Molecular Phylogenetics and Evolution 124 (2018) 122–136

Contents lists available at ScienceDirect

Molecular Phylogenetics and Evolution

journal homepage: www.elsevier.com/locate/ympev

SVDquest: Improving SVDquartets species tree estimation using exact optimization within a constrained search space

Pranjal Vachaspati, Tandy Warnow*

Department of Computer Science, University of Illinois, Urbana, IL 61801, USA

A B S T R A C T

Species tree estimation from multi-locus datasets is complicated by processes such as incomplete lineage sorting (ILS) that result in different loci having different trees. Summary methods, which estimate species trees by combining gene trees, are popular but their accuracy is impaired by gene tree estimation error. Other approaches have been developed that only use the site patterns to estimate the species tree, and so are not impacted by gene tree estimation issues. In particular, PAUP* provides a method in which SVDquartets is used to compute a set $\mathcal{Q}$ of quartet trees (i.e., trees on four leaves), and then a heuristic search is used to combine the quartet trees into a species tree T, seeking to maximize the number of quartet trees in Q that agree with T. The PAUP* method based on SVDquartets (henceforth referred to as SVDquartets + PAUP*) is increasingly used in phylogenomic studies due to its ability to reconstruct species trees without needing to estimate accurate gene trees.

We present SVDquest*, a new method for constructing species trees using site patterns that is guaranteed to produce species trees that satisfy at least as many quartet trees as SVDquartets + PAUP*. We show that SVDquest* is competitive with ASTRAL and ASTRID (two leading summary methods) in terms of topological accuracy, and tends to be more accurate than ASTRAL and ASTRID under conditions with relatively high gene tree estimation error. SVDquest* is available in open source form at https://github.com/pranjalv123/SVDquest.

1. Introduction

Species tree estimation is a fundamental part of many biological analyses. Evolutionary processes such as incomplete lineage sorting (Maddison, 1997), gene duplication and loss (Ohno, 1970), and horizontal gene transfer (Woese, 2002) can result in heterogeneity across the genome, so that different parts of the genome have different trees. Much of the recent literature has focused on species tree estimation in the presence of incomplete lineage sorting (ILS), as - according to the multispecies coalescent model (MSC), which models gene tree heterogeneity due to ILS - all large-scale phylogenomic studies are expected to be affected by ILS to some extent. Furthermore, recent research has established that one of the most commonly used methods for species tree estimation, unpartitioned concatenation using maximum likelihood (CA-ML), can be statistically inconsistent under the MSC, and may converge to the wrong tree with probability converging to 1 as the number of loci increases for some model species trees (Roch and Steel, 2015)! Simulation studies evaluating CA-ML have also shown model conditions where it can produce highly supported incorrect trees in the presence of sufficiently high ILS, as well as other model conditions in which it performs well and may possibly be statistically consistent (e.g., Kubatko and Degnan, 2007; DeGiorgio and Degnan, 2010).

New approaches for constructing species trees that are statistically consistent under the MSC have been developed (see Mallo and Posada, 2016; Allman et al., 2017 for recent reviews) and used in a number of biological analyses (e.g., Song et al., 2012; Jarvis et al., 2014; Wickett et al., 2014; Mitchell et al., 2017; Hosner et al., 2016). Many of these approaches operate by computing gene trees on different loci and then combine these estimated gene trees into a species tree, and some of these "summary methods" (e.g., MP-EST (Liu et al., 2010), NJst (Liu and Yu, 2011), ASTRAL (Mirarab et al., 2014; Mirarab and Warnow, 2015; Zhang et al., 2017), and ASTRID (Vachaspati and Warnow, 2015)) can give highly accurate results in practice in circumstances where CA-ML performs poorly due to high levels of gene tree discordance (Liu et al., 2010; Mirarab et al., 2016; Vachaspati and Warnow, 2015). Furthermore, some summary methods are typically very fast and can analyze large datasets. As a result, summary methods have become standard approaches for estimating species trees when ILS is suspected.

However, the proofs of statistical consistency for summary methods depend on having accurate gene trees (Roch and Warnow, 2015), which is generally not expected on biological datasets. In addition, the proofs

* Corresponding author at: Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N Goodwin Ave, Urbana, IL 61801, USA.
E-mail address: warnow@illinois.edu (T. Warnow).

https://doi.org/10.1016/j.ympev.2018.03.006
Received 28 April 2017; Received in revised form 5 March 2018; Accepted 7 March 2018
Available online 09 March 2018
1055-7903/

depend on all sites within each locus evolving down a single tree (i.e., c-genes), and meeting that requirement can result in very short sequences for each locus (Springer and Gatesy, 2016), which increases gene tree estimation error. Furthermore, from an empirical standpoint, there is ample evidence that gene tree estimation error increases the error of species trees estimated using summary methods (Huang et al., 2010; Patel et al., 2013; DeGiorgio and Degnan, 2014; Gatesy and Springer, 2014; Bayzid and Warnow, 2013; Mirarab et al., 2016; Springer and Gatesy, 2016; Meiklejohn et al., 2016; Molloy and Warnow, 2017), and that CA-ML can be more accurate than even the most accurate summary methods when gene tree estimation error is sufficiently high, even in the anomaly zone (see Molloy and Warnow (2017) and references therein).

The impact of gene tree estimation error on species tree estimation has led to interest in methods that can estimate species trees without needing to compute gene trees, and that are statistically consistent under the MSC. One such approach is to co-estimate gene trees and species trees; *BEAST (Heled and Drummond, 2010) and BEST (Liu, 2008) are two such methods, but both are very computationally intensive (Bayzid and Warnow, 2013; Zimmermann et al., 2014; Knowles, 2009; McCormack et al., 2013; Leavitt et al., 2016). Another type of approach estimates the tree directly from the observed site pattern frequencies using properties of the MSC, and does not also try to estimate gene trees; examples of such methods include SuperMatrix Rooted Triple (SMRT) (DeGiorgio and Degnan, 2010), SNAPP (Bryant et al., 2012), SVDquartets (Chifman and Kubatko, 2014), and METAL (Dasarathy et al., 2015; Mossel and Roch, 2017). PoMo (De Maio et al., 2015) and its improved version revPoMo (Schrempf et al., 2016) can also be considered in this category, although these methods are not established to be statistically consistent under the MSC. These "site-based" methods are considered particularly suitable for datasets generated using phylogenomic protocols such as RADseq that produce loci with very few variable sites, which makes highly accurate gene tree estimation unlikely (de Oca et al., 2017).

The most popular of these site-based methods is available in PAUP* (Swofford, 2003) and operates as follows. Given a multi-locus dataset, the loci are concatenated into a single long alignment. Then, for each set of four species, a quartet tree for that set is computed using SVDquartets. Finally, a species tree is sought that agrees with as many of these quartet trees as possible. The number of quartet trees that the species trees satisfies is called its MQSST score, where MQSST refers to the *Maximum Quartet Support Species Tree*, and the problem of finding the tree with the highest MQSST score is the MQSST problem. Because the MQSST problem is NP-hard (Jiang et al., 2001), PAUP* uses a heuristic search to seek a good solution to MQSST. This method, which we refer to as SVDquartets + PAUP*, is increasingly popular in phylogenomics studies (Leaché et al., 2015; Campillo, 2016; Manthey et al., 2016; White et al., 2016; Leavitt et al., 2016; Crowl et al., 2017; He et al., 2017; Hosner et al., 2016; Manthey et al., 2017; Moyle et al., 2016; Boucher et al., 2016; Hime et al., 2016; Mitchell et al., 2017; Alexander et al., 2017; de Oca et al., 2017; Anderson et al., 2017; White et al., 2017).

We present SVDquest, a new site-based method for estimating species trees in the presence of ILS. SVDquest has the same basic approach as SVDquartets + PAUP* in that it uses SVDquartets to estimate quartet trees, and then combines these quartet trees into a species tree; the difference between SVDquest and SVDquartets + PAUP* is the technique each uses to combine the quartet trees. Instead of employing a heuristic search strategy, SVDquest uses dynamic programming (an algorithm design technique) to find a provably optimal solution to the MQSST problem within a constrained search space. The constraints are defined by a set of *allowed bipartitions on the species set*, and we use the dynamic programming algorithm from Bryant and Steel (2001) to find a species tree that maximizes the quartet support score within that constrained search space. If the search space is not constrained, then SVDquest finds a globally optimal solution to MQSST but will run in time that is exponential in the number of species, and so be too computationally intensive to use on datasets with more than about 15–20 species. However, we show that we can constrain the search space so that the algorithm runs in polynomial time and finds very good solutions to its optimization problem. Furthermore, by selecting the bipartitions appropriately, the new method, which we refer to as SVDquest*, is *guaranteed* to satisfy at least as many quartet trees computed by SVDquartets as SVDquartets + PAUP*.

We present results from an extensive performance study using both simulated and biological datasets. We find that SVDquest* finds better MQSST scores than SVDquartets + PAUP* under most conditions, particularly under higher levels of ILS and gene tree estimation error. We compare SVDquest* to a set of leading species tree estimation methods. We include two summary methods ASTRAL (Mirarab et al., 2014; Mirarab and Warnow, 2015) and ASTRID (Vachaspati and Warnow, 2015), because these two methods have been shown to have high accuracy under a wide range of model conditions and are both statistically consistent under the MSC (again, under the assumption that each is given true gene trees). We also include CA-ML (using RAxML), since (as noted earlier) trees computed using CA-ML are often at least as accurate as trees computed using summary methods. We do not include any of the co-estimation methods, as they are too computationally intensive to use on the datasets we explore.

The relative performance between these methods depends on the model condition (and in particular on the amount of gene tree estimation error), but SVDquest* has dramatic improvements over the summary methods under conditions with high average gene tree estimation error and many genes. CA-ML has the best accuracy of all methods under conditions with low to moderate ILS, but the coalescent-based methods outperform CA-ML when ILS is sufficiently high.

Finally, we also show that returning the strict consensus of the optimal trees computed by SVDquest* provides further improvements in topological accuracy. Thus, SVDquest* is a new method with improved accuracy compared to existing coalescent-based species tree methods under a range of realistic model conditions.

2. Materials and methods

2.1. SVDquest

The input to SVDquest is a set of sequence alignments (one alignment for each locus). Phase 1 of SVDquest uses the SVDquartets implementation in PAUP* to compute a set of quartet trees and Phase 2 combines these quartet trees into a species tree on the full set of species. Thus, SVDquest is identical to SVDquartets + PAUP* in Phase 1, but differs from SVDquartets + PAUP* in Phase 2. Specifically, SVDquest uses dynamic programming to find an optimal solution to the MQSST problem within a constrained search space, similar to how ASTRAL and FastRFS (Vachaspati and Warnow, 2017b) (a method for the Robinson-Foulds Supertree problem) find optimal trees for their optimization problems within constrained search spaces. Furthermore, since there can be more than one optimal tree for the MQSST problem, we also consider an optional Phase 3, in which a consensus tree is computed on the optimal trees found in Phase 2 using SIESTA (Vachaspati and Warnow, 2017a), a method that can be used with dynamic programming methods (such as SVDquest) that solve constrained optimization problems. As shown in Vachaspati and Warnow (2017a), the use of SIESTA with ASTRAL and FastRFS to compute the strict consensus tree typically results in an improvement in overall topological accuracy, suggesting that SIESTA might also improve SVDquest*.

2.1.1. Phase 1: computing the set Q of quartet trees

Phase 1 (the quartet tree estimation phase) computes an unrooted binary tree for every set of four species. For each set of four species, we use SVDquartets as implemented in PAUP* to select the best of the three possible quartet trees, and we refer to the set of all quartet trees

computed in this way by Q. In some cases, SVDquartets will not return a tree on a set of four species because the scores are too close (e.g., this can happen when there are too few variable sites).

2.1.2. Phase 2: computing an optimal species tree from using Q

Phase 2 (the species tree estimation phase) uses the quartet trees computed in Phase 1, and attempts to find an optimal solution to the MQSST problem. Since MQSST is NP-hard, algorithms for finding the globally optimal solution are not scalable. Hence, SVDquest typically operates in a mode where instead of searching for a globally optimal tree, it finds an optimal tree within a constrained search space.

SVDquest has three modes: *unconstrained, constrained-basic*, and *constrained-enhanced*. The *unconstrained* mode does not constrain the search space at all and hence is the most computationally intensive; it can only be used when the number of species is small enough (up to 15–20 taxa). In the two constrained modes, the search space is defined by a set X of bipartitions on the species set, and the constraint is that the output species tree *must* draw its bipartitions from X. Therefore, if X is all possible bipartitions on the species set then there is no constraint on the set of species trees that can be returned; otherwise, there is a reduction in the set of species trees that can be considered during the search for the best tree.

