$ *Do*
    For $i = 1$ *to* $m(s) - V[s]$ *Do*
      {Perform $m(s) - V[s]$ duplications}
      Choose in $E(s) \times E(s)$ the gene pair (g_1, g_2) minimizing equation (1), and create the node $g = (g_1, g_2)$;
      Remove g_1 and g_2 from E and add g;
      Compute $D(g, g')$ for all $g' \in E$ using equation (2)
    End For
  End If
  Else If $m(s) < V[s]$ *Do*
    {Perform $V[s] - m(s)$ losses}
    Add $V[s] - m(s)$ artificial genes to $E(s)$, each with infinite distance to all elements of E;
  End If
End For

Complexity: Let G be the solution output by the algorithm, and suppose that G has n leaves after the inclusion of lost genes. Then exactly $n - 1$ NJ operations have been performed. Each join calculation is restricted to a subset of the genes, and so the time required to perform these joins is bounded by the time required to run the classical NJ algorithm on the n leaves of G, which is $O(n^3)$. Note that n can be as large as $|V(P)||V(S)|$, making the worst case running time $O(|V(P)|^3|V(S)|^3)$. However this worst case only occurs when $O(|V(S)|)$ losses are inserted on each branch of the solution. In practice n is in $O(|V(P)|)$.

A Multi-functional algorithm: ProfileNJ is a phylogenetic tool that generalizes several usually unrelated standard methods. Indeed, if G is a binary rooted tree, then ProfileNJ can be seen as a reconciliation tool. If G is unrooted, then ProfileNJ can be used to choose an appropriate root according to the induced reconciliation cost. On the other hand, various ways of contracting branches can be considered. For example an exploration scheme contracting the branches one by one and applying ProfileNJ can be considered, which would be equivalent to local modifications [27]. A more radical modification would be to contract all branches, leading to a star tree. In this case, ProfileNJ can be seen as a tool for computing ancestral gene content with Wagner parsimony, minimizing the cost of duplications and losses. If the star tree has all its genes belonging to a single species, ProfileNJ returns an NJ tree. Other kinds of contraction schemes can be imagined, as contracting branches around "Non Apparent Duplications" [41], or "Dubious duplications" stored in the Ensembl trees.

Notice finally that although we give the pseudo-code for a single output of ProfileNJ, it can be used to output all solutions, which allows for example selecting the gene tree reflecting the best statistical support.

Efficiency of ProfileNJ

Efficiency of the NJ criterion

We ran ProfileNJ twice on the same data sets of 20519 trees (the Ensembl Compara gene families), except that once the distance matrix was computed using the Ensembl nucleotide alignments with FastDist from the FastPhylo package [42], and once the distance matrix was random. The starting tree was computed for every family using PhyML on the nucleic alignments, and all branches with aLRT support < 0.95 were contracted. In average 55% of the branches were contracted. A histogram of the full distribution is shown in the supplementary information file SI1 (Figure S1).

Then we computed the likelihood of both trees for every family with PhyML. Among the trees for which the likelihood was different (55% of all tested trees), 76% were in favor of the trees built with the FastDist distance matrix, and the log likelihood differences were much larger for those trees, contributing 95% of the total of log likelihood differences.

The comparisons are clearly in favour of the NJ criterion over no criterion at all, while quantitatively there remains a small but non negligible part of the trees for which no criterion (the random distance matrix) gives an unexplained slightly, but significantly, better likelihood.

Efficiency of the tree space exploration strategy on simulated gene trees

We compared ProfileNJ with TreeFix, the most closely related tool, on simulated data. The principle of TreeFix is to randomly explore, by local moves, the space of trees that are statistically equivalent to the input tree, and report the one with the best reconciliation cost. Instead, we take a deterministic and more targeted approach by focusing on weakly supported branches of the tree, with a possibly deep modification of the tree. The comparison with TreeFix is therefore intended to compare these two space exploration strategies.

In [15], TreeFix has been compared with NOTUNG [24] and SPIMAP [21], showing a better accuracy than NOTUNG, and a higher speed than SPIMAP. We perform a similar comparison on the same simulated dataset of 16 fungi. This dataset consists of simulated gene families generated under the SPIMAP model and their corresponding nucleotide alignments, for four different rates of duplication and loss (DL) events: $(1 \times r_D,\ 1 \times r_L)$, $(2 \times r_D,\ 2 \times r_L)$, $(4 \times r_D,\ 4 \times r_L)$ and $(4 \times r_D,\ 1 \times r_L)$, where r_D and r_L are respectively the estimated duplication and loss rates for fungi. For instance, a $(2 \times r_D,\ 2 \times r_L)$-simulated gene family is expected to have, on average, two times more duplications and losses than a real gene family in fungi. Comparisons reported in this section are performed on 2575 simulated gene families randomly chosen from the four fungi datasets with different DL rates.

An initial maximum likelihood (ML) tree is constructed for each simulated gene family with RAxML v-8.1.2 [11], with the rapid bootstrap algorithm, under the GTR-Γ model and the majority rule consensus tree as bootstopping criterion. A randomly rooted tree is then provided as input to TreeFix (as TreeFix requires the input tree to be rooted), while a multifurcated unrooted tree obtained by contracting the branches with support lower than 95% is provided as input to ProfileNJ. We used default parameters for both programs. Among the set of all optimal binary trees output by ProfileNJ, the best statistically supported tree was selected using RAxML under the GTR-Γ model of nucleotide substitution.

For RAxML, TreeFix and ProfileNJ trees, we measured the Robinson-Foulds (RF) distance to true trees, compared the reconstructed tree with the true tree using site-wise likelihoods (see supplementary information SI1, Figure S7), measured the accuracy of the duplication and loss scenarios (supplementary information SI1, Figure S5), the sensitivity of the accuracy to gene family size (supplementary information SI1, Figure S6), the sensitivity to species tree errors (supplementary information SI1, Figure S8), and the running time.

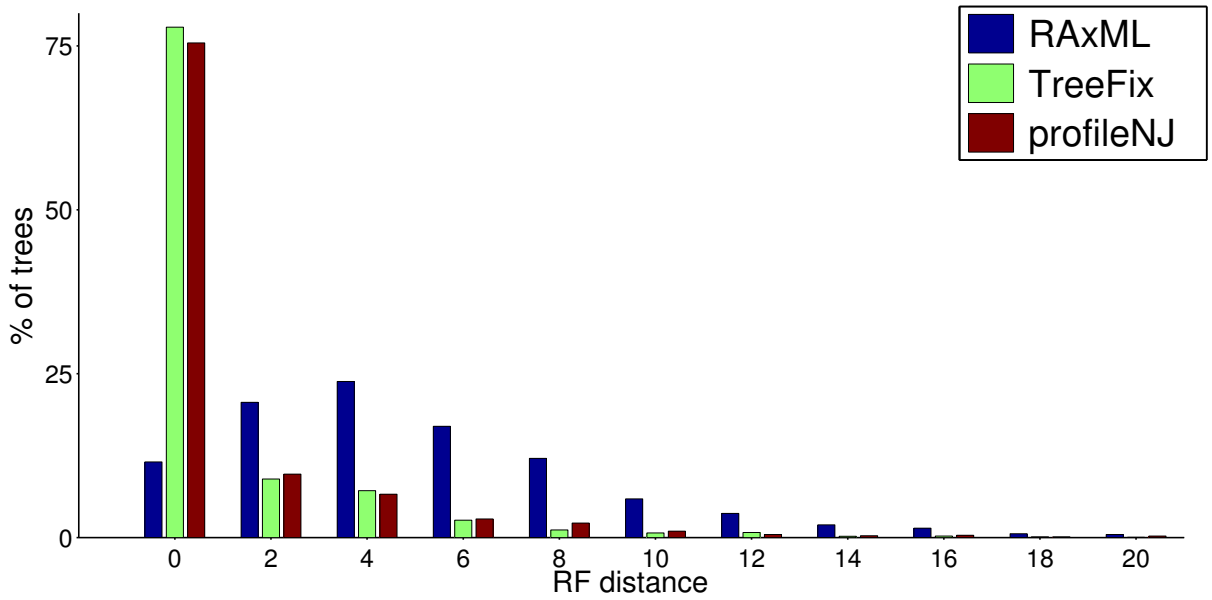

Figure 2. Topology accuracy of RAxML, TreeFix and ProfileNJ trees, mesured by RF distance with the true tree, on $\sim$ 2500 simulated trees from the fungal dataset. We use a sample of trees simulated under four different DL rate : ($1r_D$ - $1r_L$), ($2r_D$ - $2r_L$), ($4r_D$ - $4r_L$) and ($4r_D$ - $1r_L$). Percentage of reconstructed trees (y-axis) with a given RF distance (x-axis) to the true tree. TreeFix and ProfileNJ have a similar reconstruction accuracy (75% of trees match the true trees) while the input trees (RAxML) have the lowest accuracy. The graph is cut on the right, but contains more than 99% of the data.

