BIOINFORMATICS

Vol. 00 no. 00 2013
Pages 1–16

Supplementary Materials, Naive Binning Improves Phylogenomic Analyses

Md. Shamsuzzoha Bayzid and Tandy Warnow

Department of Computer Science, The University of Texas at Austin, Austin, Texas 78712, USA,

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXX

1 OVERVIEW

These supplementary materials present additional details about the methods used (Section 2) and results (Section 3), and also present some additional discussion (Section 4).

2 METHODS

2.1 Overview

We used previously generated datasets from two studies (17-taxon datasets from Yu *et al*. (2011a,b) and 11-taxon datasets from Chung and Ané (2011)), and evaluated several pipelines for estimating species trees and gene trees for these datasets. We included three ways of estimating gene trees: RAxML and FastTree-2 to estimate maximum likelihood trees from the sequence alignments, and *BEAST to co-estimate gene trees and species trees. We explored several ways of estimating species trees: BUCKy, *BEAST, MRP, Greedy Consensus, Phylonet-MDC, MP-EST, and CA-ML. Each analysis produced a set of estimated gene trees and species trees, which we could evaluate for accuracy by comparing them to the model gene and species trees. We noted the missing branch rate (false negative, or FN error) and running time usage for each method. We compared the methods and determined which results were statistically significant using Wilcoxon signed rank T-test, with $\alpha = 0.05$.

We used 11-taxon datasets with 100 genes (100 replicates) and 17-taxon datasets with up to 32 genes (also with 100 replicates). The 11-taxon datasets were generated by model conditions that violate the molecular clock and came in two forms: datasets that were generated under a high level of ILS (called "strongILS") and datasets that were generated under a low level of ILS (called "weakILS"). The 17-taxon datasets were generated under the molecular clock and had a high level of ILS; these came in two forms: 8-gene and 32-gene datasets.

Slow methods: Pipelines that included *BEAST or BUCKy were too computationally intensive to run on all the replicates; we therefore only explored these methods on a subset of the replicates. Specifically, we never ran *BEAST on 100 replicates of any model condition. Instead, we ran *BEAST (binned and unbinned) on 20 replicates of the 11-taxon datasets with at most 50 genes, and 20 replicates of the 17-taxon datasets with 8 genes and with 32 genes. For BUCKy, we were able to run it on 20 replicates (unbinned) of all model conditions tested. In addition, when we ran BUCKy with binning, we were able to run it on 100 replicates of the 11-taxon strongILS datasets and 100 replicates of the 17-taxon 32-gene datasets. The remaining methods were all fast enough for us to run on all 100 replicates of all model conditions.

Standard error: The error bars in the figures correspond to the standard error, given by $S/\sqrt{n}$, where S is the standard deviation and n is the number of datapoints.

2.2 Datasets

All datasets are available online at http://www.cs.utexas.edu/users/phylo/datasets/ILS/.

11-taxon datasets: The 11-taxon datasets were created for the study in Chung and Ané (2011), and simulated under a complex process to ensure substantial heterogeneity between genes and to deviate from the molecular clock. There were two types of model trees – ones with long branches (LB) that produce low levels of ILS, and ones with short branches (SB) that produce high levels of ILS. We have referred to these two different model conditions as weakILS and strongILS, respectively. Here we present the text from the paper, modified only to remove the references to other papers and figures.

Text from Chung and Ané (2011):

"We generated DNA alignments from 5-taxon and 11-taxon species trees. An asymmetric tree topology was chosen on 5 taxa, as this was proven to be more difficult to reconstruct in the presence of gene-to-gene discordance (Kubatko and Degnan, 2007). Our 11-taxon tree contains two copies of our 5-taxon tree (subtree with taxa 1, 2, 3, 4 and subtree with taxa 5, 7, 9, 10, both with taxon 11 as an outgroup). In one of the two copies, taxa 6 and 8 were added in order to detect potential effects of the number of taxa on the estimation of internal edges CFs. For each species tree topology, two sets of branch lengths were considered. One set had long internal branches (LB), whereas the other set had some short internal branches (SB). Species tree branch lengths were measured in coalescent units, as obtained by dividing the number of generations by the effective population size. Under the coalescent model, branch lengths in coalescent units determine the proportion of genes that share the species tree topology and the proportion of genes that have any given conflicting topology.

"In order to simulate multilocus data sets, 10, 50, or 100 unlinked gene trees were generated along the species trees. We used an

effective size of 50,000 haploid individuals in each population. The numbers of generations between speciations were determined by multiplying branch lengths in coalescent units by the population size.