To compute the basic set X of allowed bipartitions, SVDquest uses the following protocol. First, it computes maximum likelihood trees on every gene; then, it runs a subroutine in ASTRAL-II (Mirarab and Warnow, 2015) to compute a set X of bipartitions that is guaranteed to include all the bipartitions from the input gene trees. The enhanced set of allowed bipartitions is computed by adding bipartitions to X. For example, we can add the bipartitions in the SVDquartets + PAUP* tree to X; we refer to this *constrained-enhanced* variant as SVDquest*. Since SVDquest exactly solves the MQSST optimization problem within the constrained search space, the SVDquest* tree is *guaranteed* to have a MQSST score that is at least as good as the SVDquartets + PAUP* tree's MQSST score.

2.1.3. Phase 3: Returning the strict consensus of the set of optimal trees computed in Phase 2

As noted, there can be more than one species tree that has an optimal MQSST score within the constrained space. Hence we provide an optional Phase 3 in which we use SIESTA to compute the strict consensus of all the trees in that set.

2.1.4. Comparison between SVDquartets + PAUP* and SVDquest variants

SVDquest and SVDquartets + PAUP* are techniques that compute a set Q of quartet trees using SVDquartets and then attempt to find a species tree that satisfies the maximum number of quartet trees in Q. The key difference between these methods is how each it solves the MQSST problem. SVDquest and its variants use a polynomial time dynamic programming algorithm from Bryant and Steel (2001) to provably solve MQSST within a constrained search space; in contrast, PAUP* uses other techniques to attempt to solve MQSST that do not provide guarantees of optimality within any constrained search space, but have the benefit of not being explicitly constrained to a subset of the search space.

2.2. Datasets

We explored performance on 10-, 15-, and 50-taxon simulated datasets. We also analyzed a mammalian biological dataset with 37 species that was first studied in Song et al. (2012), and later used to compare coalescent-based methods (Mirarab et al., 2014; Springer and Gatesy, 2016; Bayzid and Warnow, 2013). The mammalian dataset originally included 447 loci, but 21 of these had mislabeled sequences and two were clear outliers (Mirarab et al., 2014), so we excluded them from our analysis. This left 424 loci and a total of 1,338,678 sites in the concatenated alignment. We obtained the alignments and RAxML gene trees from Song et al. (2012), and we obtained the CA-ML tree on the 424 loci from Mirarab et al. (2014). The average bootstrap support on the gene trees was 71%. The main questions are the positions of two groups: *Chiroptera* and *Scandentia*.

Table 1
Summary of simulated datasets. GTEE is gene tree estimation error (i.e., the average normalized Robinson-Foulds distance between the estimated and model gene trees). AD% measures the average normalized Robinson-Foulds distance between the model gene trees and the model species tree, and is due only to ILS.

Number of taxa	50	15	10
Number of loci	50, 100, 500, 1000	50, 100, 1000	25, 50, 200
Locus length	25, 50, 100, 300	10, 100, 300	10, 50, 100
GTEE	15%-100%	15%-72%	40%-75%
AD% (ILS)	13%, 33%, 72%	82%	43%, 84%
Number of replicates	40–50	10	10
Strict molecular clock?	No	Yes	No

The simulated datasets were derived from prior publications, described individually below. Each dataset has model gene trees that evolve down model species trees under the multi-species coalescent, and indel-free sequence alignments evolved down those gene trees under standard site evolution models. Each gene is a proper c-gene (i.e., there is no recombination within any gene), and unless specified otherwise, the strict molecular clock assumption is not enforced. In some cases, we combined sequence data from different genes together (to simulate failure to detect recombination) by concatenating sequences simulated on gene trees with different topologies; this produces a set of "supergene" datasets.

Statistics for the simulated c-gene datasets are presented in Table 1. To characterize the level of ILS, we use the AD value, which is the average normalized Robinson-Foulds (RF) distance (Robinson and Foulds, 1981) between model gene trees and model species trees (i.e., the percentage of the non-trivial bipartitions in the true gene tree that do not appear in the true species tree). We also report gene tree estimation error (GTEE), which is the average normalized RF distance between model gene trees and estimated gene trees.

The 50-taxon datasets were simulated with SimPhy (Mallo et al., 2015) and were originally presented in Mirarab and Warnow (2015). These datasets have 50 taxa, 1000 loci, and 300–1500 sites per locus. The original versions of these datasets have 200 taxa, but 150 taxa have been randomly removed from each replicate to reduce the time and memory requirements of the analysis. This dataset contains three model conditions with three different ILS levels of 13%, 33%, and 72% AD. Loci were originally of variable length, but we reduced the sequence lengths for these experiments to 25, 50, 100, and 300 sites. These datasets have a speciation rate of 10^{-6}, resulting in speciation close to the tips of the model trees (i.e., recent divergence). Sequences evolved with a GTR + Gamma model and no molecular clock.

There were 26 model conditions (all with very high ILS) on which SVDquartets + PAUP* and SVDquest failed to return a tree. In these cases, PAUP* reported that there were "No informative quartets found", and examining the sequences showed that there were very few parsimony-informative sites. Hence, we report results only on those replicates for which all methods completed, a number that varies from 48 to 50 for each model condition. See Supplementary Materials, Section 2, for additional details.

The 15-taxon simulated datasets (from Bayzid et al. (2015)) have very high ILS (82% AD), and have 1000 loci with 1000 sites evolved with a strict molecular clock using a GTR + Gamma model (i.e., the gene trees are ultrametric). Model species trees all have the same "caterpillar" topology, and gene trees obey a strict molecular clock. We used 10, 100, and 1000 sites per locus, with an average of 65%, 53%, and 18% GTEE, respectively. Each model condition has 10 replicates, and all of them completed successfully with all methods.

The 10-taxon simulated datasets (also from Bayzid et al. (2015))

have two ILS levels (43% and 84% AD). These datasets have 200 loci with 10, 50 and 100 sites per locus, and GTEE levels between 40% and 75%. Species trees were randomly generated under a Yule process, gene trees are not ultrametric, and sequence data evolved under a GTR + Gamma model. Each model condition has 20 replicates; however, as with the 50-taxon datasets, some replicates had too few parsimony-informative sites, so that SVDquartets failed to compute any quartet trees. This occurred for 10 replicates of each model condition (combination of ILS level, number of genes, gene sequence length), and we report results for the other 10 replicates of each model condition.

2.3. Species tree methods for comparison

We compared SVDquest* to ASTRAL v4.10.2 (Mirarab and Warnow, 2015), ASTRID v1.4 (Vachaspati and Warnow, 2015), SVDquartets + PAUP* as implemented in PAUP* v4.0a151 (Swofford, 2003), and unpartitioned concatenated maximum likelihood (CA-ML) under a GTR-GAMMA model using RAxML v8.2.6 (Stamatakis, 2006). The same version of RAxML was used to estimate trees on gene sequence alignments and supergene sequence alignments under a GTR-GAMMA model. We ran the Windows version of PAUP* using WINE v1.6.2 due to its improved numerical routines compared to the Linux version. Exact commands for all methods are supplied in Appendix A.

2.4. Evaluation criteria

On the simulated datasets we use Dendropy (Sukumaran and Holder, 2010) to evaluate estimated trees for topological accuracy. All model species trees are binary (i.e., fully resolved), but some of the estimated species trees are not binary; hence, we report the average of the false positive and false negative rates; this is identical to normalized Robinson-Foulds (RF) error rates (Robinson and Foulds, 1981) when the estimated trees are binary.

On the mammalian biological dataset, we evaluated the estimated species trees using established clades, taking branch support into account. For branch support on trees computed using the summary methods and CA-ML, we used the local posterior probability branch support technique (Sayyari and Mirarab, 2016) in ASTRAL, which is based on the initial set of estimated gene trees and has been shown to produce better estimates of the probability of a branch being accurate than multi-locus bootstrapping (Sayyari and Mirarab, 2016). Branch support of species trees computed using site-based methods such as SVDquartets + PAUP* or CA-ML is commonly performed using non-parametric bootstrapping, but this approach is computationally intensive because it requires the calculation of species trees for all the bootstrap replicates. For this reason, we used a modified non-parametric bootstrap support technique described below (with 100 bootstrap replicates) to produce estimates of the branch support for the SVDquest* tree, and we compare the branch support we obtained using this modified non-parametric bootstrapping technique to the support we receive using the usual non-parametric bootstrapping technique. The modification to non-parametric bootstrapping that we use is very simple, and provides an approximation to the branch support that would be obtained using full non-parametric bootstrapping. We compute the constraint set X of bipartitions using the original dataset (i.e., not bootstrapped). Then, for each of the 100 bootstrap replicates, we run SVDquartets to compute the quartet trees, and we run SVDquest on the quartet trees computed by SVDquartets, using the constraint set X. In every other respect, the estimation of branch support we use follows the same protocol as with the usual non-parametric boostrapping procedure. Note that this approach has the benefit that for every bootstrap replicate the quartet trees are based correctly on SVDquartets, and only differs from full non-parametric bootstrapping in how the search space is constrained. Hence, this branch support technique does not affect the MQSST score of any returned tree, and only constrains which trees are considered permitted solutions.

2.5. Experiments

2.5.1. Experiment 1: Comparing SVDquest* and SVDquartets + PAUP* on simulated c-genes with respect to MQSST scores

The goal of this experiment is to determine whether SVDquest* finds better MQSST scores than SVDquartets + PAUP*. We tested both methods on simulated and biological datasets and reported MQSST criterion scores.

2.5.2. Experiment 2: Comparing coalescent-based species tree estimation methods on simulated c-genes with respect to tree topology

In the second experiment, we evaluated SVDquest*, SVDquartets + PAUP*, ASTRAL, and ASTRID, with respect to tree topology accuracy on a wide range of simulated datasets where all genes are c-genes (i.e., for each gene, all the sites evolve down a common tree topology).

2.5.3. Experiment 3: Comparison of coalescent-based species tree methods on multi-locus supergene datasets

In this experiment, we explored SVDquest*, SVDquartets + PAUP*, ASTRAL, and ASTRID on multi-locus datasets where the c-genes are randomly combined into supergenes (with the same number of c-genes), so that the assumption that all the sites in a given locus evolve down the same tree is violated. This experiment is motivated by the real-world challenge of failing to detect recombination events within gene sequence alignments. We estimated ML trees on these supergene alignments, and then used these "supergene trees" as the input for ASTRAL and ASTRID. Since SVDquartets computes quartet trees using all the sites in the concatenated alignment, this does not impact SVDquartets; it also does not change MQSST scores for any estimated species tree, as these are based on the quartet trees computed using SVDquartets. Hence, the use of supergenes does not impact SVDquartets + PAUP*. However, the use of supergenes instead of genes impacts summary methods, since the supergene trees will not be equal to the gene trees. It also affects SVDquest and SVDquest*, since it can change the constraint space that it computes using ASTRAL.

2.5.4. Experiment 4: Comparison of coalescent-based methods to CA-ML on simulated datasets

In this experiment, we compared CA-ML to SVDquest*, ASTRAL, and ASTRID on all simulated datasets with respect to the normalized Robinson-Foulds topological error rates.

2.5.5. Experiment 5: Comparison of coalescent-based methods on a mammalian biological dataset

We compare the SVDquest* tree on the mammalian biological dataset to trees computed using SVDquartets + PAUP*, ASTRAL, and ASTRID (all three trees computed by us), as well as to a concatenation analysis (obtained from Mirarab et al. (2014)), with respect to branch support for established and proposed clades.

2.5.6. Experiment 6: Running time

We explore running time for SVDquartets + PAUP*, SVDquest, SVDquest*, and ASTRAL on the 37-taxon mammalian biological dataset.