Figure 2 illustrates the results for the RF distance. It shows that sequence-only does not contain enough signal to lead to the true tree for our simulated dataset, and integrating additional information from the species tree actually improves reconstruction. Indeed, TreeFix and ProfileNJ reconstruct around 75% of true trees, compared with only 10% for RAxML. We investigated some cases where erroneous gene trees were inferred, and found that often, the true scenario was not parsimonious in terms of duplications and losses, while TreeFix and ProfileNJ chose duplications that are too recent in order to avoid losses. An example is given in supplementary material (SI1, Figure S4).

The performances of TreeFix and ProfileNJ are similar in terms of distance to the true tree. As for RAxML, it gives the best likelihood, which is not surprising as it is specifically designed for that. The returned likelihood is even usually higher than the likelihood of the true tree, but not significantly according to an AU test. TreeFix is designed to produce trees which are not significantly different than the ML tree, which we could check: 1.36% of the trees fail the AU test against the ML tree at $\alpha = 0.05$, while the proportion jumps to 9.17% for ProfileNJ . It is noticeable that this has no visible consequence on the distance to the true tree.

Figure 3 shows that ProfileNJ outperforms TreeFix in running-time, the gap between the two algorithms increasing with tree size. This figure also shows that the most time-consuming step in ProfileNJ is tree selection. For a tree of size 30, ProfileNJ is about four to seven times faster than TreeFix, and about 15 times faster if we discard statistical support evaluation and tree selection step with RAxML. This includes the construction of the distance matrix, but not the construction of the initial RAxML, as it is common to both methods.

Other analyses, including the sensitivity to gene family size and the number of duplications and losses, are reported in supplementary material SI1. They lead to the same conclusions: TreeFix and ProfileNJ have similar performance on all measures except running time for which ProfileNJ is significantly better.

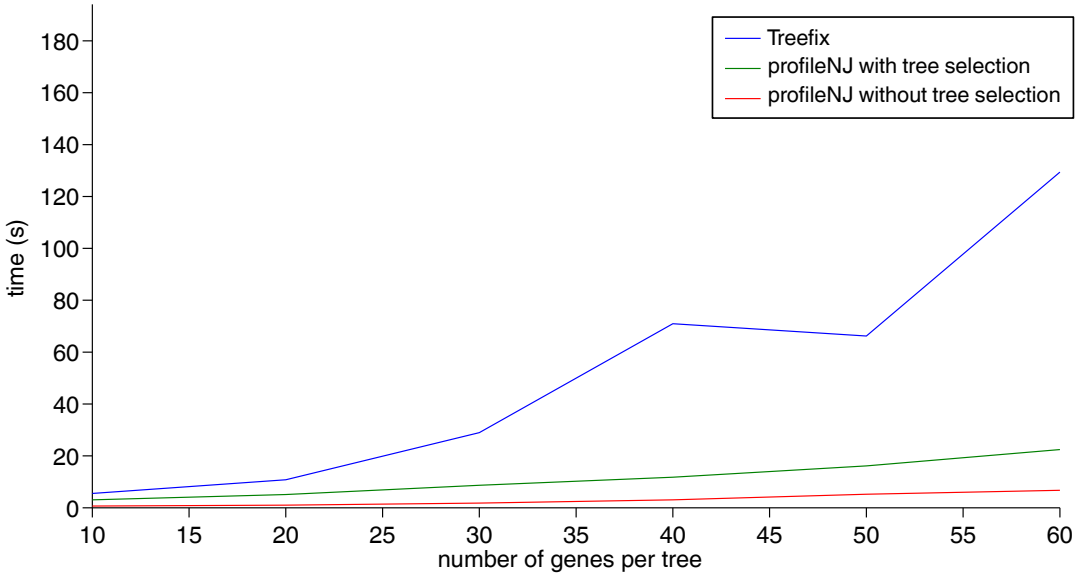

Figure 3. Runtime of TreeFix and ProfileNJ for increasing size of gene tree.

Application on biological data

RefineTree

ProfileNJ is integrated in a new modular online software, called RefineTree, designed to combine a number of correction techniques, with an easy-to-use interface (see Figure 4). The present version includes, in addition to ProfileNJ, a tool called ParalogyCorrector [32] for correcting orthology relations. ParalogyCorrector takes as input a gene tree and a set of known pairwise orthology relations between genes, which would typically be derived from synteny comparisons, and constructs the tree which is the closest to the input tree according to the RF distance, with the constraint that couples of putative orthologs must be orthologs in the reconciliation (see Method section).

RefineTree can be used in a modular way, according to the user's specifications. It has been designed to be easily extensible to other tools. For example instead of asking the user to input his own orthology relations, tools for inferring putative orthologs can be included.

Results on Ensembl gene trees

The contributions of ProfileNJ are the guarantee of optimality according to a well formulated problem, and the low running time. In particular the low running time allows to run ProfileNJ on the largest databases as Ensembl Compara, containing 63 eukaryotic whole genomes. Gene trees are constructed for 20519 families. In order to quantify the contribution of ProfileNJ and the contribution of methods using other kinds of information as synteny, we compared three sets of trees on the whole database.

- **Ensembl trees**: Trees stored in the Ensembl gene family database (see Method section);
- ProfileNJ **trees**: Trees output by ProfileNJ with unrooted PhyML trees as input (where branches with aLRT support < 0.95 are contracted) and FastDist distance matrices. A single solution is retained for the rooting leading to a minimum weighted reconciliation cost (see Method section);