"HGTsimul was used to simulate a Poisson-distributed number of genomic rate change events (with a mean of three changes) on the species tree, for genomic departure from the molecular clock. Lineage-specific rates were simulated from a gamma distribution with mean 1 and shape parameter 2.0. For each gene, branch lengths obtained from Serial SimCoal were multiplied by these lineage-specific rates, then further multiplied by a common factor to obtain a randomly chosen gene diameter (uniform in 0.024 and 0.037 substitutions per site). Next, gene tree branch lengths were modified in a gene-specific manner: for each individual gene, a Poisson-distributed number of rate change events (three changes on average) were placed on the gene tree, whose branch lengths were multiplied by a gamma-distributed rate (mean 1 and shape parameter 2.0) in between these gene-specific rate change events. Finally, sequences were simulated using the JukesCantor (JC) model and no site-specific rate variation, for computational feasibility.

"In summary, our simulations included important factors that contribute to heterogeneity among genes, such as heterogeneity in the overall rate of evolution, departure from clock-like evolution, and topological discordance."

17-taxon datasets: We used 17-taxon datasets that were simulated for Yu *et al.* (2011a,b), and provided to us by the authors. In this simulation, species trees were generated using the Yule module using Mesquite (Maddison and Maddison (2011)), and with total branch length of 800,000 generations, not counting the outgroup. Two collections of gene trees were simulated in this model: one with only 8 gene trees and one with 32 gene trees; however, the 8-gene dataset is not a subset of the 32-gene dataset. These gene trees were simulated within the species trees using the "Coalescence Contained Within Current Tree" module within Mesquite, with an effective population size of $N_e = 100,000$. Then sequences were evolved down the gene trees under the Jukes-Cantor model (without any rates-across-sites), using Seq-gen (Rambaut and Grassly (1997)), with each sequence having length 2000.

Thus, these sequences evolve under a strong molecular clock, and there is no rate variation across sites or between different genes.

Subsampling: Our 11-taxon datasets (both strongILS and weakILS) contain 100 replicates each containing 100 genes. To evaluate the impact of the number of genes on the performance of different methods, we subsample different number of genes (5, 10, 25, and 50 genes) from our available set of 100 genes. We randomly subsample a particular number of genes (5, 10, 25 etc.) from a replicate that contains 100 genes. We generated 20 set of such subsamples from each replicate. For experiments analyzing 11-taxon datasets with up to 50 genes, we generated either 20 replicates (all from one replicate alignment) or 100 replicates (from 5 different replicate alignments). For experiments analyzing 11-taxon datasets with 100 genes, we used all 100 replicates. The 17-taxon datasets came in two collections - one with 8 genes, and one with 32 genes. Therefore, for the analyses with 17-taxon datasets, we used 20 or 100 replicates of the datasets in each collection.

2.3 Methods

2.3.1 Gene tree estimation We used three methods for estimating gene trees: FastTree-2, RAxML, and *BEAST.

- FastTree-2 (v. 2.1.3 SSE3) (Price *et al.* (2010)). We used FastTree-2 to estimate ML gene trees from the sequence alignments, using the following command:
  `FastTree -gtr -nt <sequenceAlignment> > <outputFile>`
- RAxML: We ran RAxML v. 7.3.1 (Stamatakis (2006)) to estimate ML gene trees from sequence alignments. We ran 20 runs of RAxML on each of the alignments, using the following command:
  `raxmlHPC-PTHREADS -T 2 -m GTRGAMMA -s <sequenceAlignment> -n <output-name> -N 20 -p 1234`.
  For estimating bootstrap branch support for the RAxML-estimated trees, we generated 400 bootstrap trees per each gene and then drew branch support on the edges of the ML tree by using these 400 bootstrap trees. The proportion of the bootstrap trees in which a particular split is found is taken to be the degree of support for that split. We then produced a 75%-branch support version of each estimated gene tree by contracting all edges with support below 75%.
- *BEAST: We ran *BEAST in its default setting to co-estimate gene trees and species trees; details are provided below under "Species Tree Estimation".

2.3.2 Species Tree Estimation

**BEAST:* We used *BEAST v. 1.6.2 (Heled and Drummond (2010)) in default mode to co-estimate the gene trees and species tree on every dataset. For a given *BEAST analysis, we discarded the first 10% of the trees returned by the analysis, and then sampled one (1) out of each 1000 of the remaining trees. We return the maximum credibility species tree and gene trees from the *BEAST output. On the 11-taxon datasets with 5, 10, 25, and 50 genes, we ran *BEAST for 80M, 120M, 160M, and 200M MCMC iterations, respectively. We did not run *BEAST to convergence on the 100 gene datasets. On the 17-taxon datasets, we ran *BEAST for 200M MCMC iterations.

