Scaling the gene duplication problem towards the Tree of Life: Accelerating the rSPR heuristic search

André Wehe[1] and J. Gordon Burleigh[2]

[1]Department of Computer Science, Iowa State University, Ames, IA USA, {awehe@iastate.edu}

[2]Department of Biology, University of Florida, Gainesville, FL USA, {gburleigh@ufl.edu}

Abstract

The gene duplication problem seeks the species tree that implies the fewest duplications across a collection of gene trees. We describe two new improvements to existing heuristics that greatly accelerate the rSPR local search. The improvements involve (i) reducing the computational complexity by $\Theta(n/h)$ for sequential evaluations of the reconciliation cost on n taxa and a tree height of h, and (ii) a lightweight and effective heuristic search strategy. In analyses of both empirical and simulated data sets, the new improvements display tremendous speedup from the best existing heuristics with no apparent loss of accuracy. We also demonstrate with simulated data that the improved heuristics can easily compute species trees with 100,000 taxa on a single desktop processor.

1 Introduction

Phylogenetic trees can be powerful tools for examining patterns of biodiversity [1] and the structure of communities and ecosystems [2] as well as comparative analyses of species and character evolution and diversification [3, 4]. Performing such analyses on a macro-evolutionary or ecological scale often requires extremely large phylogenies. There is great interest in developing methods that can synthesize new molecular data into extremely large-scale phylogenetic hypotheses, and ultimately the tree of life. However, large-scale phylogenetic inference presents several enormously challenging computational problems, including the limitations of available memory and the computational burden of efficient heuristics [5]. In this paper, we present two new approaches that greatly speed up the inference of large-scale phylogenetic trees based on gene tree species tree reconciliation, allowing phylogenetic analyses on a unprecedented scale.

Maximum parsimony (MP) and maximum likelihood (ML) are among the most popular methods for inferring trees. Both take an alignment of gene sequences and attempt to identify the optimal tree based on an optimality criterion, either the tree that implies the fewest character substitutions (MP) or the tree that maximizes the probability of observing the data based on a model of evolution (ML). Both MP and ML methods have been used to build phylogenetic trees with tens of thousands of taxa (e.g., [6, 7]). However, these methods have some limitations for large-scale phylogenetics. First, the size of the character matrices can create an enormous memory burden that, in practice, limits the number of genes that can be incorporated into an analysis. For example, if the average gene is conservatively 1000 base pairs (bp) long, a character matrix with 100,000 taxa and just 10 genes would include 1 billion cells and 1000 genes would be 100 billion cells. Second, both MP and ML implicitly assume that all of the genes share a common evolutionary history. However, evolutionary processes such as gene duplications and losses, incomplete lineage sorting (or deep coalescence), horizontal transfer, and recombination can affect the topologies of gene trees, producing incongruence between the gene tree and species tree topologies [8]. This incongruence among gene trees can produce error in MP and ML analyses (e.g., [9, 10]).

One approach to inferring species trees from genes with complex evolutionary histories is gene tree parsimony (GTP), which seeks to identify the species tree that implies the minimum number of events that cause conflict among the gene trees (e.g., [11, 8]). Specifically, GTP takes a collection of gene trees and seeks a species tree that contains all taxa represented in the gene trees and implies the fewest gene duplications, duplications and losses, or deep coalescence events (e.g., [12]). Thus, unlike MP or ML, GTP does not assume a single underlying phylogeny for gene trees. Furthermore, since the input for GTP analysis is a set of trees rather than gene alignments from each gene, GTP has a much lower memory burden than MP or ML. However, GTP is an NP-hard problem (e.g., [13]), and analyses with more than ≈15 taxa require heuristic solutions.

Recent improvements in the speed of local search heuristics have allowed GTP to be applied to genome-

scale data sets with hundreds of taxa (e.g., [14, 15]). For example, a recent GTP analysis included 18,896 gene trees from 131 plant taxa [16]. Yet this is still far from the tree of life. We present new improvements to GTP heuristics that in practice produce enormous speedup in the existing heuristic with no apparent loss of accuracy. These new heuristic approaches enable GTP analyses to incorporate large sections of the tree of life from large proportions of the genome.