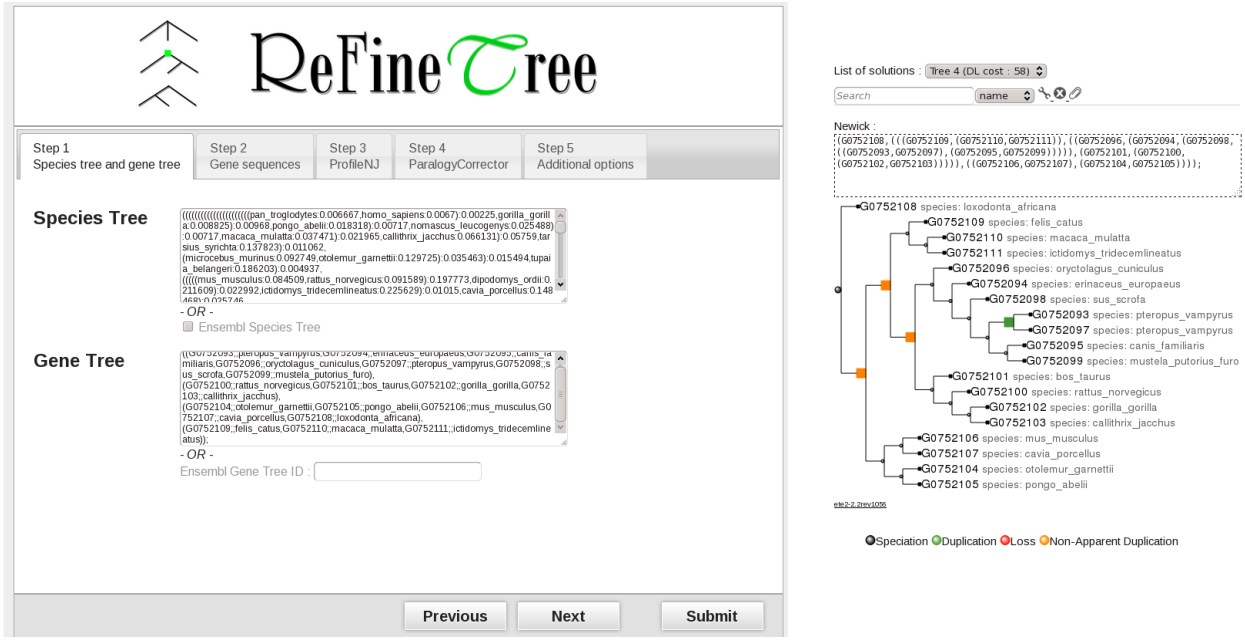

Figure 4. RefineTree web interface. The input is a species tree (or by default the Ensembl species tree) and a gene tree (or an Ensembl gene tree ID), gene sequences and additional options such as the branch contraction threshold, the request to test all rootings, the maximum number of trees to be output by ProfileNJ and sorted by likelihood, etc. The integrated algorithms are ProfileNJ and ParalogyCorrector. Using this second algorithm requires, in addition, the input of a set of orthology constraints.

- **Synteny trees**: Trees output by either ParalogyCorrector or Unduplicator [32] (the two are computed and the most likely according to the sequence is chosen) with ProfileNJ trees as input, using PhylDiag [43] and DeCo [34] to infer synteny constraints (see Method section and Figure 5)

We evaluated the resulting trees according to sequence likelihood, ancestral genome content and ancestral chromosome linearity. The ancestral genome content metric is based on the assumption that the distribution of ancestral gene content sizes should be close to that of extant genomes. Incorrect trees are known to require additional duplications to be reconciled with the species tree, which tends to increase the number of genes in ancestral genomes. The ancestral chromosome linearity metric is based on the assumption that the linearity of ancestral genomes is expected to be as close as possible to that of the extant genomes, with each gene having zero, one or two neighbors, with a peak at two (having genes with zero or one neighbor is usually due to partially assembled genomes).

Results are given in Figure 6. ProfileNJ trees show a better behaviour than Ensembl trees according to the three measures: more than 2/3 of the trees have a better likelihood than Ensembl trees, ancestral genome content distribution is much closer to the extant one, and linearity of chromosomes is higher. Therefore this set of trees, achieving better performance according to sequence evolution, gene content evolution and chromosome evolution, is arguably a better dataset than the one stored in the Ensembl database.

However, results obtained when we include synteny information are less clear. Indeed, quality of synteny trees drops in terms of likelihood (Figure 6 (A)), but jumps in terms of the stability of gene content and the linearity of ancestral chromosomes (Figure 6 (B) and (C)).

Modes of evolution in eukaryotes

Partial patterns of duplications and losses in eukaryotes have been considered in previous studies, as for example by [8] in mammals with a subset of gene families, or by [44] in vertebrates with a subset of species. The ability of ProfileNJ to handle the whole Ensembl database allowed us to perform a more exhaustive study. In addition to gene trees, we reconstructed all ancestral gene contents and organizations. Gene content is computed according to reconciliation (see Methods), and genome organization, which consists in sets of links between consecutive genes, is inferred with DeCo. Genes are not always clustered into full linear genomes. Such non-linearity has diverse causes that we do not wish to mask with an *ad-hoc* linearization method. An interesting property of DeCo is to highlight genes or groups of genes evolving together in parts of the tree. For example 8488 blocks of co-duplicated genes are inferred by DeCo on the considered eukaryote dataset. Most of them contain only a few number of genes (83% contain 2 genes). The largest blocks are found in the terminal branches leading to *Danio rerio* and *Caenorhabditis elegans*.

Figure 7 shows the result for the full genomes of the full phylogeny of the 63 Ensembl species. As seen in Figure 7, duplication rates are highly variable across branches of the phylogeny. Branches with a large number of duplications (hot branches) are those leading to vertebrates, which is in agreement with the two rounds of whole genome duplication hypothesis. Interestingly, the speciation event leading to *Petromyzon marinus*, which is usually thought to have diverged after these events [45], preceeds the hot branches. This may be in agreement with recent results based on the analysis of Hox clusters in the Japanese lamprey [46]. Another hot branch leads to eutherian mammals, which was also found by two other studies [8, 44] with partial data. These two hottest internal branches are exactly the ones found by Mahmudi et al [44] using a probabilistic technique, but using only 9 species due to computational cost. Other hot branches are terminal, the hottest being those leading to *Caenorhabditis elegans* and *Danio rerio*. This is possibly due to ongoing dynamics of polymorphic copy number variations. The same tree showing the number of losses is provided in supplementary material (SI1, Figure S10).

Discussion

ProfileNJ is a new gene tree correction method based on exploring a restricted tree space and choosing the