We were able to run *BEAST on 11-taxon datasets with up to 50 genes. We observed very high ESS values (all the ESS values were greater than 100, and many of them were in the thousands) except for 5 and 10-gene cases, where some ESS values were less than 100. On 17-taxon 8 and 32-gene datasets, we observed very high ESS values (all the ESS values are greater than 100, and many of them were in the thousands) when we ran it for 200M iterations. When used with binning on 11- and 17-taxon datasets, we ran 50M iterations on the supergenes and observed very high ESS values.

We ran *BEAST on 11-taxon 100-gene datasets with 50M iterations; each of these analyses took around 100 hours per replicate dataset, but produced very poor ESS values (we observed many parameters having less than 100 ESS). Therefore, we did not report results for *BEAST on the 11-taxon 100-gene datasets.

Additional information about the running time for *BEAST is given below.

BUCKy: We used BUCKy v. 1.4.0 (Ané *et al.*, 2007; Larget *et al.*, 2010) in default mode; thus, $\alpha = 1$. As noted in the paper, most of the experiments involving BUCKy were run with input gene tree distributions computed using RAxML. However, we also used *BEAST in Experiment 4. We used the following command:

```
bucky -n <numberOfGenerations>
-o <outputFileRoot> <inputFiles>
```

For the analyses with distributions produced by *BEAST, we ran 80M, 120M, 160M, 200M iterations of *BEAST for 5, 10, 25, and 50 genes, respectively, and we sampled one tree out of each 1000 iterations; this produced 80K, 120K, 160K, and 200K trees in each distribution for datasets with 5, 10, 25, and 50 genes, respectively. We discarded the first 10% of these trees as *burn-in*, and used the remaining trees as the input to BUCKy. We ran BUCKy with 30M generations for 5- and 10-gene cases, 40M generations on 25-gene cases, and 50M generations for 50-gene cases. For 17-taxon datasets, we ran 40M generations. Note therefore that we did not test BUCKy on gene tree distributions estimated by *BEAST on the 100-gene datasets, because *BEAST was too expensive to run on these datasets.

We also ran BUCKy on RAxML-bootstrap trees, using 400 bootstrap trees per gene. We ran 500M generations of BUCKy for 5-, 10-, and 25-gene cases, and 200M generations for 50-gene cases. When run with binning, we ran 500M and 50M generations of BUCKy on 11-taxon strongILS and weakILS datasets, respectively. On 17-taxon datasets (both binned and unbinned), we ran 100M generations of BUCKy.

As with *BEAST, there is no strict condition for convergence of BUCKy; however, an "Average SD of mean sample-wide CF" below 0.05 may be adequate to have high confidence about the convergence. Samples of the standard deviation (SD) for the CF statistics for different BUCKy analyses follow:

- 11-taxon 50-gt, RAxML trees: SD = 0.000 to $\sim$ 0.004
- 11-taxon 50-gt, *BEAST trees: SD = 0.000
- 11-taxon 25-gt, RAxML trees: SD = 0.001 to $\sim$ 0.006
- 11-taxon 25-gt, *BEAST trees: SD = 0.000
- 11-taxon 10-gt, RAxML trees: SD = 0.000 to $\sim$ 0.007
- 11-taxon 10-gt, *BEAST trees: SD = 0.000
- 11-taxon 5-gt, RAxML trees: SD = 0.000 to $\sim$ 0.001
- 11-taxon 5-gt, *BEAST trees: SD = 0.000
- 17-taxon 32-gt, RAxML trees: SD = 0.000
- 11-taxon 32-gt, *BEAST trees: SD = 0.000 to $\sim$ 0.003
- 17-taxon 8-gt, RAxML trees: SD = 0.000
- 11-taxon 8-gt, *BEAST trees: SD = 0.000

The following statistics are for the binned analyses:

- 11-taxon 50-gt (10 bins): SD = 0.000
- 11-taxon 25-gt (5 bins): SD = 0.000
- 17-taxon 32-gt (8 bins): SD = 0.000

BUCKy returns two trees: one is the population tree (referred to as "BUCKy-pop") and the other is the concordance tree (referred to as "BUCKy-con"). BUCKy-pop is statistically consistent in the presence of ILS, but BUCKy-con is not.