3. Results

3.1. Results for Experiment 1

Experiment 1 compares SVDquest, SVDquest* and SVDquartets + PAUP* with respect to the MQSST scores they find. Although SVDquest does not always find better scores than SVDquartets + PAUP*, by design SVDquest* is guaranteed to find scores that are at least as large as those found by SVDquartets + PAUP*. In this section, we report the number of cases where SVDquest* finds a better

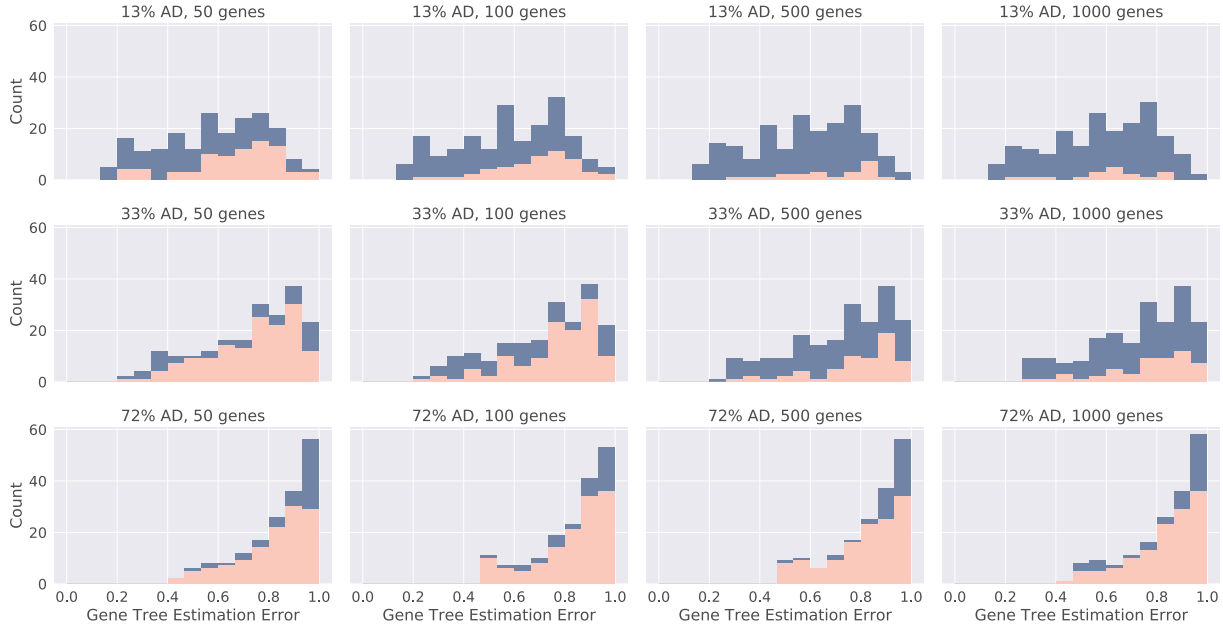

Fig. 1. Results for Experiment 1 on 50-taxon data, showing how often SVDquest* finds better MQSST scores than SVDquartets + PAUP*. Pink sections of bars represent replicates where SVDquest* finds a better scoring tree; blue sections represent replicates where both methods find the same scoring tree. It is impossible for SVDquartets + PAUP* to find a better scoring tree than SVDquest*. Total heights of bars represent distribution of gene tree estimation error (maximum possible value is 1.0) in datasets. Each subfigure shows results for 200 replicates, with the exception of the AD = 72% datasets, which have 195–200 replicates each.

score than SVDquartets + PAUP* and the number of cases where they have the same score; we also show the distribution of GTEE on the various datasets.

On the 50-taxon data, shown in Fig. 1, a few basic trends are clear. SVDquest* has a much greater advantage over SVDquartets + PAUP* at higher ILS levels, almost always finding better scores on the highest ILS model condition. SVDquest* also has a larger advantage when there are 50 or 100 genes, as opposed to 500 or 1000 genes. Generally, the advantage of SVDquest*' over SVDquartets + PAUP* improves as gene tree estimation error (GTEE) increases, until the very highest GTEE rates where the advantage starts to fall. See also Supplementary Materials Figs. S1–S3 for histograms of differences in MQSST scores between SVDquartets + PAUP* and SVDquest* on these datasets.

Results on the 15-taxon data (AD = 82%), shown in Fig. 2, also show that SVDquest* has a bigger advantage when there are fewer genes in the dataset. When there are 1000 genes, SVDquest* and SVDquartets + PAUP* almost always find trees with the same score, but SVDquest* frequently finds better trees when there are 50 or 100 genes. The impact of GTEE on the advantage with respect to MQSST score is less obvious on 50- and 100-gene datasets, but this may be because the range of GTEE is less than in the 50-taxon data.

The results on the 10-taxon data, shown in Fig. 3, once again show that increasing the ILS level or decreasing the number of genes increases the frequency with which SVDquest finds a tree with a better MQSST score than SVDquartets + PAUP*. Like the 15-taxon datasets, which have similar levels of GTEE, the relationship between GTEE and the relative performance of the two methods on these datasets is less clear than on the 50-taxon datasets.

3.2. Results for Experiment 2

Experiment 2 evaluates SVDquest*-strict (i.e., the strict consensus of all optimal trees found by SVDquest*, computed using SIESTA) in terms of topological accuracy on simulated c-gene datasets, and also compares it to ASTRID, ASTRAL, and SVDquartets + PAUP. The comparison between SVDquest*-strict and SVDquartets + PAUP* on the 50-taxon datasets (Supplementary Materials Figs. S4–S7) shows that although SVDquest*-strict and SVDquartets + PAUP* have similar accuracy, SVDquest* has an advantage over SVDquartets + PAUP*.

A comparison between SVDquest*-strict, ASTRAL, and ASTRID on the 50-taxon datasets with 500 genes is shown in Fig. 4 (see Supplementary Materials Fig. S8 for other numbers of genes). These datasets do not evolve under a strict molecular clock and vary in ILS levels (reflected in AD percentages), GTEE, and number of genes. ASTRAL and ASTRID have similar accuracy levels under most conditions. At low levels of GTEE, all methods are fairly accurate. With high GTEE, SVDquest*-strict is much more accurate than ASTRAL and ASTRID. At high levels of ILS and low GTEE, ASTRAL and ASTRID are more accurate than SVDquest*-strict. Across all model conditions, the crossover point where SVDquest*-strict becomes more accurate is approximately 50% GTEE. SVDquest*-strict also has a bigger advantage over ASTRAL and ASTRID when ILS levels are lower and there are more genes. In the most extreme case with close to 100% GTEE, 13% AD, and 1000 genes (see Supplementary Materials, Fig. S8), ASTRAL has approximately 75% estimation error while SVDquest*-strict has only 10% estimation error.

Fig. 5 shows results on the 15-taxon datasets (AD = 82%), which

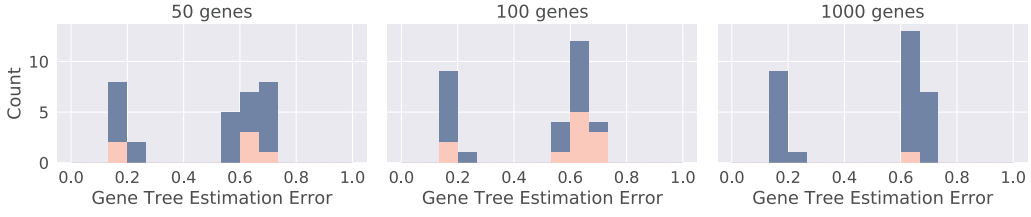

Fig. 2. Results for Experiment 1 on 15-taxon data with 82% AD (high ILS), showing how often SVDquest* finds a better scoring tree than SVDquartets + PAUP*. Pink sections of bars represent replicates where SVDquest* finds a better scoring tree; blue sections represent replicates where both methods find the same scoring tree. It is impossible for SVDquartets + PAUP* to find a better scoring tree than SVDquest*. Total heights of bars represent distribution of gene tree estimation error (maximum possible value is 1.0) in datasets. Each subfigure shows results for 30 replicates.

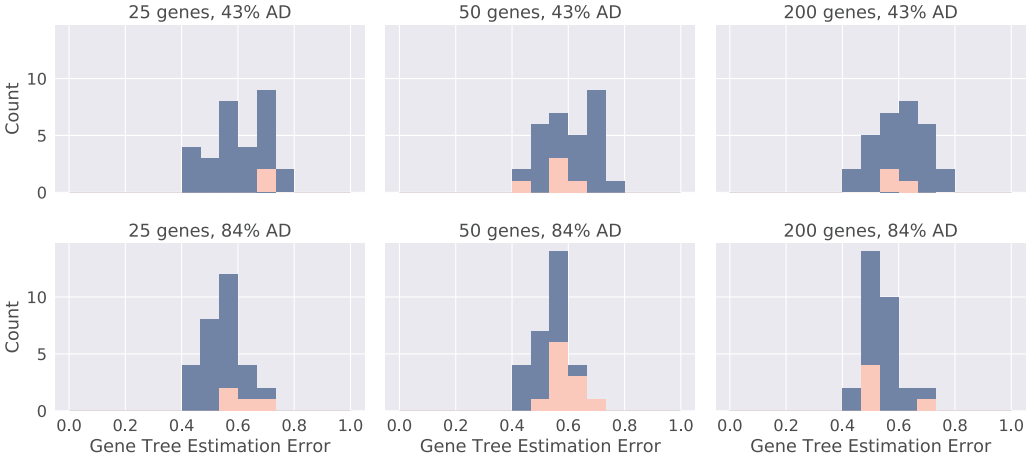

Fig. 3. Results for Experiment 1 on 10-taxon data, showing how often SVDquest* finds a better scoring tree than SVDquartets + PAUP*. Pink sections of bars represent replicates where SVDquest* finds a better scoring tree; blue sections represent replicates where both methods find the same scoring tree. It is impossible for SVDquartets + PAUP* to find a better scoring tree than SVDquest*. Total heights of bars represent distribution of gene tree estimation error (maximum possible value is 1.0) in the datasets. Each subfigure shows results for 30 replicates.

evolve under a strict molecular clock. ASTRAL is the most accurate method in all cases. The comparison between SVDquest*-strict and ASTRID shows that SVDquest*-strict has an advantage for the model conditions with largest number of genes (1000) and highest GTEE (40–60%), ASTRID has an advantage for the model conditions with fewest genes (50–100) and lowest GTEE (0–20%), and otherwise the two methods have similar species tree estimation error. However, this dataset has a relatively limited range of gene tree error - no replicate has greater than 60% average GTEE, which is the model condition where we would expect the best performance from SVDquest*-strict.

Results on the 10-taxon data (which do not evolve under a strict molecular clock) are shown in Fig. 6. All three methods have similar levels of accuracy under most conditions. However, ASTRAL frequently returns slightly more topologically accurate trees than the other two methods, and ASTRAL and ASTRID are somewhat more accurate than SVDquest*-strict when there is low GTEE. Like the 15-taxon data, this model condition has no replicates with greater than 60% average GTEE.

3.3. Results for Experiment 3

Experiment 3 compares SVDquest*-strict to ASTRAL, ASTRID, and SVDquartets + PAUP* on supergene datasets (i.e., when loci are not recombination-free). We report both MQSST scores and topological accuracy.

The comparison between SVDquartets + PAUP* and SVDquest*-strict shows that SVDquest*-strict typically matches or improves on SVDquartets + PAUP* with respect to topological accuracy (Supplementary Materials, Figs. S9–S11). In fact, the advantage of using SVDquest*-strict over SVDquartets + PAUP* is greater on the supergene datasets than on c-gene datasets. In what follows, we compare SVDquest*-strict to ASTRAL and ASTRID.

Results on the 50-taxon datasets are shown in Fig. 7. The recombination-free loci have only 25 sites; all other lengths indicate supergenes obtained by combining c-genes. On all the model conditions with 25-site loci, SVDquest*-strict has a substantial advantage over ASTRID and ASTRAL. On the lowest ILS model condition, SVDquest*-strict retains the same accuracy as the c-genes are combined into supergenes. However, as the number of c-genes per supergene increases, ASTRAL and ASTRID become more accurate, eventually equaling or improving over SVDquest*-strict. For example, on the 33% AD model condition, SVDquest*-strict has an advantage when the supergenes have at most two c-genes, but then only ties with ASTRAL and ASTRID when there are more c-genes per supergene. On the 72% AD model condition, SVDquest*-strict retains an advantage regardless of the number of c-genes per supergene, but the advantage decreases with the length of the supergene.

Results on the 15-taxon datasets are seen in Fig. 8. The c-genes have only 10 sites; all other lengths indicate supergenes obtained by binning together different c-genes. At the longest supergenes with 1000 sites (each composed of 100 recombination-free loci), ASTRAL and ASTRID find more accurate trees than SVDquest*-strict, but at lower levels of binning, SVDquest*-strict finds trees that are more accurate than ASTRID but less accurate than ASTRAL. ASTRAL finds slightly more accurate trees when the loci are recombination-free, while ASTRID improves substantially with increased binning, especially when there are 1000 loci. The impact of binning on SVDquest*-strict is minimal.