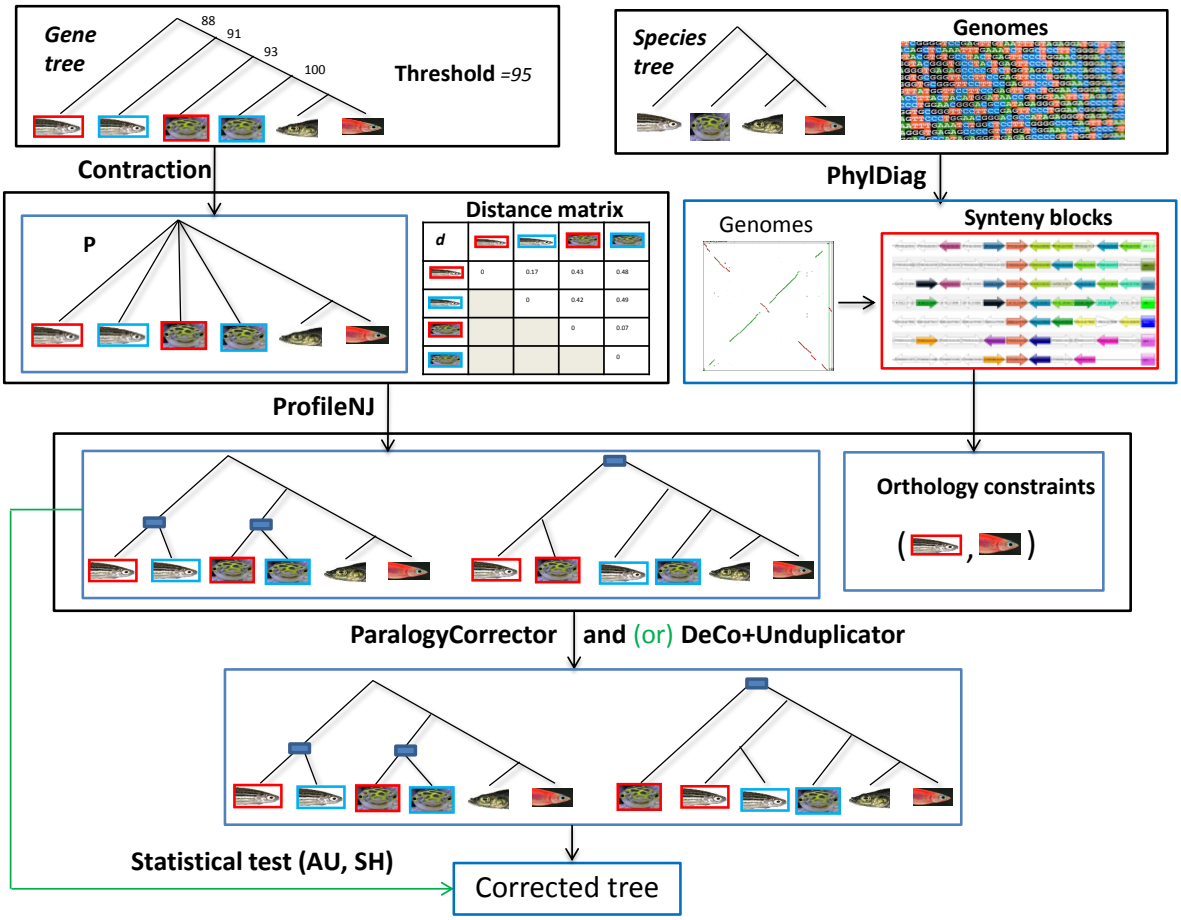

Figure 5. A general view on RefineTree when run on the Ensembl Compara gene families. An example is given for a species tree S of four fish species, a gene family of six genes (a gene is represented by the picture of the species it belongs to, and two paralogs belonging to the same species are distinguished by a different frame color), a rooted gene tree G (although it can be unrooted in general) with branch support, and a given threshold for branch contraction. Data framed in black are the input and those framed in blue are the output of the correction algorithm labeling the edge linking the considered frames. Black arrows depict the use we make of RefineTree on the Ensembl gene trees. The green arrow and the green "or" are alternative uses avoiding one or both of the correction tools ParalogyCorrector and Unduplicator. Any framed set of data can be alternatively provided to the pipeline as input. For example, orthology constraints obtained from various sources can be directly provided as input to ParalogyCorrector. The method for inferring orthology constraints from synteny blocks is described in the text.

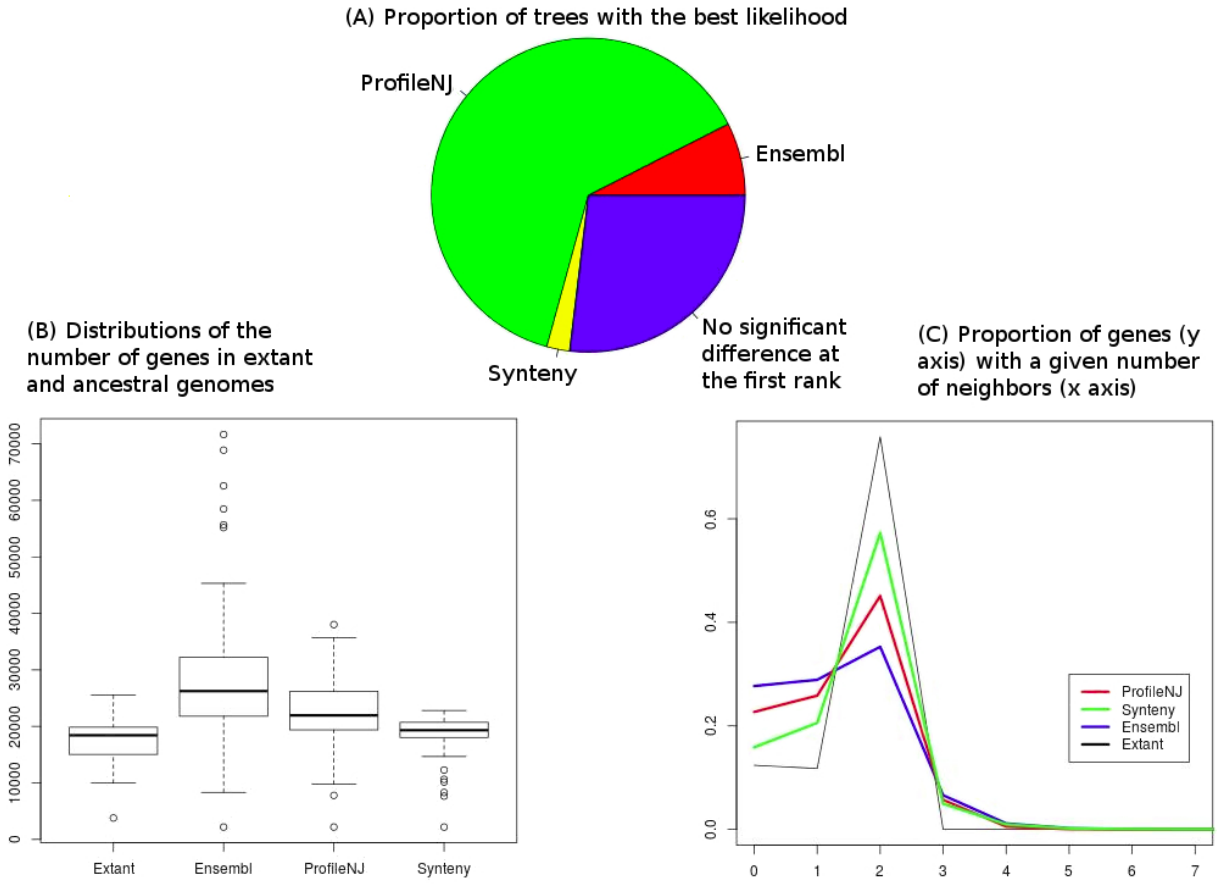

Figure 6. Sequence likelihood, ancestral genome content and ancestral chromosome linearity for ProfileNJ, Synteny and Ensembl trees: **(A)** Proportion of trees with a significantly better likelihood computed with PhyML. AU tests were computed for the three trees for each family, and if the tree at the first rank was significantly better than the second, it was stored as the best likelihood, and if not, it was stored as "no significant difference at the first rank". **(B)** Gene content computed with DeCo. Gene content has one value for each node of the phylogeny of 63 species, except for extant genomes, for which it has one value for each leaf. **(C)** Genome linearity computed with DeCo. Genome linearity is represented by a graph, whose x axis is the number of neighbors a gene can have, and the y axis shows the proportion of genes having this number of neighbors. Parameters from extant genomes are given as a reference in (B) and (C). Statistics for ancestral genomes are assumed better when close to the extant ones.