MP-EST: We used MP-EST v. 1.2 (Liu *et al.*, 2010) to estimate the species tree from input gene trees. MP-EST requires rooted gene trees as input; our datasets all include outgroups, and we root the estimated gene trees using these outgroups. MP-EST is statistically consistent in the presence of ILS, and maximizes a pseudo-likelihood function in order to estimate the species tree. We ran it in its default setting with MAXROUND=1000000.

Matrix Representation with Parsimony (MRP): MRP (Ragan (1992)) is a supertree method that we use as a consensus method (since all the gene trees have the same set of taxa). MRP has two steps: in the first step, it encodes each input source tree as a matrix over {0,1, ?}, with one row for each taxon in the full set of taxa, and with each character corresponding to one edge bipartition in one source tree. These matrices are then concatenated together to obtain a single matrix. The MRP supertree is obtained by analyzing the character matrix using a maximum parsimony approach.

We created MRP matrices using a custom Java program, and solved MRP heuristically using the default approach implemented in PAUP* (v. 4. 0b10) (Swofford (1996)). By default, PAUP* generates an initial tree through random sequence addition (adding sequences one at a time in the most parsimonious position in a tree) and then performs Tree Bisection and Reconnection (TBR) moves until it reaches a local optimum. This process is repeated 1000 times, and at the end the most parsimonious tree is returned. When multiple trees are found with the same maximum parsimony score, the "extended majority consensus" of those trees is returned.

Below is the PAUP* block:

```
begin paup;
set criterion=parsimony maxtrees=1000
increase=no;
hsearch start=stepwise addseq=random
nreps=100 swap=tbr;
filter best=yes;
savetrees file = <treeFile> replace=yes
format=altnex;
contree all/ strict=yes
treefile = <strictConsensusTreeFile>
replace=yes;
tcontree all/ majrule=yes strict=no
treefile = <majorityConsensusTreeFile>
replace=yes;
contree all/ majrule=yes strict=no
le50=yes
treefile = <greedyConsensusTreeFile>
replace=yes;
log stop;
quit; end;
```

Phylonet: We use the Phylonet v. 2.4 (Than *et al.*, 2008) to solve MDC heuristically or exactly, depending on the dataset size. For the 11-taxon datasets, we use the version that is guaranteed to solve MDC optimally, and for the 17-taxon datasets we use the heuristic

version. The input to Phylonet in each case is a set of gene trees restricted to the branches with bootstrap support at least 75% (i.e., with all low-support branches contracted). The version of Phylonet we used on these partially resolved gene tree estimates solves the following problem: Given a set of (partially resolved) unrooted gene trees $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ (not necessarily on the same set of taxa), find binary refinements t_i^* for each t_i, and species tree T, so that the MDC score of T with respect to $\mathcal{T}^* = \{t_1^*, t_2^*, \ldots, t_k^*\}$ is minimum among all such sets $\mathcal{T}^*$ and species trees T. Thus, Phylonet solves a constrained version of MDC, taking bootstrap support into consideration. See Yu *et al.* (2011a,b) for more details and the proof of correctness. See also Bayzid and Warnow (2012) for the proof that Phylonet handles missing taxa correctly.

Greedy Consensus: We ran the greedy consensus technique (also called the extended majority consensus) using PAUP* v. 4.0b10. The greedy consensus begins by computing the majority consensus (the tree whose edge-induced taxon bipartitions are those that appear in more than half of the input trees), and then adds compatible bipartitions, one at a time, in an order reflecting the frequency with which each bipartition appears.

Below is the PAUP* block:

```
begin paup;
set autoclose = yes warntree = no
warnreset = no notifybeep = no
monitor = yes taxlabels = full;
set criterion = parsimony;
set increase = auto;
gettrees file = <nexusFile> allblocks = yes
warntree = no unrooted = yes;
contree all / strict = no
majrule = yes le50 = yes
treefile = <greedyConsensusTreeFile>;
end;
```

Combined Analyses using Maximum Likelihood (CA-ML): This method concatenates the alignments on all genes into one super-alignment, and then estimates a tree from the super-alignment using maximum likelihood, treating the alignment as unpartitioned. We used RAxML for this analysis, using the following command:

```
raxmlHPC-PTHREADS -T 2 -m GTRGAMMA
-s <sequence> -n <output-name> -N 10
-p 1234.
```

2.4 Running time

**BEAST running time:* We tested three 11-taxon datasets with 100 genes without using binning and using 50M iterations; these analyses ranged from 80 to 150 hours. Based on the ESS values, none of these came close to convergence; hence, the running times here are suggestive of lower bounds for time needed to use *BEAST. However, these datasets were run on Condor, and so running times are approximate.