2 Notations

We consider only rooted, binary trees T, where $V(T)$ are the nodes and $E(T)$ the edges of a tree T. We denote a species tree and gene tree with S and G respectively. Each terminal in S is uniquely labeled with a taxon name, among a total of n taxa, and every terminal in the gene trees maps to one of of these taxa. A subtree X of a tree T can be any connected subgraph of T where the root of X is the node closest to $root(T)$, whereas a clade, denoted as T_v, is the subtree induced by an interior node v and all of its descendants. The root of a subtree or tree is denoted with $root(T)$. All ancestral nodes of a node v are denoted with $ancestor(v)$, i.e. all nodes along the path from v to the $root(T)$ where T is the tree, and the parent of a node v is its immediate ancestor $parent(v) = x \mid x \in ancestor(v)$ *and* $(v, x) \in E(T)$. The tree rearrangement operation rSPR (rooted subtree prune and regraft [17, 18]) is informal: relocating a clade into a different edge or to the root in a rooted tree (e.g. Figure 1). We will perform these rSPR operations exclusively on species trees, and denote with $r\mathcal{SPR}_S(v)$ the set of all rSPR operations such that S_v is the pruned clade. The LCA (lowest or most recent common ancestor) of two nodes is $lca(u, v)$. A LCA mapping between a gene tree G and a species tree S is a node mapping between both trees induced by the genetic terminal mapping and the LCAs in S. The node mapping is denoted with $s = map(g)$ where g is a gene tree node and s the corresponding species tree node.

3 Sequential Search Steps

We first describe a computational speed up for the rSPR heuristic search that is designed to take advantage of the characteristics of typical empirical data sets. Our heuristic search objective is the minimum number of gene duplications. We give an algorithm that updates prior results for sequential evaluation steps. The rSPR heuristic search is a stepwise process of optimizing a species tree, often called the guidance tree, where the rSPR defines the local search neighborhood of species trees which are evaluated for a better reconciliation cost (lower gene dupliaction cost). The fundamental computational steps are to compute (1) the LCA for the guidance tree, (2) the mapping between gene trees and guidance tree, and (3) the gene duplication cost. Step 1 can be solved in linear time to the species tree size $V(S)$ using RMQ (Range Minimum Query) [19]. Typically step 2 and 3 are the most computationally costly ones, since the gene tree input size $m = \sum_{i=1}^{k} |V(G_i)|$ by far excels the species tree one $V(S)$. Step 3 merely requires mapping changes to update the gene duplication cost.

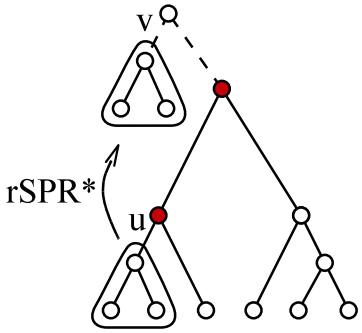

Figure 1: A rSPR* operation on a species tree where a clade is regrafted to the root. The red nodes indicates the set A where all mapping changes occur.

The computational complexity for the mapping is linear to m, and in practice can be computed initially quickly even for huge trees, but not for frequent sequential rSPR operations. In theory, all node mappings can be altered by a single rSPR operation on the guidance tree. However, in practice it is likely that only a minor partition of node mappings, denoted with set L, is actually altered and $|L| \ll m$. This is due to the fact that multiple gene duplications occur throughout the species tree, and the species tree nor guidance tree is most likely not completely pectinate.

Problem 1. Given a LCA mapping, predicting the altering node mappings when applying a rSPR operation on the species tree.

By solving this problem efficiently we can simply recompute the LCA on the guidance tree and all predicted altering node mappings to obtain the new LCA mapping between a new guidance tree and the gene trees.

The mapping between every pair of G_i and S can be computed independently, so without loss of generality we simply investigate changes for a single gene tree G_i only.

Definition 2. A gene tree node g is colored blue when the node mapping $map(g)$ is altered for any rSPR op-

eration on the prune clade P, otherwise the node is colored white.

Figure 2 shows an example gene tree where only the node mappings from the blue indicated nodes change for some rSPR operation on the species tree S, and all other node mappings remain unchanged. These blue nodes are of particular interest, whether for a particular rSPR operation the set of blue nodes must not necessarily be identical to the set L, in fact the set of blue nodes is often a superset of L.

Now, there is a direct relationship between the node mappings of a gene tree node's descendants and the color of a node.