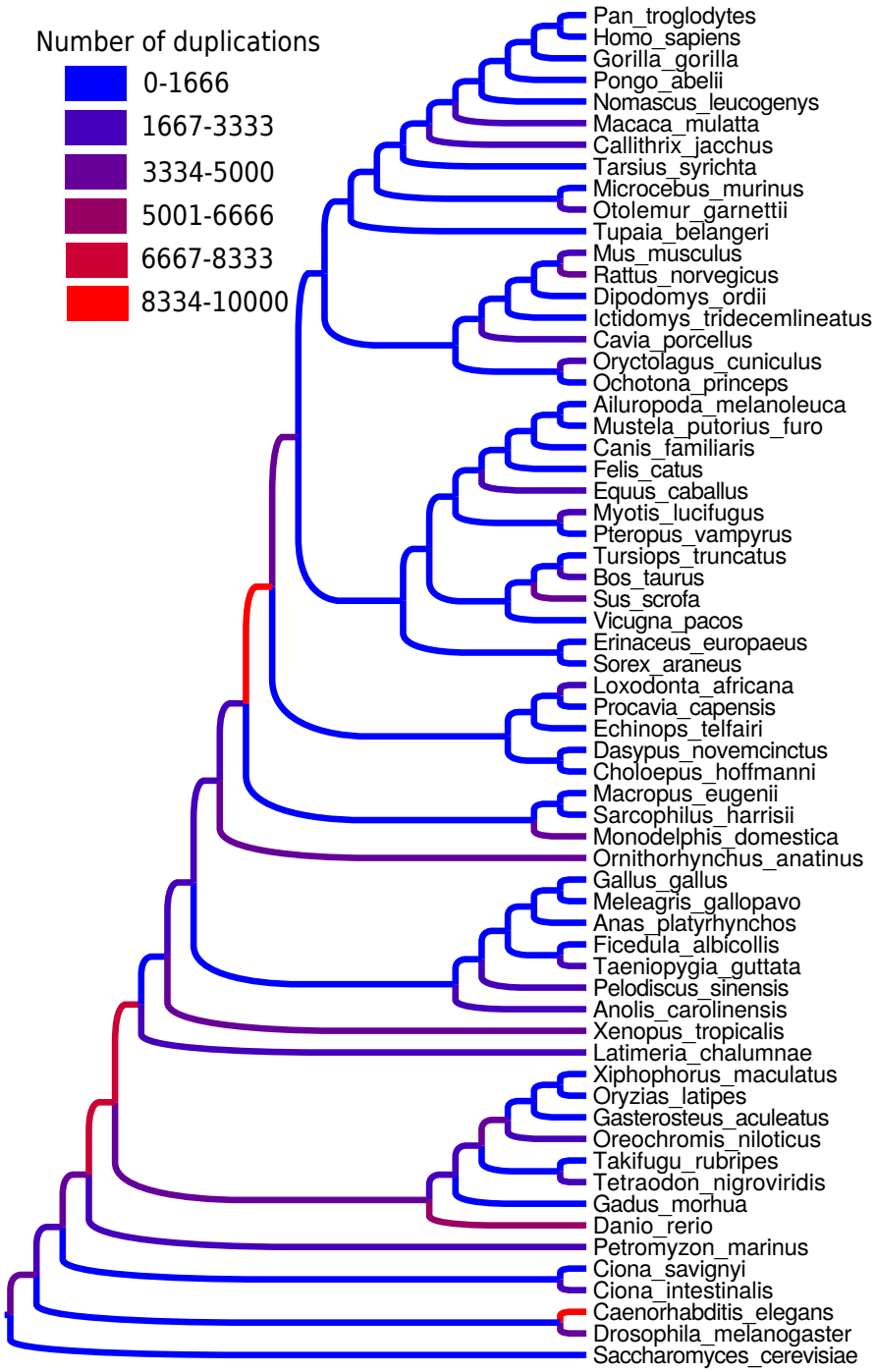

Figure 7. Numbers of duplications in the eukaryote phylogeny, estimated with reconciled ProfileNJ trees from PhyML starting trees on the whole Ensembl Compara database, version 73. Drawn with Figtree [47].

most likely tree according to a species tree and a distance matrix on gene sequences. It is shown to be accurate and it outperforms in running time the most comparable existing correction methods. Efficiency in running time allowed us to apply ProfileNJ to the entire Ensembl database.

Trees obtained by correcting PhyML trees with ProfileNJ are arguably better than gene trees stored in Ensembl, according to sequence likelihood, ancestral genome content and ancestral chromosome linearity. We also corrected directly the Ensembl trees and the results (not shown) were similar, ProfileNJ giving better ancestral genomes and more likely trees than the starting trees. Based on such accurate trees, we have been able to perform an exhaustive study of the patterns of duplication and loss on the phylogeny of the 63 Eukaryote species included in Ensembl.

Accounting for synteny

In addition to sequence and phylogeny, we also used synteny information and orthology relations as correction criteria. As gene trees contain the most complete information about a gene family history, detecting orthologs or studying gene repertoire evolution should be achieved by interpreting trees. But due to the rate of errors in the current trees stored in databases, orthology is often assessed with a series of techniques including synteny [48] and Reciprocal Best Hits, while the evolution of gene repertoires is often studied with phyletic profile techniques [49]. What we presented here is a way of integrating those diverse techniques into a phylogenetic framework.

Unfortunately, integrating synteny information results in a drop of gene tree quality in terms of likelihood. One interpretation is that achieving orthology constraints may require breaking well supported branches. Part of the likelihood drop could also be interpreted as an inadequacy of the sequence evolution models to appropriately account for gene families with a high rate of duplications. Further, as observed in our simulations, the true tree is not necessarily the ML tree. Finally, the likelihood is computed with an alignment that usually results from a guiding tree, estimated using fast but crude approaches, and often different from the tested tree. Some synteny trees might therefore be better trees even in cases where sequence likelihood disagrees, because sequence likelihood can be incorrect.

However there is a third interpretation. Synteny information describes the history of loci [50], while phylogenetic models describe the evolution of sequences. Loci and sequences often have the same history, but they may differ following gene conversion or incomplete lineage sorting (ILS).

In case of ILS or gene conversion, two different true versions of the gene history are concurrent. In Figure 8 the gene as a locus has a history depicted by the right tree, while the gene as a sequence has a history depicted by the left tree. None of the two are wrong, but they are significantly different. They highlight the ambiguity of the definition of a gene, which yields an ambiguity in its history. Sequence trees will have a high likelihood and mediocre results for gene contents and synteny when constructed from duplication and loss scenarios, while it is the opposite for locus trees. A probabilistic model that incorporates ILS in sequence and duplications and losses in loci has been proposed [50]. However, no model is currently able to handle conversion.

Not only the gene trees

Using genome evolution in the construction of the gene trees, we get ancestral genomes as a byproduct. They are made of genes and sets of gene adjacencies. They are still too big (in terms of gene number) and too non linear to be fully trusted. This is partly due to incorrect gene trees in our output, or incorrect inferences from DeCo, but also to problems in sequencing, assembling, annotating genomes, clustering families or inferring the species tree. Good methods for finding linear structures from a set of adjacencies exist [51]. Here we rather used non-linearity as a testimony of the flaws of the data and methods used to reconstruct genome evolution.

Although gene trees are "better" with our correction, they still could be improved. The likelihood drop for synteny correction is indeed surprising, as these corrections lead to ancestral genomes that are closer to

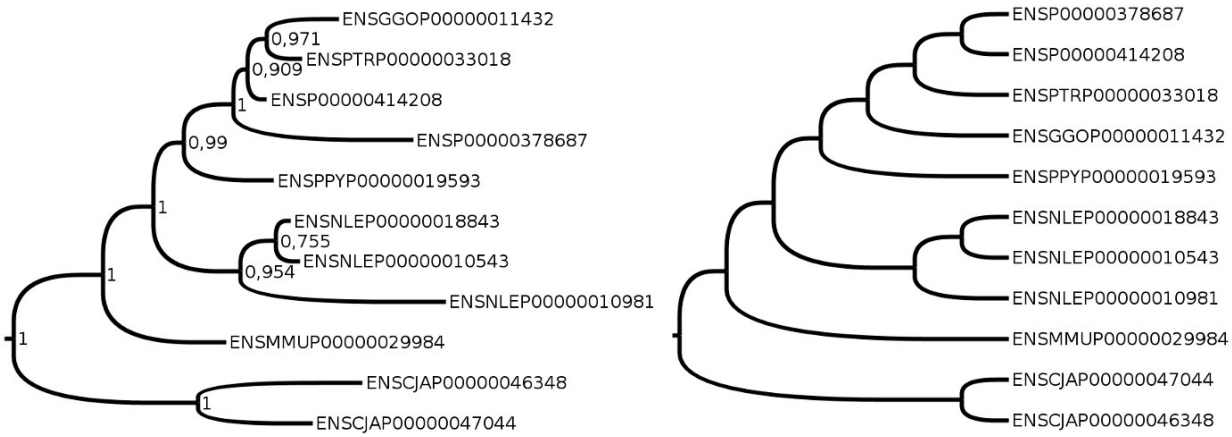

Figure 8. A probable example of ILS visible on a subtree of an ensembl gene family. The monophyly of the chimpanze and gorilla genes (ENSPTRP00000033018 and ENSGGOP00000011432) is well supported by the sequences (left tree, constructed by PhyML, with aLRT supports), while synteny argues for orthology of both with the human genes (ENSP00000414208 and ENSP00000378687) (right tree, constructed by ProfileNJ followed by ParalogyCorrector), so that a scenario of duplications and losses compatible with the left tree is unlikely.

gene content and gene neighborhoods of extant genomes. We would need better exploration schemes with integrated models to really trust gene trees on a whole genome database within a deep phylogeny.