Relative performance on the 10-taxon data, shown in Fig. 9, is similar to the 15-taxon data. ASTRAL typically becomes less accurate at higher levels of binning, while SVDquest*-strict is relatively unaffected, and ASTRID sometimes improves. These trends are more evident at the 84% AD level; at the 43% AD level, there is relatively little change with increased binning.

3.4. Results for Experiment 4

Experiment 4 compares coalescent-based methods to unpartitioned concatenation using RAxML (i.e., CA-ML) on simulated datasets. On the 50-taxon data, seen in Fig. 10 for 500 gene datasets (see Supplementary

Fig. 4. Species tree topological error rates (maximum possible is 1.0) for 50-taxon simulated data, as a function of percent gene tree estimation error (maximum possible is 1.0); the first two figures show results for 200 replicates and the last figure shows results for 198 replicates. Error bars show standard error.

Fig. 5. Species tree topological error rates (maximum possible is 1.0) for 15-taxon simulated data (AD = 82%), as a function of gene tree estimation error (maximum possible is 1.0); each subfigure shows results on 30 replicates. Error bars show standard error.

Materials Fig. S8 for other numbers of genes), CA-ML and SVDquest*-strict tend to perform similarly, and better than ASTRID and ASTRAL when GTEE is greater than 60%. On the lower ILS (13% AD) condition, CA-ML is somewhat more accurate than SVDquest*-strict when there are fewer genes (Supplementary Materials Fig. S8), but this advantage is reduced for 500 or 1000 genes. At the highest ILS level, CA-ML is actually less accurate than SVDquest*-strict when there are few genes and low GTEE, but both of these methods are less accurate than ASTRAL and ASTRID.

On the 15-taxon data (AD = 82%), shown in Fig. 11, ASTRAL is always the most accurate method. ASTRID performs worse than CA-ML and SVDquest*-strict when there are 50 or 100 genes. With 1000 genes, all methods perform well, but ASTRAL and ASTRID slightly outperform CA-ML and SVDquest*-strict.

On the 10-taxon data, shown in Fig. 12, CA-ML is typically the best method on the 43% AD data. CA-ML slightly outperforms the other methods except when there are 50 genes and low GTEE, in which case ASTRAL and ASTRID perform slightly better. On the 84% AD data, SVDquest*-strict and CA-ML are the worst performing methods, and ASTRAL is typically the best method.

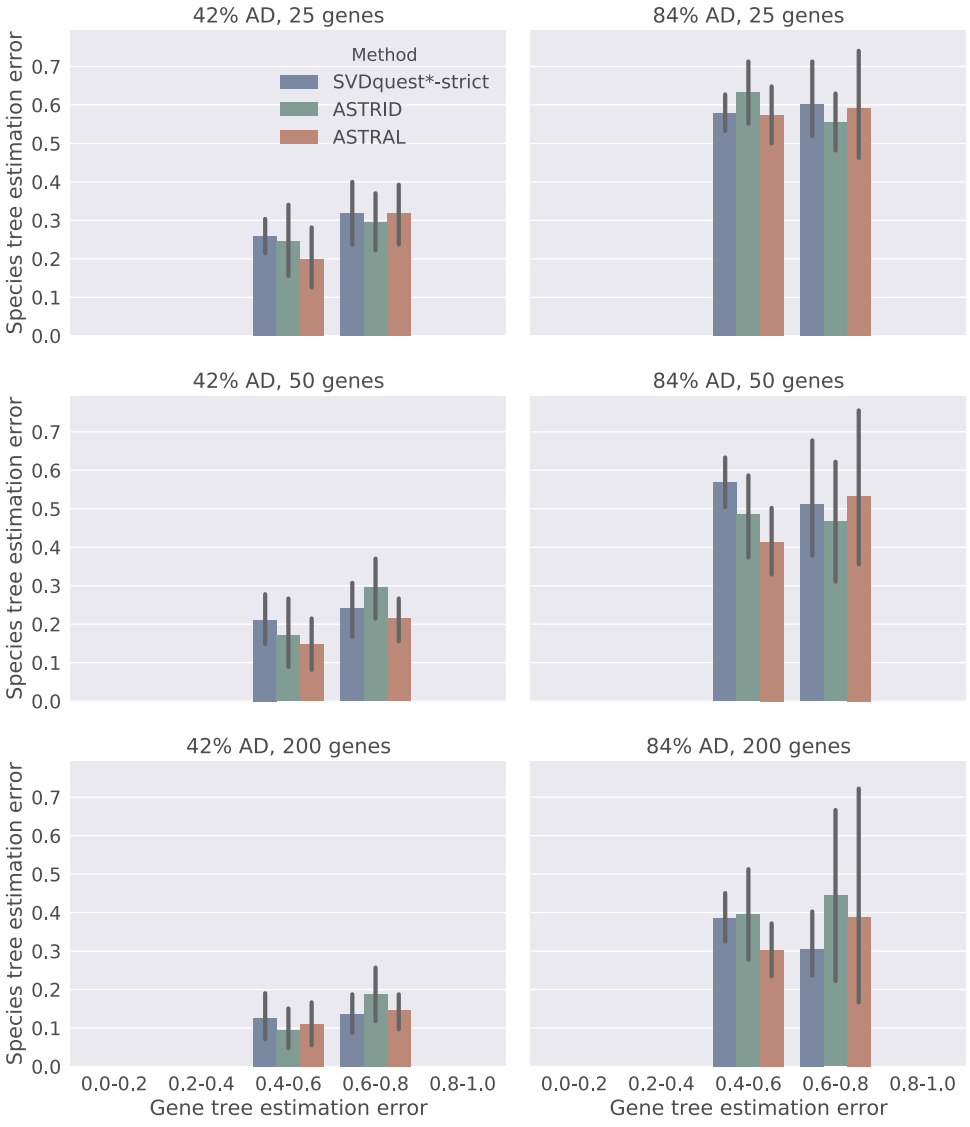

Fig. 6. Species tree topological error rates (maximum possible is 1.0) for 10-taxon simulated data, as a function of gene tree estimation error (maximum possible is 1.0); each subfigure shows results for 30 replicates. Error bars show standard error.

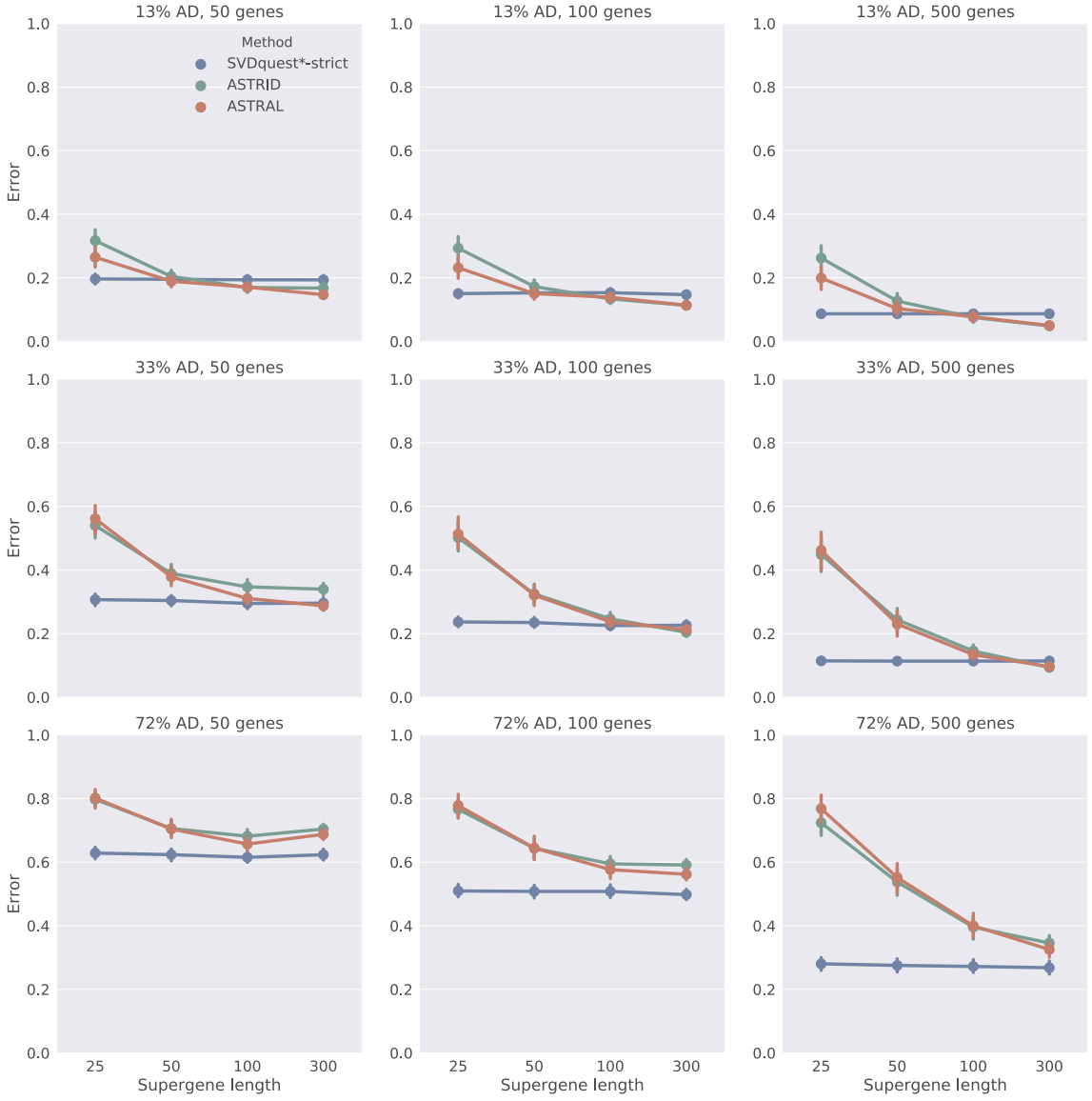

Fig. 7. Species tree error rates (maximum possible is 1.0) on 50-taxon simulated data using supergenes (concatenations of c-genes) that may not be recombination-free. Error bars represent standard error of the mean. The c-genes in this experiment are 25 sites long, and multiple loci were concatenated to form supergenes. Thus, the 25-site genes have sites coming from one c-gene, the 50-site genes have sites coming from two c-genes, and the 100-site genes have sites coming from four c-genes. Each data point in a particular subfigure represents an analysis on the same number of sites. Each data point corresponds to an average over 50 replicates, except for the AD = 72% 25-site 50-gene data point, which corresponds to 48 replicates, and 9 other AD = 72% data points, each of which corresponds to 49 replicates.

We calculated the rank correlation coefficient between the MQSST scores and topological errors for PAUP* and SVDquest* in order to determine whether a better MQSST score was correlated with a topologically more accurate tree. We found that there was a statistically significant correlation ($P < 0.05$) with a Spearman rank correlation coefficient of $\rho = 0.32$ for the 50-taxon datasets.

3.4.1. Results for Experiment 5

We compared SVDquest* to SVDquartets + PAUP*, CA-ML, ASTRAL, and ASTRID trees on the mammalian biological dataset. CA-

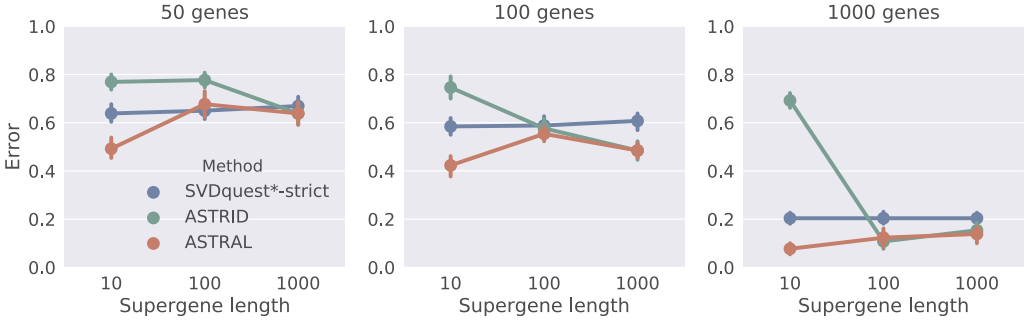

Fig. 8. Species tree error rates (maximum possible is 1.0) on 15-taxon simulated data (AD = 82%) for three different numbers of c-genes, then binned into supergenes (concatenations of c-genes); results shown are averaged over 10 replicates with error bars representing standard error of the mean. The c-genes in this experiment have 10 sites, so that longer loci are supergenes. Each data point in a particular subfigure represents an analysis on the same total number of sites.