Lemma 3. *A blue node has a descendant mapping to a node in P and a node in $S \setminus P$.*

All blue nodes can be identiied quickly considering the following two properties: (i) The blue nodes always form a single connected subtree within the gene tree G and include the root (e.g. indicated in Figure 2), this can be derived from the heritage of cascading clades, where the property of Lemma 3 is preserved for any clade containing at least one blue node. (ii) The node color can indirectly be determined from its current node mapping, that is, the node mapping before a rSPR operation, by tolerating situational false-positives. Every blue node's mapping must map to a red node, indicated in Figure 1, whose form the set $A = ancester(root(P))$, because all depend on descendant's mappings into P. By precomputing A we can lookup in constant time if $map(g)$ is contained in the set or not. Other white nodes can also map into A and these are counted as false-positives. Later in this section we discuss how limited likely a false-positve is.

Outlined, our algorithm is identifying the blue subtree including false-positives, by traversing G from the root. Afterward the node mapping is recomputed in the post-order of the subtree.

Applicability – Although our algorithm works for any type of rSPR operation, a special operation rSPR* is of most importance for modern heuristic searches, since this operation is exclusively been used in the search algorithm [14] and the program DupTree [20].

Definition 4. A rSPR* operation is a rSPR operation where either u or v, respectively the prune and regraft edge, is connected to the root of the species tree.

Figure 1 shows a rSPR* operation where a clade is moved to the root of S. These rSPR* operations can efficiently be solved with our algorithm, because for any rSPR* operation the set of blue nodes is identical to L. Let us check the rSPR* where v is connected to the root of S, then prior to the rSPR* operation all

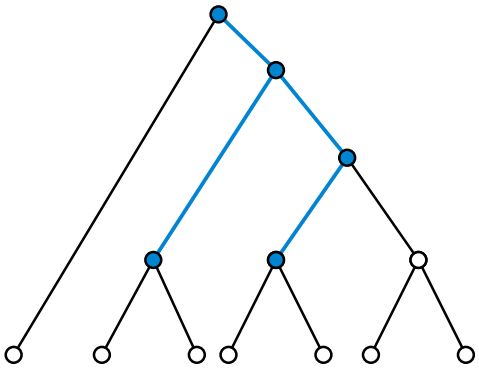

Figure 2: The blue nodes forming a subtree within a gene tree, indicating possible node mapping change.

blue node's mappings go to some species node in S. Now the rSPR* to the root introduces a new node, the root node, connected directly to P and the remaining tree. By lemma 3 any blue node's mapping depends directly or subsequently on a mapping in P, thus all blue node's mappings will map to the new root node. For symetry reasons the rSPR* where u is connected to the root of S is the inverse of the former rSPR*. That means for any rSPR* operation, the blue nodes exactly identify the set of node mappings that will change.

Complexity – The exact run time of our algorithm correlates to the actual mapping changes. Its complexity is bounded by $O(b + |A|)$ where b is the number of nodes identified as being blue. In the worst case this can be as bad as $O(m + n)$, but that being argued is not the typical situation, and in the best case the run time is bounded by $\Theta(k + n)$. In a heuristic search we methodically evaluate rSPR operations on all clades in S, and we can give an amortized complexity for the search. A more or less homogeneous occurrence of gene duplications is likely throughout the time line even for the suboptimal guidance trees, so with a probability of $P(map(g) \in A) = |A|/(2n - 1)$ a mapping is going into $|A|$. This implies the median blue subtree is of size $b = m\ P(map(g) \in A)$. A node can mistakenly be identified as blue with the same probability $P(map(g) \in A)$, so the number of mapping changes is to be expected not less than $b/2$. With $|A| \leq h$ being bounded by the height of the guidance tree we can bound the complexity for the sequential rSPR evaluations on average with $O(n+m\ h/n)$, and the amortized complexity for a complete rSPR* neighborhood evaluation with $O(n^2 + m\ h)$, this is a $\Theta(n/h)$ speedup compared to DupTree [20].

Algorithm 1 Lightweight heuristic search.

1. Let Q be a round robin queue filled with $V(S)$ in random order.
2. Let node v be the next node in Q.
3. If rSPR(S_v) is valid,
   then compute the rSPR(S_v) neighborhood.
4. If found, apply most optimizing rSPR on S.
5. Goto step 2, unless step 4 failed $|Q|$ times.

4 Heuristic search

We also can achieve a large speedup in practice with the computational lightweight hill climbing heuristic shown in algorithm 1. This heuristic obtains its speed from optimizing the species tree without necessarily following the steepest descent. The lightweight heuristic dynamically partitions the rSPR-neighborhood into smaller $r\mathcal{SPR}_S$-neighborhoods and optimizes the guidance tree whenever possible.