Methods

Families, alignments and trees were taken from Ensembl Compara release 73, sept 2013. They were computed with a pipeline called TreeBest, but we simply call them the "Ensembl trees". Trees are rooted and available with branch support and annotation. There are 20529 trees, each corresponding to a gene family, for a total of 1091891 genes taken from 63 species. Information on gene position on chromosomes, scaffolds or contigs is available at `ftp://ftp.ensembl.org/pub/release-73/emf/ensembl-compara/homologies/`.

Use of ProfileNJ on Ensembl

PhyML was used with default parameters to compute maximum likelihood trees from the protein multiple alignments taken from Ensembl. An aLRT support was computed, and all branches with aLRT < 0.95 were contracted. FastDist was run on DNA alignments to provide a distance matrix. Then ProfileNJ was run with the command (an example is given for the first family).

```
ProfileNJ -s Compara.73.species_tree  \\
          -g data/famille_1.start_tree \\
          -d data/famille_1.dist \\
          -o data/famille_1.tree \\
          -n -r best -c nj --slimit 1 \\
          --plimit 1 --firstbest --cost 1 0.99999
```

We tested the sensitivity of the method to the choice of the threshold parameter for contracting unsupported branches. The threshold is a trade-off between the amount of change in a tree and the

probability that the resulting tree is rejected. Values that are too high would avoid exploring a large space around the starting tree while small values would lead to low likelihood trees. It has to be settled empirically. For example .80 was considered an acceptable threshold in some genomic studies [52].

Ancestral Genomes (gene content and order) from the LCA Reconciliation

For a rooted binary gene tree G, we suppose there is a mapping m from its leaves to the leaves of a rooted binary species tree S. The *reconciliation* of G with S consists in extending m to the internal nodes v of G by letting $m(v) = lca(m(L(G_v)))$, where G_v is the subtree of G rooted at v, $L(G_v)$ is its leaves, $m(L(G_v)))$ is the set of leaves of S associated to $L(G_v)$ by m, and lca is their lowest common ancestor in S. In addition an *event* is associated to each internal node v of G: it is a duplication if v has a child which also maps to $m(v)$ by m, and it is a speciation otherwise. For any branch ab (suppose a is an ancestor of b) of G with $m(a) \neq m(b)$, let $m(a), s_1, \ldots, s_l, m(b)$ be the path joining $m(a)$ and $m(b)$ in S. Subdivide the branch ab by adding l nodes, respectively mapped to $s_1, \ldots, s_l$ by m. If a is a duplication, add an additional node between a and its direct descendant (s_1 if it exists or b), mapped to $m(a)$.

Then we can define the number of duplications induced by G, which is the number of duplication nodes, the number of losses, which is the number of nodes with one descendant, and the genes in $s \in S$: each node mapped to s but whose parent is not mapped to s is considered as a gene.

Testing the linearity of ancestral genomes with DeCo

DeCo [34] computes ancestral gene neighborhoods that are highly dependant on both the shape of the considered gene tree and extant gene neighborhoods. Indeed, adjacencies in extant genomes, *i.e.* the immediate proximity of two consecutive genes, are taken as input and putative adjacencies in ancestral genomes are constructed by a parsimony principle minimizing the number of gains and losses of adjacencies. As two contemporaneous adjacencies are supposed to evolve independently one from the other, the linearity of extant genomes, *i.e.* the property that one gene never has more than two neighbors linked by an adjacency, does not guarantee the linearity of ancestral ones.

The apparent weakness of this feature is in fact a strength to evaluate the quality of gene trees. Indeed, a high part of the non linearity of ancestral genomes is not due to the inadequacy of the software itself, but to the quality of the input data. It has been remarked that a significant improvement in the linearity of ancestral genomes was obtained by constructing gene trees according to more integrative models [8, 53].

Note that in extant genomes, no gene can have more than two neighbors, and most genes have two. But many genes have 1 or 0, because of the poor assembly of some genomes, many contigs contain one or a few genes.

Information from extant synteny

Orthology constraints are inferred as follows. If several genes are found consecutive in one genome, and their homologs are also found consecutive in the other genome, the common linear arrangement was in the ancestor and the homologous genes are probably orthologous. This hypothesis is incorrect in at least three cases : (1) if the whole block of genes was duplicated, (2) if there is a tandem duplication of a gene followed by a differential loss in the two species, or (3) if a gene is converted by a paralog. To handle these cases, we require that (1) the majority of the homologous genes are indeed predicted as orthologs by phylogeny, (2) the common ancestor of two homologous genes does not lead to two paralogous descendants placed in tandem in one species. In case (3), we are in a situation where the loci are orthologous but not the sequences. In that case we construct the "locus tree" [50] and trust syntenic information over gene sequence information.

First we ran PhylDiag as follows, for each pair of genomes. Files genome_1, genome_2 and ancestral_genes respectively contain the ordered list of genes from each genome, and the list of families clustering the genes as in the Ensembl database.

```
phylDiag.py genome_1 genome_2 ancestral_genes \\
    -gapMax=2 -pThreshold=0.00000005 \\
    -filterType=InBothSpecies -multiprocess \\
    -minChromLength=2 >syntenyblocks_1_2
```

The statistical threshold is calculated in order to minimize the number of false positives, taking into account the number (2211) of comparisons between pairs of species and the expected number (500) of synteny blocks for each comparison ($0.05/(2211 * 500) \approx 5e-8$).

For each synteny block found by PhylDiag, we kept only the genes that had one single exemplar in the two blocks from both species. We counted the number of such pairs of genes, and refered to an LCA reconciliation of the output trees of ProfileNJ to check that most pairs are orthologs (their common ancestor is labeled by a speciation). We discarded the blocks that did not fit this condition. This discards possible block duplications.

For the remaining blocks, and for each couple of uniquely represented genes a and b, we required that the LCA node X of a and b in the reconciled ProfileNJ tree is not a supported duplication: let X_1 and X_2 be the two children of the node X labeled as a duplication (so X_1 and X_2 are in the same species as X), the genes a and b are not kept as putative orthologs if one of the branches XX_1 and XX_2 has a high support (> 0.95), and there are two genes, x_1 and x_2, which respectively descend from X_1 and X_2, which are located on the same genome. This discards possible tandem duplications in the block, followed by differential losses of copies.

The output trees from ProfileNJ as well as the filtered pairs of putative orthologs were given as input to ParalogyCorrector, which finds the tree that is as close as possible to the input tree in terms of RF distance, such that in an LCA reconciliation, all pairs of putative orthologs have an LCA node annotated as a speciation.

Information from ancestral synteny

From the results of DeCo on the output gene trees produced by ProfileNJ, we used an "unduplication" principle as in [33] everytime we found that an ancestral gene x had three neighbors a, b, c, two of them (say a, b) arising from a duplication node d in a single gene tree. In that case, we rearranged the four grand children of d so that the clade under d has an LCA which is annotated as a speciation in the LCA reconciliation. See an insight into its functionning in Figure 9.

Likelihood ratio tests

We computed the likelihood of all trees according to the HKY85 model with PhyML on nucleotide alignments. To test the significance of a likelihood difference, we computed the AU (Appoximately unbiased) tests with RAxML. They consist in bootstraping the sites of an alignment, each site having a likelihood according to several trees. Then a probability is associated to each tree from this bootstrap, according to the number of replicates which place it above the others in terms of the bootstrapped likelihood. Unless otherwise stated, we use "significantly" better for a likelihood with a AU value > 0.95. Tests were ran with Consel [54].