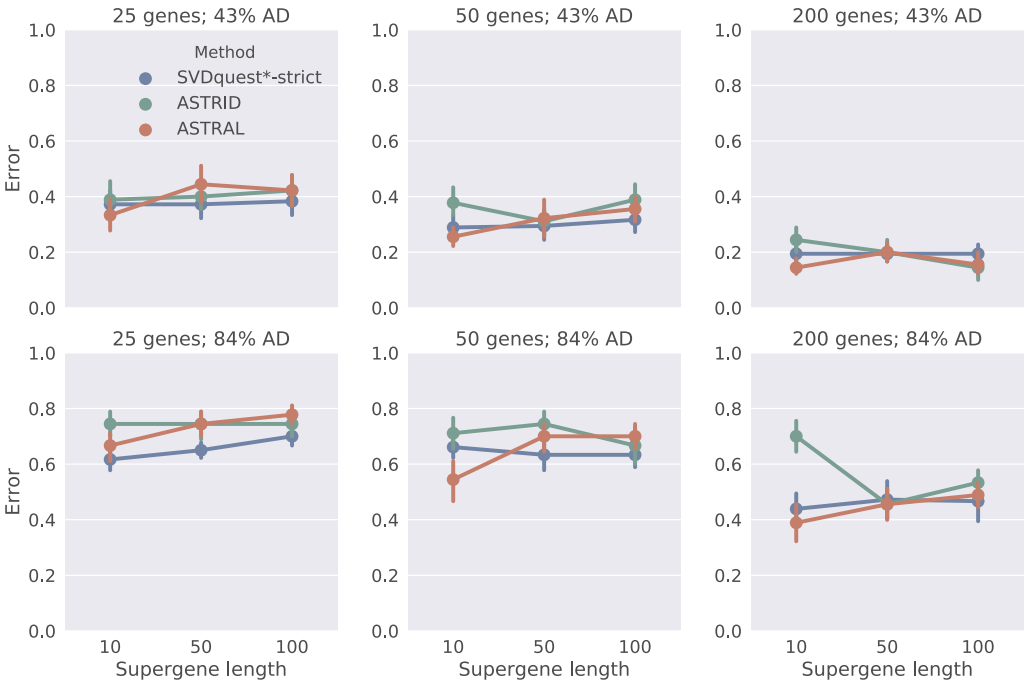

Fig. 9. Species tree error rates (maximum possible is 1.0) on 10-taxon simulated data using supergenes (concatenations of c-genes), averaged over 10 replicates; error bars represent standard error of the mean. The c-genes in this experiment have 10 sites, and longer sequences are supergenes. Each data point in a particular subfigure represents an analysis on the same number of sites.

ML, ASTRAL, and ASTRID all return the same tree, which recovers the major accepted mammalian clades and the relationships between them, but SVDquest* returns a single tree that is different from the tree found by the other methods. See Fig. 13 for the SVDquest* tree with bootstrap branch support, and Supplementary Materials Fig. S12 for the ASTRAL/ASTRID/CA-ML tree with ASTRAL branch support. The SVDquest* tree has very high bootstrap branch support on nearly all the edges (100% support on all but four edges and 99% support on one edge), and the ASTRAL/CA-ML/ASTRID tree has over 90% support using ASTRAL's local posterior probability for all of its branches.

The SVDquest* tree agrees with SVDquartets + PAUP* but differs from the tree found using CA-ML, ASTRID, and ASTRID in two ways: the placement of tree shrews (Scandentia) and the topology of the clade Scrotifera with respect to the placement of bats (Chiroptera). CA-ML, ASTRAL and ASTRID place Scandentia as sister to Glires, while SVDquest* places Scandentia as sister to Primates. Both these placements have 100% support (bootstrap support for the SVDquest* tree, local support for the ASTRAL/CA-ML/ASTRID tree).

Scrotifera consists of three major clades - Chiroptera (bats), Cetartiodactyla (even-toed ungulates and cetaceans), and Zooamata (odd-toed ungulates and carnivores). CA-ML, ASTRAL and ASTRID resolve this clade with Chiroptera as the outgroup with 90% local support. SVDquest* resolves this with Zooamata as the outgroup, but with very low support (only 23% bootstrap support using the modified bootstrapping technique).

The existing literature presents varied hypotheses for Scrotifera. The SVDquest* analysis presents support for a clade that consists of Cetartiodactyla and Chiroptera, which has been presented by Hou et al. (2009). The CA-ML, ASTRAL, and ASTRID analyses present support for Fereuungulata, which contains Cetartiodactyla and Zooamata. More recent analyses (Zhou et al., 2012) have found increased support for Fereuungulata over Cetartiodactyla + Chiroptera, but the phylogeny is not yet settled. However, the bootstrap support for Cetartiodactyla + Chiroptera in the SVDquest* tree is quite low (only 23%), and collapsing this edge makes the tree compatible with both of these two possibilities. SVDquest* establishes Zooamata with 69% support, which is only moderate. Collapsing edges in the SVDquest* tree with less than 75% bootstrap support resolves Cetartiodactyla, Chiroptera, Carnivora, and Perissodactyla as clades, but does not determine a relationship between them.

Finally, we also performed the standard non-parametric bootstrapping analysis on the SVDquest* tree, to evaluate the impact of using the modified bootstrapping technique for defining branch support. The results from the two techniques were nearly identical. All but three of the branches in the SVDquest* tree received exactly the same support using both techniques (91% for one branch, 100% for all the others). The differences in the support for the remaining three branches were very small. One branch that had 99% using the modified technique had 100% using the standard technique. The remaining two branches had support less than 75% using the modified bootstrapping technique, and their support changed by at most 5%: Ceteratiodactyla + Chiroptera received branch support of 23% using the modified technique and 19% using the standard technique, Zoomata received 69% support using the modified technique and 74%

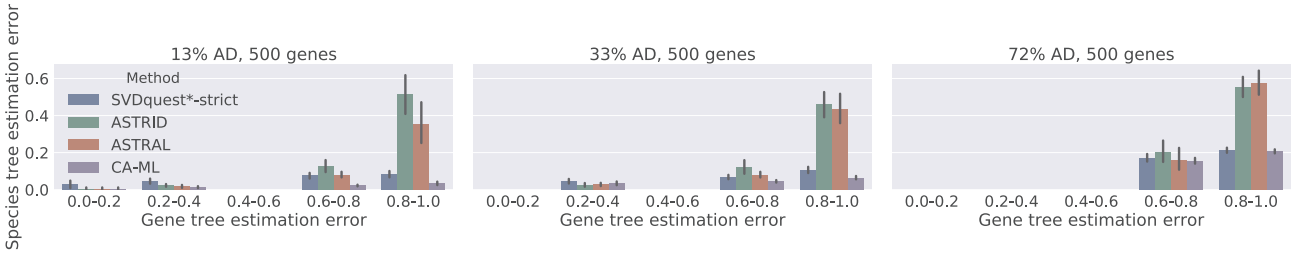

Fig. 10. Species tree topological error rates (maximum possible is 1.0) for 50-taxon simulated data, as a function of gene tree estimation error (maximum possible is 1.0). Each figure shows means and standard error; results in the first two subfigures are for 200 replicates and 198 replicates for the third subfigure.

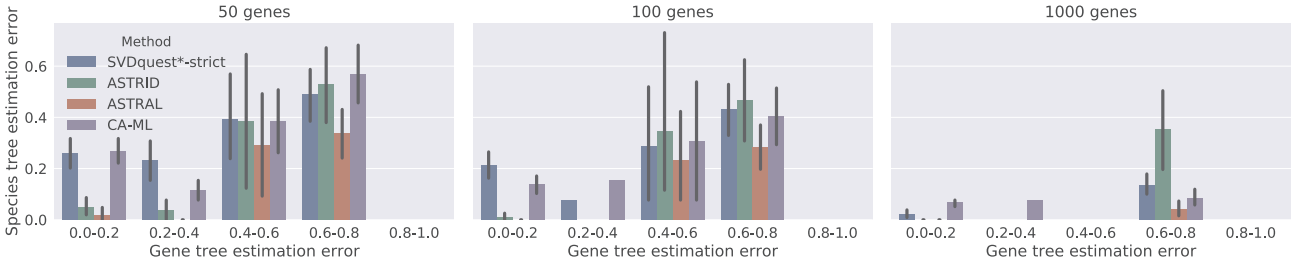

Fig. 11. Species tree topological error rates (maximum possible is 1.0) for 15-taxon simulated data (AD = 82%), as a function of gene tree estimation error (maximum possible is 1.0). Error bars show standard error over 10 replicates.

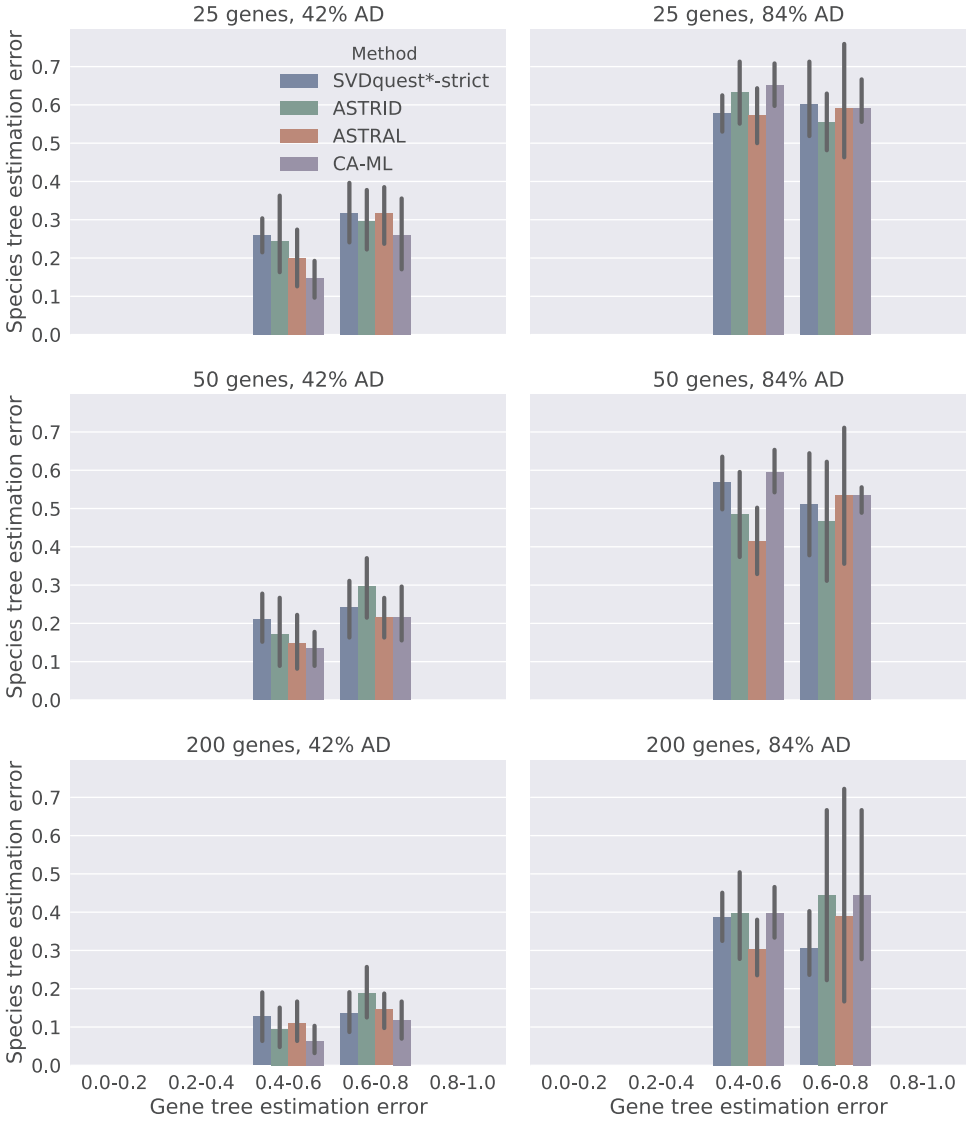

Fig. 12. Species tree topological error rates (maximum possible is 1.0) for 10-taxon simulated data, as a function of gene tree estimation error (maximum possible is 1.0). Error bars show standard error over 10 replicates.

using the standard technique. Thus, the modified bootstrapping technique to provide branch support produces branch support values that are very close to that produced using the standard bootstrapping technique, while being much faster.