5 Experiments

Empirical Data Sets – We first tested the new heuristic and compared it against the heuristic implemented in DupTree [20] using three empirical data sets of very different sizes. First are gene trees from a published study of green plants (18896 gene trees; 136 total species; [16]). We also made sets of gene trees for the gymnosperm and Saxifragales plant clades. These trees were made by downloading from GenBank (`http://www.ncbi.nlm.nih.gov`) the core nucleotide sequences from the specified clade and appropriate outgroups. The sequences were clustered into sets of homologs based on BLAST scores [21] and the clusters were aligned using MUSCLE [22]. The gene trees were inferred using ML implemented in RAxML [23] and rooted using the outgroup taxa. The gymnosperm data set has 77 gene trees and 950 total species, and the Saxifragales data set has 51 gene trees and 958 total species. For each data set, we ran both our new heuristic and DupTree 100 times. All analyses were performed on an Intel®Xeon®2.53GHz processor.

The fastest runs for the new heuristic ranged from 28x faster than DupTree in the green plant data set to over 200x faster in the gymnosperm data set (Table 1). Both heuristics appeared to quickly find an optimal topology for the green plant data set. In all the data sets except the green plants, the score of the best tree varied among runs, suggesting that the rSPR search can get trapped in local optima. However, in 100 runs there was no evidence that our improved heuristic found worse trees than DupTree (Table 1). In fact, in the Saxifragales data set, the improved heuristic found trees with lower reconciliation cost than any trees found by DupTree (Table 1).

data	our heuristic		DupTree	
set	dup. range	time	dup. range	time
1	248757	31s	248757	842s
2	1843 - 1880	7s	1847 - 1879	1568s
3	813 - 848	4s	825 - 839	489s

Table 1: Timings (in seconds) and final gene duplications (minimum - maximum of 100 individual searches) for the data sets (1) green plants, (2) gymnosperms, and (3) Saxifragales.

Simulated Data Sets – We also performed analyses on data sets that we generated by simulation. We made both small data sets (400, 800, 1200, 1600, and 2000 species; Figure 3) and large data sets (10,000, 20,000, 40,000, 60,000, 80,000, and 100,000 species; Figure 4). For both the small and large simulated data sets, we first generated a species tree based on the Yule pure birth process using r8s [24]. For the small data sets, we then generated 500 gene trees with a minimum of 25 taxa and a maximum of half the number of taxa in the species tree, and for the large data sets we generated 1000 gene trees with a minimum of 5% of the total species and a maximum of 50% of the total species. To make the gene trees, we first randomly selected internal nodes in the species tree that had a sufficient number of descending leaves. We then randomly chose a number between 0 and 100%, and, if this percentage of leaves descending from our chosen node exceeded our minimum size requirement, we randomly selected this percentage of the descending leaves. We then made a subtree, our gene tree, which included only the randomly selected nodes. This process was repeated until we had the required number of gene trees. We finally introduced conflict to represent the conflict we observe in gene trees. The conflict consisted of SPR swaps on 25% of the nodes, moving them a maximum of 3 nodes from their original placement.

We created 100 starting species trees with a simple leaf adding heuristic and proceeded with the best one, and then averaged over multiple heuristic searches. On the small data sets we ran both our heuristic and DupTree (Figure 3). The 2000 taxon data sets took on average of 10.5 hours with DupTree, and we found that larger data sets become computationally unwieldy with DupTree. Thus, we only ran the larger data sets with our new heuristic, which was capable of inferring a species tree 50 times bigger than the small data set with 100,000 taxa in 36.3 hours on average.

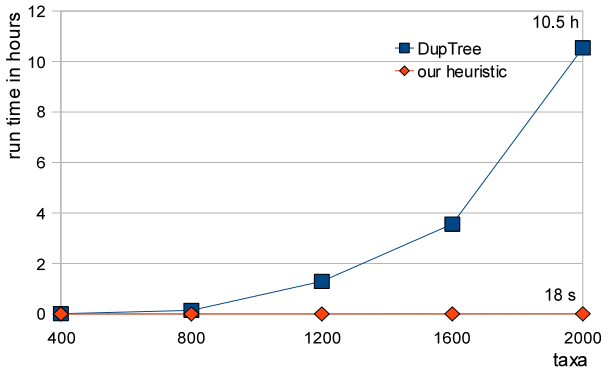

Figure 3: Performance comparison of heuristic searches on simulated data sets. The average run time of our heuristic is with 18 seconds at its worst where DupTree quickly reaches its limits.