Data access

The 2575 simulated gene families used for our simulation represent a subset of the original SPIMAP simulated fungi datasets (see http://compbio.mit.edu/spimap/). Those data and the RAxML trees constructed from sequence alignment are available. We also provide the two sets of 20529 trees, as an output from ProfileNJ and with the additional synteny-aware corrections. All software are freely accessible for academic purpose, under a GPL license.

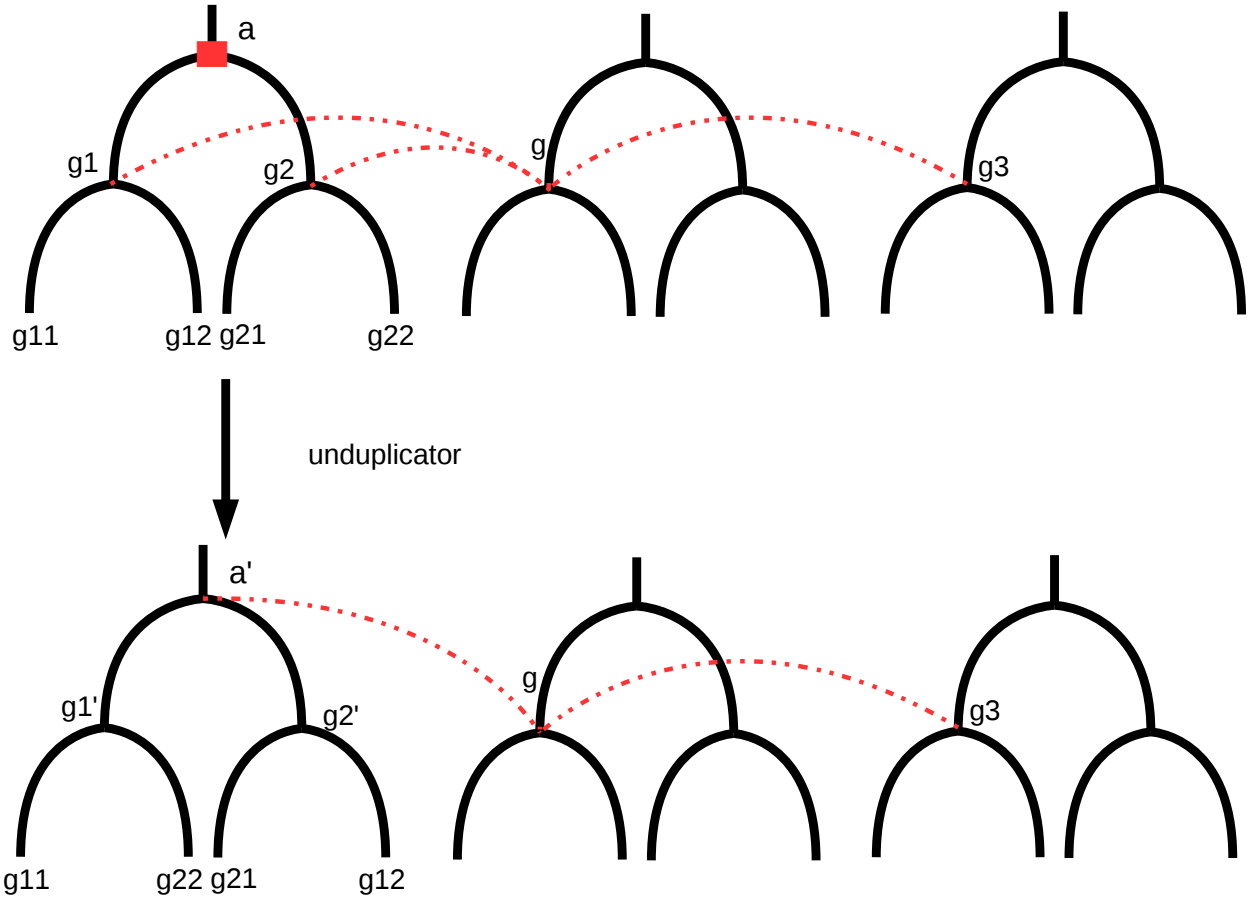

Figure 9. The unduplication principle (figure redrawn from [33]). A non linearity is detected in an ancestral genome (gene g has three neighbors). Two of its neighbors g_1 and g_2 are issued from a possibly dubious duplication labeled node. The tree is rearranged so that its root is labeled with a speciation instead of a duplication. In the resulting configuration g_1' and g_2' are in two different species, so that g can have only one neighbor in this family, and linearity is recovered.

Acknowledgments

MS, LG, BB and ET were supported by the French Agence Nationale de la Recherche (ANR) through Grant ANR-10-BINF-01-01 "Ancestrome". EN, ML, JS and NEM were supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the "Fonds de recherche du Québec Nature et technologies" (FRQNT) of Quebec. Computations were made on the supercomputer "Briarée" from Université de Montréal, managed by Calcul Québec and Compute Canada.

Supporting information

SI file 1: Additional validity and robustness tests of ProfileNJ, followed by a representation of the number of losses along a phylogeny inferred from ProfileNJ trees.

References

1. Vilella AJ, Severin J, Ureta-Vidal A, Heng L, Durbin R, Birney E. EnsemblCompara gene trees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Research. 2009;19:327-335.

2. Penel S, Arigon AM, Dufayard JF, Sertier AS, Daubin V, Duret L, et al. Databases of homologous gene families for comparative genomics. BMC Bioinformatics. 2009;10 Suppl 6:S3. Available from: http://dx.doi.org/10.1186/1471-2105-10-S6-S3.

3. Datta RS, Meacham C, Samad B, Neyer C, Sjölander K. Berkeley PHOG: PhyloFacts orthology group prediction web server. Nucleic Acids Research. 2009;37:W84-W89.

4. Pryszcz LP, Huerta-Cepas J, Gabaldón T. MetaPhOrs: orthology nd paralogy predictions from multiple phylogenetic evidence using a consistency-based confidence score. Nucleic Acids Research. 2011;39:e32.

5. Huerta-Cepas J, Capella-Gutierrez S, Pryszcz LP, Denisov I, Kormes D, Marcet-Houben M, et al. PhylomeDB v3.0: an expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. Nucleic Acids Research. 2011;39:D556-D560.

6. Mi H, Muruganujan A, Thomas PD. PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. Nucleic Acids Research. 2012;41:D377-D386.

7. 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 Sep;12(5):423–435. Available from: http://dx.doi.org/10.1093/bib/bbr034.

8. Boussau B, Szöllősi GJ, Duret L, Gouy M, Tannier E, Daubin V. Genome-scale coestimation of species and gene trees. Genome Research. 2013;23:323-330.

9. Flicek P, Amode MR, Barrell D, Beal K, Billis K, Brent S, et al. Ensembl 2014. Nucleic Acids Research. 2014 Jan;42(Database issue):D749–D755. Available from: http://dx.doi.org/10.1093/nar/gkt1196.

10. Guindon S, Gascuel O. A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic Biology. 2003;52:696- 704.

11. Stamatakis A. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analysis with thousands of taxa and mixed models. Bioinformatics. 2006;22:2688-2690.

12. Ronquist F, Huelsenbeck JP. MrBayes3: Bayesian phylogenetic inference under mixed models. Bioinformatics. 2003;19:1572- 1574.

13. Lartillot N, Philippe H. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Molecular Biology and Evolution. 2004 Jun;21(6):1095–1109. Available from: http://dx.doi.org/10.1093/molbev/msh112.

14. 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.

15. Wu YC, Rasmussen MD, Bansal MS, Kellis M. TreeFix: Statistically informed gene tree error correction using species trees. Systematic Biology. 2013;62(1):110- 120.

16. Durand D, Haldórsson BV, Vernot B. A hybrid micro-macroevolutionary approach to gene tree reconstruction. Journal of Computational Biology. 2006;13:320–335.

17. Zimmermann T, S M, Warnow T. BBCA: Improving the scalability of BEAST using random binning. BMC Genomics. 2014;15(Suppl 6):S11. Proceedings of RECOMB-CG.