3.4.2. Experiment 6: Running Time

We compare the (sequential) running times of ASTRAL, SVDquest*, and SVDquartets + PAUP* on the 37-species mammalian dataset with 424 loci and a total of 1,338,678 characters. SVDquartets + PAUP* completed in under 4 min. ASTRAL and ASTRID both finished in just under 3.5 h (less than one second difference), of which all but 4 s was spent computing ML gene trees. SVDquest* returned only one tree, and completed in under 3 h and 32 min, which was just seconds more than what was needed to compute the ASTRAL and SVDquartets + PAUP* trees. The detailed running time analysis for SVDquest* is as follows:

- Computing maximum likelihood gene trees: 210 min.
- Applying ASTRAL to the set of maximum likelihood gene trees, to obtain the constraint set of bipartitions: 6 s.
- Using PAUP* to compute SVDquartets quartet weights: 3 min.

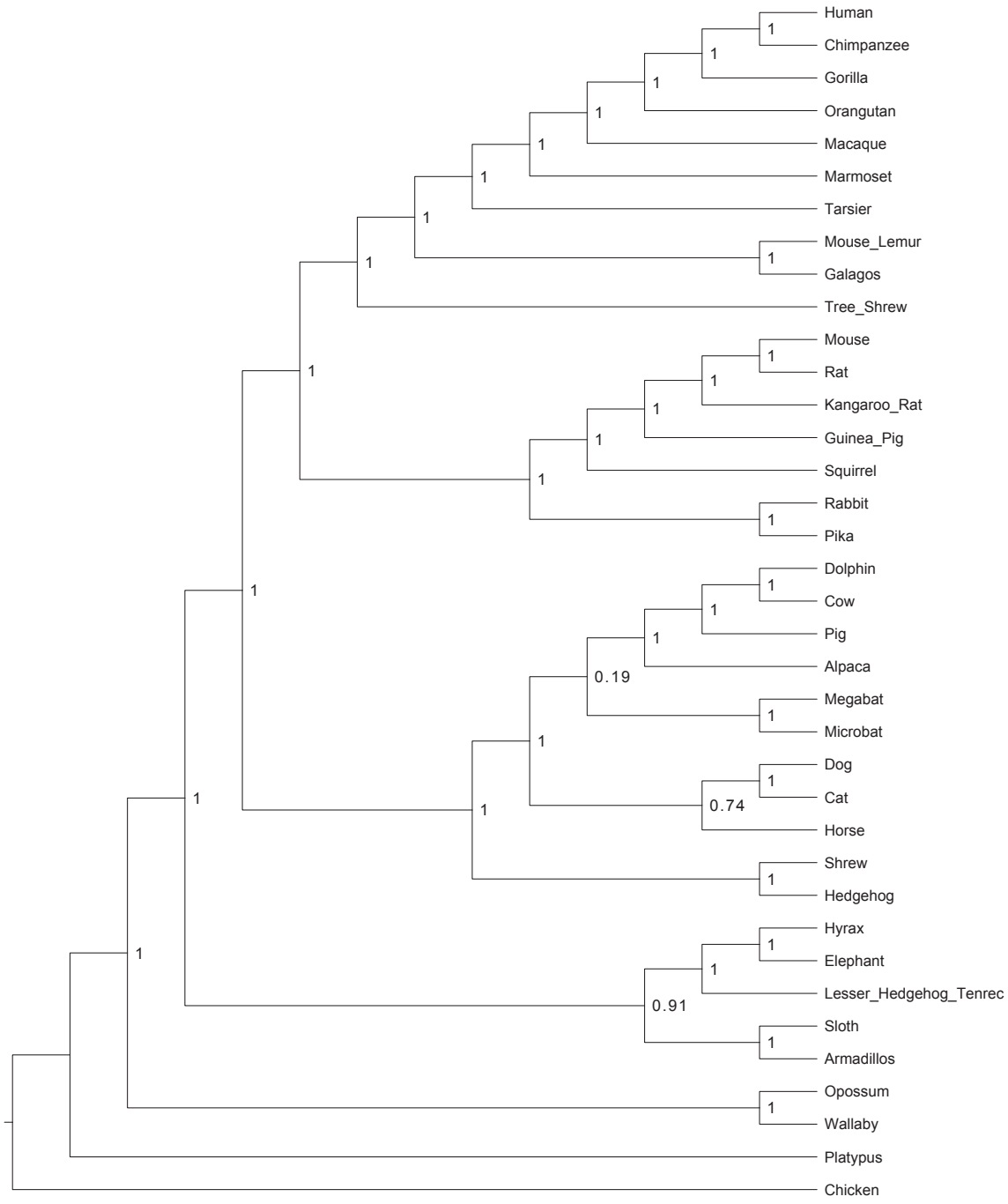

Fig. 13. Mammalian SVDquest* tree with branch support, computed using a modified non-parametric bootstrapping approach, from 100 bootstrap replicates.

- Applying PAUP* to the quartet trees to return the species tree: < 1 s.
- Running the dynamic programming within SVDquest* to find the optimal tree: 1 s.

Thus, the running time for SVDquest* (and for SVDquest*-strict) is essentially no different from that of running ASTRAL or ASTRID, and is dominated by the time used to compute ML gene trees. Also, SVDquartets + PAUP* is much faster than SVDquest* because SVDquartets + PAUP* does not need to compute gene trees.

4. Discussion

4.1. MQSST scores

By design, it is impossible for SVDquest* to produce a tree with a worse MQSST score than SVDquartets + PAUP*. Hence, the question is how much better SVDquest* is than SVDquartets + PAUP* at finding good MQSST scores, and how the different model conditions affect the frequency with which SVDquest* improves on SVDquartets + PAUP*.

Our data show that SVDquest* often finds better scores than SVDquartets + PAUP*, but the frequency of this improvement depends on the model conditions, and is clearly related to the difficulty of the MQSST problem instance. Obviously, if SVDquartets + PAUP* finds an optimal solution, there is no better solution for SVDquest* to find. More

generally, conditions that make it easy to find near-optimal MQSST using PAUP*'s heuristic search strategies will make it difficult for SVDquest* to do better than SVDquartets + PAUP*. ILS levels and number of genes both impact the relative performance, with an increasing advantage to SVDquest* over SVDquartets + PAUP* as ILS level increases or as the number of genes decreases. Both these trends are consistent with the hypothesis that easy conditions tend to reduce the advantage of SVDquest* over SVDquartets + PAUP* at finding good solutions to MQSST. The impact of GTEE is more complicated. Below about 30% GTEE, SVDquartets + PAUP* often finds a good MQSST score, so there is less room for SVDquest* to find an improvement. Above approximately 80% GTEE, SVDquartets + PAUP* might not find a good solution, but the gene trees have so much error that the constraint set computed by SVDquest* does not include parts of the solution space where better trees can be found. However, under other conditions (i.e., when GTEE is neither extremely low or extremely high), SVDquest* tends to produce better MQSST scores than SVDquartets + PAUP*.

These observations provide insights into the impact of GTEE on SVDquest*. It is well known that summary methods, such as ASTRAL and ASTRID, directly rely on estimated gene trees, and compute species trees based on summary statistics on the gene trees - quartet distributions or average internode distances. In contrast, gene tree estimation error only impacts how SVDquest* constrains the search space, and does not impact the criterion scores of any trees it can examine. Furthermore, the only real problem with using poorly estimated gene trees occurs when all the estimated gene trees are poor – because then the bipartitions of the true species tree may not end up in the constraint set. Adding bipartitions from poor gene trees to the constraint set expands the search space and hence increases the running time, but will never reduce the criterion score produced by SVDquest*. This suggests a general strategy of adding estimated species trees to the constraint set, even those that are not likely to be highly accurate; these expand the search space for SVDquest*, and are useful as long as they have a positive probability of containing a bipartition from a higher-scoring tree.

4.2. Species tree accuracy

A comparison between SVDquartets + PAUP* and SVDquest*-strict with respect to topological accuracy reveals that generally the differences are small, but that when the trees are different there is usually an improvement obtained by using SVDquest*-strict. The difference in accuracy is often small, but can be large (i.e., up to 10–15% in normalized RF). Hence, SVDquest*-strict provides an advantage (although slight) over SVDquartets + PAUP* in terms of species tree topology estimation.

The relative performance of ASTRAL and ASTRID in our study generally favored ASTRAL, in the sense that although the two methods were often very close in accuracy (and sometimes had identical accuracy), ASTRID was more impacted by GTEE than ASTRAL, and so was less accurate for the conditions with very short loci.

Both summary methods had very good accuracy – outperforming the other methods – when GTEE was sufficiently low and ILS was sufficiently high. However, CA-ML had the best accuracy under sufficiently low ILS levels, and even had the best accuracy under high ILS levels when GTEE was sufficiently high. SVDquest*-strict was less accurate than ASTRAL and ASTRID when GTEE was sufficiently low, but was as accurate as ASTRID and ASTRAL, and sometimes more accurate, when GTEE was very high.

Finally, although CA-ML typically dominated SVDquest*-strict, there were a few 10-taxon model conditions where SVDquest*-strict improved on the accuracy of CA-ML. Specifically, on the 10-taxon model conditions with high ILS (84% AD), high GTEE, and at least 50 genes, SVDquest*-strict was slightly more accurate than CA-ML.

4.3. Comparison to prior studies

Several other studies (surveyed in Molloy and Warnow (2017)) have compared coalescent-based methods and CA-ML under various simulated model conditions. These studies made the same general observations about the relative performance between the summary methods and CA-ML. Two prior studies (Chou et al., 2015; Molloy and Warnow, 2017) have compared SVDquartets + PAUP* to other methods, including ASTRAL, NJst, ASTRID, and CA-ML; although SVDquartets + PAUP* sometimes improved over the summary methods when GTEE was sufficiently high, it was only rarely more accurate than CA-ML. Although we report results for SVDquest*-strict (which directly improves on SVDquartets + PAUP* for optimizing MQSST trees), our findings are also consistent with these general trends.

It is not clear what factors influence the relative accuracy of SVDquartets-based methods and CA-ML, although these studies as a whole suggest that when ILS is low enough, then CA-ML should be more accurate than SVDquest*-strict and SVDquartets + PAUP*. The total number of sites also seems to influence the relative performance, so that under high enough ILS and a large enough number of sites, SVDquartets-based methods may have an advantage over CA-ML. However, in our studies, when there was an advantage, it was small.

Experiment 3 suggests that species trees based on supergene trees (instead of on trees computed on c-genes) can sometimes improve the accuracy of species trees computed using SVDquest*-strict, as well as ASTRAL and ASTRID. The improvement for ASTRAL and ASTRID is consistent with a similar study (but applied to different summary methods) where supergenes are also based on random collections of genes (Bayzid and Warnow, 2013); furthermore, Lanier and Knowles (2012) also observed that coalescent-based summary methods were generally robust to recombination within loci. The improvement is perhaps surprising, since current theoretical justifications for using summary methods require that the loci be recombination-free. Furthermore, Springer and Gatesy (2016) argue that recombination-free loci may be extremely short (as few as 12 base pairs), and point out that on this basis the theoretical justification of summary methods is flawed. This concern is justified. However, from an empirical standpoint the results in these experiments suggest that failure to break loci into recombination-free regions may not be a substantial problem - and may even lead to improvements in some (but not all) cases.

4.4. Running time considerations

Our study also examined running time, and showed that SVDquest*-strict was reasonably fast. However, SVDquest* needs ASTRAL and SVDquartets + PAUP* to compute the constraint set, and so is necessarily more computationally intensive than both SVDquartets + PAUP* and ASTRAL. By far the dominant part of the running time for SVDquest*-strict is the gene tree estimation part, but this can be parallelized (i.e., each gene tree can be calculated independently of the others). In particular, it is feasible to run SVDquest*-strict on any dataset on which the full set of quartet trees can be computed using SVDquartets, which is also easily parallelized.

4.5. Future work

This study suggests multiple directions for future research. We used the default setting within PAUP* and we computed quartet trees for every four leaves; these choices are supposed to maximize the accuracy of SVDquartets + PAUP*, but it is possible that some other way of combining quartet trees within PAUP* would result in topologically more accurate trees. Similarly, quartet tree amalgamation is a basic algorithmic problem, and SVDquartets + PAUP* could be improved through the use of new quartet amalgamation methods. In addition, a branch-swapping heuristic could be developed that begins with the SVDquest* tree and searches for better solutions to MQSST; thus,

SVDquartets + PAUP* can also be improved by incorporating SVDquest* as a starting tree. Furthermore, the basic strategy within SVDquest* of using other species tree methods to add bipartitions to the constraint set enables SVDquest to remain useful, even as PAUP* improves through the use of new quartet amalgamation heuristics.