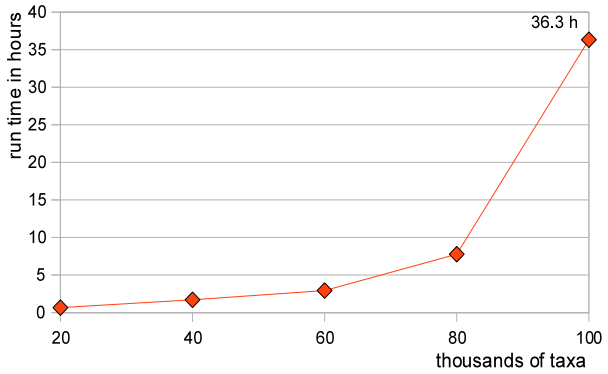

Figure 4: Average run time of our heuristic on large simulated data sets.

6 Discussion

In this paper, we described two new approaches that vastly improve the speed phylogenetic estimation based on GTP with little or no apparent loss in accuracy. In analyses of numerous empirical and simulated data sets, our new implementation of GTP is much faster than the GTP implementation in DupTree [20], which itself represented a tremendous increase in speed over the original naïve rSPR heuristics implemented in GeneTree [25]. With our new speedup it is now possible on a single desktop processor to use GTP to estimate species trees with 100,000 taxa, which, to our knowledge, is larger than any published phylogenetic trees computed from scratch.

While faster heuristics are desirable, ideally the increase in speed will not result in finding less optimal solutions than slower heuristics. Reassuringly, in our analyses of numerous empirical and simulated data sets, the best reconciliation scores using the new, faster heuristics are very similar to, and sometimes better than, scores from the previous local rSPR heuristic (Table 1, Figure 3). However, the rSPR searches may get trapped in local optima. In this case, our new heuristic allows many more runs than previous heuristics, and this should lead to better results than relying on a single or few runs. For example, in our 2000 taxon simulated data set, we could run our new heuristic 2100 times while DupTree is performs a single rSPR search. Still, we are currently working on new approaches, such as a ratchet search (see [26]), to ameliorate the problem of local optima.

Our new rSPR heuristic is currently limited to the duplication cost model of gene tree species tree reconciliation. With incomplete gene sampling, it is difficult to distinguish a gene loss from missing data. Therefore, unless there is complete genomic sequence data for all taxa, which is very unlikely across huge species sets, it may be appropriate to count only gene duplications. Still, we note that any sequential rSPR operation, like those used to infer trees under the duplication loss and deep coalescence cost models [15], can potentially be sped up with our initial algorithm, and we are working to generalize our heuristic speedup to other methods of gene tree reconciliation.

The new rSPR heuristic suggests that gene tree reconciliation may be an effective method to infer the tree of life. A parallel computing approach was effective for increasing the speed and scale of previous GTP heuristics [27], and such an approach may enable our current approach to scale to the size of the tree of life. The GTP approach may be especially amenable to computing the tree of life. GTP uses gene trees as input, as opposed to large alignments, which greatly reduces the memory requirements and makes it practical to incorporate thousands of genes into an analysis. Furthermore, GTP based on duplications or duplications and losses do not require orthologous genes, and unlike ML or MP, GTP can easily incorporate data large gene families in phylogenetic inference. With greater taxon sampling, it is more likely that gene trees will be incongruent due to events such as duplication, loss, deep coalescence, or horizontal transfer, and thus, GTP may be especially appropriate for inferring large-scale phylogeny from multiple genes from multi-gene data sets. Still, we note that GTP is certainly not a replacement for more conventional phylogenetic methods like ML or MP. In fact, GTP relies on such methods to build the input gene trees, and the computational burden required to build the gene tree data sets likely far exceeds the computational burden of inferring the species tree using GTP. Thus, while our new heuristic approaches to GTP make it possible to estimate large-scale phylogenies from gene tree, estimating the

tree of life will also require further speedup in other phylogenetic methods.

References

[1] T. J. Davies, S. A. Fritz, R. Grenyer, C. D. L. Orme, J. Bielby, O. R. P. Bininda-Emonds, M. Cardillo, K. E. Jones, J. L. Gittleman, G. M. Mace, A. Purvis, Phylogenetic trees and the future of mammalian biodiversity, Proc. Natl. Acad. Sci. USA 105 (2008) 11556–11563.