18. Szöllősi GJ, Rosikiewicz W, Boussau B, Tannier E, Daubin V. Efficient exploration of the space of reconciled gene trees. Systematic Biology. 2013 Nov;62(6):901–912. Available from: http://dx.doi.org/10.1093/sysbio/syt054.

19. Akerborg O, Sennblad B, Arvestad L, Lagergren J. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. Proceedings of the National Academy of Sciences USA. 2009;106(14):5714–5719.

20. Arvestad L, Berglund AC, Lagergren J, Sennblad B. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In: RECOMB; 2004. p. 326-335.

21. Rasmussen MD, Kellis M. A bayesian approach for fast and accurate gene tree reconstruction. Molecular Biology and Evolution. 2011;28(1):273- 290.

22. Thomas PD. GIGA: a simple, efficient algorithm for gene tree inference in the genomic age. BMC Bioinformatics. 2010;11:312.

23. Nguyen TH, Ranwez V, Pointet S, Chifolleau AMA, Doyon JP, Berry V. Reconciliation and local gene tree rearrangement can be of mutual profit. Algorithms for Molecular Biology. 2013;8(1):12. Available from: http://dx.doi.org/10.1186/1748-7188-8-12.

24. Chen K, Durand D, Farach-Colton M. Notung: Dating Gene Duplications using Gene Family Trees. Journal of Computational Biology. 2000;7:429–447.

25. Gorecki P, Eulenstein O. A linear-time algorithm for error-corrected reconciliation of unrooted gene trees. In: ISBRA. vol. 6674 of LNBI. Springer-Verlag; 2011. p. 148-159.

26. Gorecki P, Eulenstein O. Algorithms: simultaneous error-correction and rooting for gene tree reconciliation and the gene duplication problem. BMC Bioinformatics. 2011;13(Supp 10):S14.

27. Chaudhary R, Burleigh JG, Eulenstein O. Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence. BMC Bioinformatics. 2011;13(Supp.10):S11.

28. Berglund-Sonnhammer AC, Steffansson P, Betts MJ, Liberles DA. Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. Journal of Molecular Evolution. 2006;63:240-250.

29. Doroftei A, El-Mabrouk N. Removing Noise from Gene Trees. In: WABI. vol. 6833 of LNBI/LNBI; 2011. p. 76-91.

30. Swenson KM, Doroftei A, El-Mabrouk N. Gene Tree Correction for Reconciliation and Species Tree Inference. Algorithms for Molecular Biology. 2012;7(1):31.

31. Lafond M, Swenson KM, El-Mabrouk N. An Optimal Reconciliation Algorithm for Gene Trees with Polytomies. In: LNCS. vol. 7534 of WABI; 2012. p. 106-122.

32. Lafond M, Semeria M, Swenson KM, Tannier E, El-Mabrouk N. Gene tree correction guided by orthology. BMC Bioinformatics. 2013;14 (supp 15)(S5).

33. Chauve C, El-Mabrouk N, Guéguen L, Semeria M, Tannier E. Duplication, Rearrangement and Reconciliation: A Follow-Up 13 Years Later. In: Chauve C, El-Mabrouk N, Tannier E, editors. Models and Algorithms for Genome Evolution. London: Springer; 2013. p. 47–62.

34. Bérard S, Gallien C, Boussau B, Szöllősi GJ, Daubin V, Tannier E. Evolution of gene neighborhoods within reconciled phylogenies. Bioinformatics. 2012 Sep;28(18):i382–i388. Available from: http://dx.doi.org/10.1093/bioinformatics/bts374.

35. Semple C, Steel M. Phylogenetics. Oxford Univ Press; 2003.

36. Kluge AG, Farris JS. Quantitative phyletics and the evolution of anurans. Syst Zool. 1969;18:1–32.

37. Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution. 1987;4:406-425.

38. Maddison WP. Gene trees in species trees. Syst Biol. 1997;46:523–536.

39. Lafond M, Noutahi E, El-Mabrouk N. Efficient Non-binary Gene Tree resolution with Weighted Reconciliation Cost; 2016. Submitted.

40. Gascuel O, Steel M. Neighbor-joining revealed. Mol Biol Evol. 2006 Nov;23(11):1997–2000. Available from: http://dx.doi.org/10.1093/molbev/msl072.

41. Lafond M, Chauve C, Dondi R, El-Mabrouk N. Polytomy refinement for the correction of dubious duplications in gene trees. Bioinformatics. 2014 Sep;30(17):i519–i526. Available from: http://dx.doi.org/10.1093/bioinformatics/btu463.

42. Khan MA, Elias I, Sjölund E, Nylander K, Guimera RV, Schobesberger R, et al. Fastphylo: fast tools for phylogenetics. BMC Bioinformatics. 2013;14:334. Available from: http://dx.doi.org/10.1186/1471-2105-14-334.

43. Lucas JM, Muffato M, Roest Crollius H. PhylDiag: identifying complex synteny blocks that include tandem duplications using phylogenetic gene trees. BMC Bioinformatics. 2014;15(1):268. Available from: http://dx.doi.org/10.1186/1471-2105-15-268.

44. Mahmudi O, Sjöstrand J, Sennblad B, Lagergren J. Genome-wide probabilistic reconciliation analysis across vertebrates. BMC Bioinformatics. 2013;14 Suppl 15:S10. Available from: http://dx.doi.org/10.1186/1471-2105-14-S15-S10.

45. Smith JJ, Kuraku S, Holt C, Sauka-Spengler T, Jiang N, Campbell MS, et al. Sequencing of the sea lamprey (Petromyzon marinus) genome provides insights into vertebrate evolution. Nature genetics. 2013;45(4):415–421.

46. Mehta TK, Ravi V, Yamasaki S, Lee AP, Lian MM, Tay BH, et al. Evidence for at least six Hox clusters in the Japanese lamprey (Lethenteron japonicum). Proceedings of the National Academy of Sciences. 2013;110(40):16044–16049. Available from: http://www.pnas.org/content/110/40/16044.abstract.

47. Rambaut A. Figtree; 2006. http://tree.bio.ed.ac.uk/software/figtree/.

48. Sonnhammer ELL, Gabaldón T, Sousa da Silva AW, Martin M, Robinson-Rechavi M, Boeckmann B, et al. Big data and other challenges in the quest for orthologs. Bioinformatics. 2014 Jul;p. btu492.

49. Cohen O, Ashkenazy H, Burstein D, Pupko T. Uncovering the co-evolutionary network among prokaryotic genes. Bioinformatics. 2012 Sep;28(18):i389–i394. Available from: http://dx.doi.org/10.1093/bioinformatics/bts396.

50. Rasmussen MD, Kellis M. Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome Research. 2012 Apr;22(4):755–765. Available from: http://dx.doi.org/10.1101/gr.123901.111.

51. Maňuch J, Patterson M, Wittler R, Chauve C, Tannier E. Linearization of ancestral multichromosomal genomes. BMC Bioinformatics. 2012;13 Suppl 19:S11. Available from: http://dx.doi.org/10.1186/1471-2105-13-S19-S11.

52. Abby SS, Tannier E, Gouy M, Daubin V. Lateral gene transfer as a support for the tree of life. Proceedings of the National Academy of Sciences USA. 2012 Mar;109(13):4962–4967.

53. Patterson M, Szöllősi G, Daubin V, Tannier E. Lateral gene transfer, rearrangement, reconciliation. BMC Bioinformatics. 2013;14 Suppl 15:S4. Available from: http://dx.doi.org/10.1186/1471-2105-14-S15-S4.

54. Shimodaira H, Hasegawa M. CONSEL: for assessing the confidence of phylogenetic tree selection. Bioinformatics. 2001;17(12):1246–1247. Available from: http://bioinformatics.oxfordjournals.org/content/17/12/1246.abstract.