Another interesting direction would be to modify the optimization problem that we solve. Thus, in the MQSST problem, there is exactly one tree on every four species, and each of these quartet trees has unit weight. A weighted version of MQSST would be very interesting to examine, where instead of taking the best topology for each four taxa, the three possible topologies are weighted based on their SVD scores or on the statistical support for the quartet tree (Gaither and Kubatko, 2016).

SVDquest*-strict could also be compared to PoMo (De Maio et al., 2015) and its improved version revPoMo (Schrempf et al., 2016), which estimate species trees from multi-locus datasets under a model of site evolution that allows each node in the tree to be polymorphic. While these methods have not been shown to be statistically consistent under the MSC, they have shown very good accuracy on simulated data, even when gene tree heterogeneity due to ILS is present, and so may provide excellent accuracy in practice.

The accuracy of SVDquartets for computing quartet trees on biological datasets is not well understood, and this also presents multiple opportunities for future research. For example, this study examined the use of SVDquest* with multi-locus datasets, and assumed that gene trees can be computed on each of the loci. However, the basic algorithmic strategy in SVDquest can be used with SNP data as well, as we now describe. When the number of species is small enough (i.e., at most 20), then SVDquest could be used in its unconstrained mode: quartet trees can be computed using SVDquartets, and then a species tree optimizing the MQSST score can be found using the dynamic programming algorithm in SVDquest. For datasets with larger numbers of species, the constrained version can be used in several ways. For example, the constraint set can be initialized to the bipartitions in the SVDquartets + PAUP* tree, and then enlarged using standard CA-ML analyses, PoMo and revPomo (as described earlier), trees computed on bootstrap replicates, or other techniques. Similarly, our study examined SVDquest* on supergene datasets (formed by randomly concatenating c-genes) and showed good accuracy, but true recombination will produce patterns that are somewhat different, and the impact of recombination on SVDquest* and summary methods needs to be explored.

Another limitation of our study is that the simulations we performed evolved sequences only with substitutions (i.e., no insertions and deletions), and so alignment estimation was not necessary; yet alignment error is quite common in practice, especially when the datasets span large evolutionary timescales. Although alignment error also increases gene tree estimation error, several studies have shown that accurate gene trees can be computed even in the presence of some alignment error (Liu et al., 2009), so that it is possible that SVDquest* and other site-based methods could be more negatively impacted than summary methods. Hence, the impact of alignment error is an important aspect to consider. If alignment error negatively impacts SVDquartets, it may be that approaches that select sites within alignments to use within SVDquartets will be helpful.

Finally, although SVDquest*-strict is fast enough to be used on whole genome datasets with moderately large numbers of species, we only tested SVDquest*-strict under conditions where all quartet trees could be computed. Therefore, when the number of species is large enough (i.e., 200 or more), then this becomes computationally infeasible. For this reason, when the number of species is too large, PAUP* uses random sampling on the quartets, uses SVDquartets to compute quartet trees, and then combines these quartet trees using its quartet amalgamation heuristics. In its current implementation, SVDquest cannot be used with such inputs, but the dynamic programming algorithm in SVDquest can be used with any way of weighting quartet trees, and so could be used with sparsely sampled quartet trees by assigning equal weights to all three quartet trees on any unsampled quartet. However, sparse sampling of quartet trees for use with quartet amalgamation methods has been shown to have reduced accuracy compared to analyses that use all the quartet trees (Swenson et al., 2011), suggesting that when the number of species makes SVDquest* inapplicable, summary methods or concatenation may be a better choice than SVDquartets-based approaches. Thus, the best modifications to SVDquest* to enable it to be used to good advantage on datasets with large numbers of species will require some investigation.

5. Summary

We presented SVDquest*, a site-based method for species tree estimation. Like the implementation within PAUP* (which we refer to as SVDquartets + PAUP*), SVDquest* operates by computing quartet trees using SVDquartets, and then seeks a species tree with the largest MQSST score. Unlike SVDquartets + PAUP*, which uses a heuristic search through treespace, SVDquest* uses an exact algorithm for this optimization problem, and achieves polynomial time by constraining the search space using a set of bipartitions on the species set that it computes from the input. By design, SVDquest* is guaranteed to obtain a score that is at least as large as the score produced using SVDquartets + PAUP*. In practice, SVDquest* typically finds trees with better MQSST scores than SVDquartets + PAUP*, especially on datasets with higher levels of gene tree estimation error and lower numbers of genes.

Our study evaluated SVDquest*-strict in comparison to two summary methods (ASTRAL and ASTRID), SVDquartets + PAUP*, and CA-ML under a wide range of ILS levels, numbers of species, and numbers of genes. Although our study was limited to conditions with at most 1000 genes and 50 species, we observed several significant and interesting trends. While ASTRAL and ASTRID can be more accurate than SVDquest*-strict when GTEE is low, SVDquest*-strict is typically more accurate than these summary methods when GTEE is high, as GTEE impacts summary methods directly, introducing error into the summary statistics they use to construct species trees. CA-ML is surprisingly accurate, and more accurate than the summary methods under conditions with high GTEE (even when ILS is high); interestingly, we also observed that sometimes SVDquest*-strict improves on CA-ML. Thus, the relative accuracy between these methods depends on the model condition, and in particular on the ILS and GTEE levels, but SVDquest*-strict provides advantages over the other coalescent-based methods under several biologically realistic conditions.

This study also shows that SVDquest*-strict is fast enough to use on genome-scale biological datasets. SVDquest*-strict includes calls to both ASTRAL and SVDquartets + PAUP*, and is otherwise very fast; hence, any dataset on which both of these methods can be run can be analyzed by SVDquest*-strict. Furthermore, a comparison of running times between these methods and concatenation suggests that for large enough datasets, concatenation analyses are likely to become computationally extremely expensive. For example, a concatenated maximum likelihood analysis of the 48-species avian phylogenomics dataset with 14,446 loci took more than 200 CPU years (Jarvis et al., 2014), while an analysis using the new implementation of ASTRAL took only 32 h after the gene trees were computed (Zhang et al., 2017). The calculation of 14,446 ML gene trees is expensive, but completes in well under a month (and is very fast if parallelized) (Jarvis et al., 2014). Hence, summary methods are generally computationally much more feasible than concatenation analyses for large datasets, which means that SVDquest*-strict is a computationally feasible approach for many genome-scale datasets.

This study adds to the current literature evaluating site-based approaches to species tree estimation. Although we did not find that SVDquest*-strict improved on the competing coalescent-based approaches under all conditions, our study shows that SVDquest*-strict can provide improved accuracy under some conditions with high GTEE. This trend suggests the potential for SVDquest*-strict to be particularly

beneficial for genome-scale datasets, where GTEE is likely to be high as a result of either ILS or variable rates of evolution across the genome. In addition, SVDquest*-strict had very good accuracy on supergene datasets, suggesting it may be robust to failure to detect recombination events. Finally, the relative performance between SVDquest*-strict (and other methods based on SVDquartets), summary methods, and CA-ML might well depend on the number of loci, so that SVDquest*-strict (or other methods based on SVDquartets) could become the method of choice when the number of loci and ILS level are both very large.

Funding

This work was supported by the NSF Graduate Research Fellowship Program under grant DGE-1144245 (to PV) and by NSF grant CCF-1535977 (to TW).

Acknowledgments

The authors thank Associate Editor Guillermo Orti and the two anonymous reviewers for helpful comments, and also Michael Nute, Erin Molloy, and Sarah Christensen for comments on earlier drafts.

Appendix A

A.1. Commands

The following commands were used to run the software in the experimental analyses:

- RAxML 8.2.6, for estimating gene trees and for unpartitioned concatenated analyses:
  ```
  raxml -m GTRGAMMA -p 12345 -s < path to input file >
  ```
- PAUP* 4.0a151, for computing SVD quartet weights:
  ```
  exe < path to input file > ;
  svd showScores = yes evalQuartets = all treeInf = None
  qformat = qmc replace = no;
  ```
- PAUP* 4.0a151, for computing SVDquartets + PAUP* species trees:
  ```
  exe < path to input file > ;
  svd showScores = no evalQuartets = all qformat = qmc
  replace = no;
  savetrees file= < output file > format = newick;
  ```
  This is the default setting within PAUP*.
- SVDquest v1.0, for computing species trees (along with all SIESTA variants and runtime information):
  ```
  SVDquest -g < gene trees > -q < svdquartets
  weights >
      -o < output file > -strict -majority -greedy
  -count
      -single -timing
  ```
- SVDquest* v1.0, for computing species trees (along with all SIESTA variants and runtime information):
  ```
  SVDquest -g < gene trees > -q < svdquartets
  weights >
  -e < SVDquartets + PAUP* tree >
      -o < output file > -strict -majority -greedy
      -count -single -timing
  ```
- ASTRAL v4.10.2, for computing species trees:
  ```
  ASTRAL -i < gene trees > -o < output file >
  ```
- ASTRAL v4.10.2, for computing the constraint set X (called internally by SVDquest):
  ```
  ASTRAL -i < gene trees > -o < output file >
  -k searchspace_norun > /dev/null
  ```
- ASTRID v1.4, for computing species trees:
  ```
  ASTRID -i < gene trees > -o < output file >
  ```
- Dendropy v4.0.3, for computing species tree and gene tree estimation error:
  ```
  dendropy.calculate.treecompar-
  e.false_positives_and_negatives()
  ```

Appendix B. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ympev.2018.03.006.

References

Alexander, A.M., Su, Y.-C., Oliveros, C.H., Olson, K.V., Travers, S.L., Brown, R.M., 2017. Genomic data reveals potential for hybridization, introgression, and incomplete lineage sorting to confound phylogenetic relationships in an adaptive radiation of narrow-mouth frogs. Evolution 71 (2), 475–488.

Allman, E.S., Degnan, J.H., Rhodes, J.A., 2017. Split Probabilities and Species Tree Inference Under The Multispecies Coalescent Model. arXiv preprint arXiv:1704.04268.

Anderson, B.M., Thiele, K.R., Krauss, S.L., Barrett, M.D., 2017. Genotyping-by-sequencing in a species complex of Australian Hummock Grasses (Triodia): methodological insights and phylogenetic resolution. PloS one 12 (1), e0171053.

Bayzid, M.S., Warnow, T., 2013. Naive binning improves phylogenomic analyses. Bioinformatics 29 (18), 2277–2284.

Bayzid, M.S., Mirarab, S., Boussau, B., Warnow, T., 2015. Weighted statistical binning: enabling statistically consistent genome-scale phylogenetic analyses. PloS One 10 (6), e0129183.

Boucher, F., Casazza, G., Szövényi, P., Conti, E., 2016. Sequence capture using rad probes clarifies phylogenetic relationships and species boundaries in Primula sect. Auricula. Molec. Phylogenet. Evol. 104, 60–72.

Bryant, D., Steel, M., 2001. Constructing optimal trees from quartets. J. Algor. 38 (1), 237–259.

Bryant, D., Bouckaert, R., Felsenstein, J., Rosenberg, N.A., RoyChoudhury, A., 2012. Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. Molec. Biol. Evol. 29 (8), 1917–1932.

Campillo, L.C., 2016. Use of Genomic Data to Resolve Gene Tree Discordance in a Southeast Asian Genus: Readdressing Paraphyly in the Spiderhunter (Nectariniidae, Arachnothera) Phylogeny. Master's thesis. University of Kansas.

Chifman, J., Kubatko, L., 2014. Quartet inference from SNP data under the coalescent model. Bioinformatics 30 (23), 3317–3324.

Chou, J., Gupta, A., Yaduvanshi, S., Davidson, R., Nute, M., Mirarab, S., Warnow, T., 2015. A comparative study of SVDquartets and other coalescent-based species tree estimation methods. BMC Genom. 16 (Suppl 10), S2.

Crowl, A.A., Myers, C., Cellinese, N., 2017. Embracing discordance: phylogenomic analyses provide evidence for allopolyploidy leading to cryptic diversity in a mediterranean campanula (campanulaceae) clade. Evolution 71 (4), 913–922.

Dasarathy, G., Nowak, R., Roch, S., 2015. Data requirement for phylogenetic inference from multiple loci: a new distance method. IEEE/ACM Trans. Comput. Biol. Bioinf. 12 (2), 422–432.