[2] B. C. Emerson, R. G. Gillespie, Phylogenetic analysis of community assembly and structure over space and time, Trends Evol. Ecol. 23 (2008) 619–630.

[3] P. H. Harvey, M. D. Pagel, The comparative method in evolutionary biology, Oxford University Press, New York.

[4] R. E. Ricklefs, Estimating diversification rates from phylogenetic information, Trends Evol. Ecol. 22 (2007) 601–610.

[5] D. A. Bader, U. Roshan, A. Stamatakis, Computational grand challenges in assembling the tree of life: problems & solutions, Advances in Computing in Computational Biology and Bioinformatics 68 (2006) 127–176.

[6] M. Ott, J. Zola, S. Aluru, A. Stamatakis, Large-scale maximum likelihood-based phylogenetic analysis on the IBM BlueGene/L, Proc. IEEE/ACM Supercomputing Conference.

[7] P. A. Goloboff, S. A. Catalano, J. M. Mirande, C. A. Szumik, J. S. Arias, M. Källersjö, J. S. Farris, Phylogenetic analysis of 73 060 taxa corroborates major eukaryotic groups, Cladisitics 25 (2009) 211–230.

[8] W. P. Maddison, Gene trees in species trees, Syst. Biol. 46 (1997) 523–536.

[9] R. Beiko, W. Doolittle, R. L. Charlebois, The impact of reticulate evolution on genome phylogeny, Syst. Biol. 57 (2008) 844–856.

[10] D. Penny, W. T. White, M. D. Hendy, M. J. Phillips, A bias in ML estimates of branch lengths in the presence of multiple signals, Mol. Biol. Evol. 25 (2008) 239–242.

[11] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, G. Matsuda, Fitting the gene lineage into its species lineage. a parsimony strategy illustrated by cladograms constructed from globin sequences, Syst. Zool. 28 (1979) 132–163.

[12] R. Guigó, I. Muchnik, T. F. Smith, Reconstruction of ancient molecular phylogeny, Molecular Phylogenetics and Evolution 6 (2) (1996) 189–213.

[13] B. Ma, M. Li, L. Zhang, From gene trees to species trees, SIAM J. Comput. 30 (2000) 729–752.

[14] M. S. Bansal, J. G. Burleigh, O. Eulenstein, A. Wehe, Heuristics for the gene-duplication problem: A $\theta(n)$ speed-up for the local search, RECOMB LNCS 4453 (2007) 238–252.

[15] M. S. Bansal, J. G. Burleigh, O. Eulenstein, Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models, BMC Bioinformatics.

[16] J. G. Burleigh, M. S. Bansal, O. Eulenstein, S. Hartmann, A. Wehe, T. J. Vision, Genome-scale phylogenetics: inferring the plant tree of life from 18,896 discordant gene trees., Syst. Biol.

[17] B. L. Allen, M. Steel, Subtree transfer operations and their induced metrics on evolutionary trees, Annals of Combintorics 5 (2001) 1–13.

[18] M. Bordewich, C. Semple, On the computational complexity of the rooted subtree prune and regraft distance, Annals of Combintorics 8 (2004) 409–423.

[19] M. A. Bender, M. Farach-Colton, The LCA problem revisited, in: Latin American Theoretical INformatics, 2000, pp. 88–94.

[20] A. Wehe, M. Bansal, J. G. Burleigh, O. Eulenstein, Duptree: A program for large-scale phylogenetic analyses using gene tree parsimony, Bioinformatics 24(13) (2008) 1540–1541.

[21] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic Local Alignment Search Tool, Journal of Mathematical Biology 215 (1990) 403–410.

[22] R. C. Edgar, Muscle: multiple sequence alignment with high accuracy and high throughput, Nucleic Acids Research 32 (2004) 1792–1797.

[23] A. Stamatakis, Raxml-vi-hpc: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models, Bioinformatics 22 (2006) 2688–2690.

[24] M. J. Sanderson, r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock, Bioinformatics 19 (2003) 301–302.

[25] R. Page, Genetree: comparing gene and species phylogenies using reconciled trees, Bioinformatics 14 (1998) 819–820.

[26] K. C. Nixon, The parsimony ratchet: a new method for rapid parsimony analysis, Cladistics 15 (1999) 407–414.

[27] A. Wehe, W. Chang, O. Eulenstein, S. Aluru, A scalable parallelization of the gene duplication problem, JPDC.