De Maio, N., Schrempf, D., Kosiol, C., 2015. PoMo: an allele frequency-based approach for species tree estimation. Syst. Biol. 64 (6), 1018–1031.

de Oca, A.N.-M., Barley, A.J., Meza-Lázaro, R.N., García-Vázquez, U.O., Zamora-Abrego, J.G., Thomson, R.C., Leaché, A.D., 2017. Phylogenomics and species delimitation in the knob-scaled lizards of the genus Xenosaurus (squamata: Xenosauridae) using ddRADseq data reveal a substantial underestimation of diversity. Molec. Phylogenet. Evol. 106, 241–253.

DeGiorgio, M., Degnan, J.H., 2010. Fast and consistent estimation of species trees using supermatrix rooted triples. Molec. Biol. Evol. 27 (3), 552–569.

DeGiorgio, M., Degnan, J.H., 2014. Robustness to divergence time underestimation when inferring species trees from estimated gene trees. Syst. Biol. 63 (1), 66–82.

Gaither, J., Kubatko, L., 2016. Hypothesis tests for phylogenetic quartets, with applications to coalescent-based species tree inference. J. Theoret. Biol. 408, 179–186.

Gatesy, J., Springer, M., 2014. Phylogenetic analysis at deep timescales: Unreliable gene trees, bypassed hidden support, and the coalescence/concatalescence conundrum. Molec. Phylogenet. Evolut. 80 (November), 231–266.

He, K., Shinohara, A., Helgen, K.M., Springer, M.S., Jiang, X.-L., Campbell, K.L., 2017. Talpid mole phylogeny unites shrew moles and illuminates overlooked cryptic species diversity. Molec. Biol. Evol. 34 (1), 78–87.

Heled, J., Drummond, A.J., 2010. Bayesian inference of species trees from multilocus data. Molec. Biol. Evol. 27 (3), 570–580.

Hime, P.M., Hotaling, S., Grewelle, R.E., O'Neill, E.M., Voss, S.R., Shaffer, H.B., Weisrock, D.W., 2016. The influence of locus number and information content on species delimitation: an empirical test case in an endangered Mexican salamander. Molec. Ecol. 25 (23), 5959–5974.

Hosner, P.A., Braun, E.L., Kimball, R.T., 2016. Rapid and recent diversification of curassows, guans, and chachalacas (Galliformes: Cracidae) out of Mesoamerica: phylogeny inferred from mitochondrial, intron, and ultraconserved element sequences. Molec. Phylogenet. Evol. 102, 320–330.

Hou, Z.-c., Romero, R., Wildman, D.E., 2009. Phylogeny of the ferungulata (mammalia: Laurasiatheria) as determined from phylogenomic data. Molec. Phylogenet. Evol. 52 (3), 660–664.

Huang, H., He, Q., Kubatko, L.S., Knowles, L.L., 2010. Sources of error inherent in species-tree estimation: impact of mutational and coalescent effects on accuracy and implications for choosing among different methods. Syst. Biol. 59 (5), 573–583.

Jarvis, E.D., Mirarab, S., Aberer, A.J., Li, B., Houde, P., Li, C., Ho, S.Y., Faircloth, B.C.,

Nabholz, B., Howard, J.T., et al., 2014. Whole-genome analyses resolve early branches in the tree of life of modern birds. Science 346 (6215), 1320–1331.
Jiang, T., Kearney, P., Li, M., 2001. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. SIAM J. Comput. 30 (6), 1942–1961.
Knowles, L.L., 2009. Estimating species trees: methods of phylogenetic analysis when there is incongruence across genes. Syst. Biol. 58 (5), 463–467.
Kubatko, L.S., Degnan, J., 2007. Inconsistency of phylogenetic estimates from concatenated data under coalescence. Syst. Biol. 56, 17–24.
Lanier, H., Knowles, L., 2012. Is recombination a problem for species-tree analyses? Syst. Biol. 61 (4), 691–701.
Leaché, A.D., Chavez, A.S., Jones, L.N., Grummer, J.A., Gottscho, A.D., Linkem, C.W., 2015. Phylogenomics of phrynosomatid lizards: conflicting signals from sequence capture versus restriction site associated DNA sequencing. Gen. Biol. Evol. 7 (3), 706–719.
Leavitt, S.D., Grewe, F., Widhelm, T., Muggia, L., Wray, B., Lumbsch, H.T., 2016. Resolving evolutionary relationships in lichen-forming fungi using diverse phylogenomic datasets and analytical approaches. Scient. Rep., 6.
Liu, K., Raghavan, S., Nelesen, S., Linder, C.R., Warnow, T., 2009. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science 324 (5934), 1561–1564.
Liu, L., 2008. BEST: Bayesian estimation of species trees under the coalescent model. Bioinformatics 24 (21), 2542–2543.
Liu, L., Yu, L., 2011. Estimating species trees from unrooted gene trees. Syst. Biol. 60 (5), 661–667.
Liu, L., Yu, L., Edwards, S.V., 2010. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. BMC Evolut. Biol. 10 (1), 302.
Maddison, W.P., 1997. Gene trees in species trees. Syst. Biol. 46 (3), 523–536.
Mallo, D., Posada, D., 2016. Multilocus inference of species trees and DNA barcoding. Phil. Trans. R. Soc. B 371 (1702), 20150335.
Mallo, D., Oliviera Martins, L., Posada, D., 2015. SimPhy: comprehensive simulation of gene, locus, and species trees at the genome-wide level. Syst. Biol. 65 (2), 334–344. < https://code.google.com/p/simphy-project > .
Manthey, J.D., Campillo, L.C., Burns, K.J., Moyle, R.G., 2016. Comparison of target-capture and restriction-site associated dna sequencing for phylogenomics: a test in cardinalid tanagers (aves, genus: Piranga). Syst. Biol. 65 (4), 640–650.
Manthey, J.D., Geiger, M., Moyle, R.G., 2017. Relationships of morphological groups in the northern flicker superspecies complex (colaptes auratus & c. chrysoides). System. Biodiv. 15 (3), 183–191.
McCormack, J.E., Harvey, M.G., Faircloth, B.C., Crawford, N.G., Glenn, T.C., Brumfield, R.T., 2013. A phylogeny of birds based on over 1500 loci collected by target enrichment and high-throughput sequencing. PLoS One 8 (1), e54848.
Meiklejohn, K., Faircloth, B., Glenn, T., Kimball, R., Braun, E., 2016. Analysis of a rapid evolutionary radiation using ultraconserved elements: evidence for a bias in some multispecies coalescent methods. Syst. Biol. 65 (4), 612–627. http://dx.doi.org/10.1093/sysbio/syw014.
Mirarab, S., Warnow, T., 2015. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. Bioinformatics 31 (12), i44–i52.
Mirarab, S., Reaz, R., Bayzid, M.S., Zimmermann, T., Swenson, M., Warnow, T., 2014. ASTRAL: Accurate Species TRee ALgorithm. Bioinformatics 30 (17), i541–i548.
Mirarab, S., Bayzid, M.S., Warnow, T., 2016. Evaluating summary methods for multilocus species tree estimation in the presence of incomplete lineage sorting. Syst. Biol. 65 (3), 366–380.
Mitchell, N., Lewis, P.O., Lemmon, E.M., Lemmon, A.R., Holsinger, K.E., 2017. Anchored phylogenomics improves the resolution of evolutionary relationships in the rapid radiation of Protea L. Am. J. Bot. 104 (1), 102–115.
Molloy, E.K., Warnow, T., 2017. To include or not to include: the impact of gene filtering on species tree estimation methods. Syst. Biol. 67 (2), 285–303.
Mossel, E., Roch, S., 2017. Distance-based species gree estimation: Information-theoretic trade-off between number of loci and sequence length under the coalescent. Ann. Appl. Probab. 27 (5), 2926–2955.
Moyle, R.G., Oliveros, C.H., Andersen, M.J., Hosner, P.A., Benz, B.W., Manthey, J.D., Travers, S.L., Brown, R.M., Faircloth, B.C., 2016. Tectonic collision and uplift of Wallacea triggered the global songbird radiation. Nat. Commun. 7.
Ohno, S., 1970. Evolution by Gene Duplication. Springer-Verlag, New York.
Patel, S., Kimball, R.T., Braun, E.L., 2013. Error in phylogenetic estimation for bushes in the tree of life. J. Phylogenet. Evolut. Biol. 1, 110.
Robinson, D.F., Foulds, L.R., 1981. Comparison of phylogenetic trees. Math. Biosci. 53 (1), 131–147.
Roch, S., Steel, M., 2015. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. Theoret. Popul. Biol. 100, 56–62.
Roch, S., Warnow, T., 2015. On the robustness to gene tree estimation error (or lack thereof) of coalescent-based species tree methods. Syst. Biol. 64 (4), 663–676.
Sayyari, E., Mirarab, S., 2016. Fast coalescent-based computation of local branch support from quartet frequencies. Molec. Biol. Evol. 33 (7), 1654–1668.
Schrempf, D., Minh, B.Q., Maio, N.D., von Haeseler, A., Kosiol, C., 2016. Reversible polymorphism-aware phylogenetic models and their application to tree inference. J. Theoret. Biol. 407 (Supplement C), 362–370.
Song, S., Liu, L., Edwards, S.V., Wu, S., 2012. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. Proc. Nat. Acad. Sci. 109 (37), 14942–14947.
Springer, M., Gatesy, J., 2016. The gene tree delusion. Molec. Phylogenet. Evol. 94, 1–33.
Stamatakis, A., 2006. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22 (21), 2688–2690.
Sukumaran, J., Holder, M., 2010. Dendropy: a Python library for phylogenetic computing. Bioinformatics 26 (12), 1569–1571.
Swenson, M.S., Suri, R., Linder, C.R., Warnow, T., 2011. An experimental study of Quartets MaxCut and other supertree methods. Algor. Molec. Biol. 6 (1), 7.
Swofford, D., 2003. PAUP*. Phylogenetic Analysis Using Parsimony (*and other methods). Version 4. Sinauer Associates, Sunderland, Massachussets.
Vachaspati, P., Warnow, T., 2015. ASTRID: accurate species trees from internode distances. BMC Genom. 16 (Suppl 10), S3.
Vachaspati, P., Warnow, T., 2017a. Enhancing searches for optimal trees using SIESTA. In: Meidanis, J., Nakhleh, L. (Eds.), Proceedings of the 15th International Workshop, RECOMB -Comparative Genomics. Barcelona, Spain, October 4–6. Springer International Publishing, pp. 232–255.
Vachaspati, P., Warnow, T., 2017b. FastRFS: fast and accurate Robinson-Foulds Supertrees using constrained exact optimization. Bioinformatics 33 (5), 631–639.
White, N.D., Barrowclough, G.F., Groth, J.G., Braun, M.J., 2016. A multi-gene estimate of higher-level phylogenetic relationships among nightjars (aves: Caprimulgidae). Ornitología Neotropical 27, 223–236.
White, N.D., Mitter, C., Braun, M.J., 2017. Ultraconserved Elements Resolve the Phylogeny of Potoos (Aves: Nyctibiidae). J. Avian Biol. doi:http://dx.doi.org/10.111/jav.01313.
Wickett, N.J., Mirarab, S., Nguyen, N., Warnow, T., Carpenter, E., Matasci, N., Ayyampalayam, S., Barker, M.S., Burleigh, J.G., Gitzendanner, M.A., et al., 2014. Phylotranscriptomic analysis of the origin and early diversification of land plants. Proc. Nat. Acad. Sci. 111 (45), E4859–E4868.
Woese, C.R., 2002. On the evolution of cells. Proc. Nat. Acad. Sci. 99 (13), 8742–8747.
Zhang, C., Sayyari, E., Mirarab, S., 2017. ASTRAL-III: Increased scalability and impacts of contracting low support branches. In: Meidanis, J., Nakhleh, L. (Eds.), Comparative Genomics: 15th International Workshop, RECOMB CG 2017, Barcelona, Spain, October 4–6, 2017, Proceedings. Springer International Publishing, Cham, pp. 53–75.
Zhou, X., Xu, S., Xu, J., Chen, B., Zhou, K., Yang, G., 2012. Phylogenomic analysis resolves the interordinal relationships and rapid diversification of the Laurasiatherian mammals. Syst. Biol. 61 (1), 150.
Zimmermann, T., Mirarab, S., Warnow, T., 2014. BBCA: Improving the scalability of *BEAST using random binning. BMC Genom. 15, S11 Special issue for RECOMB-Comparative Genomics.