The remaining analyses were on at most 50 genes, or used binning to analyze 100 genes (and so had only 20 supergenes). Each analysis is of one dataset only, and was done on a dedicated 64-bit machine with 32173 MB memory.

- Unbinned analyses
  - 11-taxon strongILS 50-gt, 200M iterations: 57 hours
  - 11-taxon strongILS 25-gt 160M iterations: 20 hours
  - 17-taxon 32-gt, 200M iterations: 35 hours
- Binned analyses (5 genes per bin)
  - 11-taxon strongILS 100-gt with 20 bins with 50M iterations: 10 hours using 4 threads
  - 11-taxon strongILS 50-gt with 10 bins (5 genes in each bin), 50M iterations: 6.4 hours
  - 11-taxon 25-gt strongILS with 5 bins, 50M iterations: 3.1 hours
  - 17-taxon 32-gt with 8 bins, 50M: 5.6 hours

BUCKy running time: We performed several BUCKy analyses for all three model conditions. These analyses showed that the running time was determined by the the type of input distribution, and whether it was from one of the two 11-taxon model conditions or from the 17-taxon model condition; however, 11-taxon strongILS and 11-taxon weakILS analyses took the same amount of time.

Results on unbinned analyses with RAxML gene tree distributions:

- 11-taxon 100-gt, RAxML trees, 200M generations: 2.2 hours
- 11-taxon 50-gt, RAxML trees, 200M generations: 2.1 hours
- 11-taxon 25-gt, RAxML trees, 500M generations: 3.5 hours
- 11-taxon 10-gt, RAxML trees, 500M generations: 2.36 hours
- 11-taxon 5-gt, RAxML trees, 500M generations: 1.75 hours
- 17-taxon 8-gt, RAxML trees, 100M generations: 40 mins
- 17-taxon 32-gt, RAxML trees, 100M generations: 2.07 hours

Results on unbinned analyses with *BEAST gene tree distributions:

- 11-taxon 50-gt, *BEAST trees, 50M generations: 21 mins
- 11-taxon 25-gt, *BEAST trees, 40M generations: 11 mins
- 11-taxon 10-gt, *BEAST trees, 30M generations: 7 mins
- 11-taxon 5-gt, *BEAST trees, 30M generations: 3 mins
- 17-taxon 32-gt, *BEAST trees, 40M generations: 15 mins
- 17-taxon 8-gt, *BEAST trees, 40M generations: 6 mins

Note the difference in running time between *BEAST and RAxML distributions, indicating that BUCKy converges with fewer MCMC iterations when run with *BEAST distributions than when run with RAxML bootstrap distributions! However, *BEAST takes much more time to run, so the total running time when based on *BEAST is much longer.

Running time for binned analyses:

- 11-taxon 25-gt (5 bins), RAxML trees, 500M generations: 1.1 hours

- 11-taxon 50-gt (10 bins), RAxML trees, 500M generations: 1.75 hours
- 17-taxon 32-gt (8 bins), RAxML trees, 100M generations: 13 mins

RAxML bootstrapping: We generated 400 bootstrap replicates per gene; each analysis took under 2 minutes on each gene sequence alignment, whether it was a single gene or a supergene. Specific results are:

- 11-taxon dataset strongILS and weakILS: less than 1 minute per gene
- 17-taxon dataset: less than 2 minutes per gene
- 11-taxon 50-gt, 10 bins (5 genes in each): less than 2 minutes per supergene
- 11-taxon 25-gt, 5 bins (5 genes in each): less than 2 minutes per supergene
- 11-taxon 100-gt, 20 bins (5 genes in each): less than 2 minutes per supergene

3 ADDITIONAL RESULTS

3.1 Experiment 1: Evaluating fast species tree estimation methods on 100 replicate datasets

CA-ML showed substantial improvements over the next best method (typically MP-EST, but in one case MRP) in Experiment 1 for the 11-taxon datasets, with biggest improvements on the 11-taxon weakILS datasets. CA-ML was also more accurate than the next best method on the 17-taxon datasets, but the differences were smaller. As can be seen, the improvements were statistically significant for all conditions, with $p < 0.003$ on the 11-taxon datasets (both strongILS and weakILS), and $p \leq 0.043$ on the 17-taxon datasets.

- 11-taxon strongILS 5-gt: (CA-ML vs. MRP): $p < 10^{-6}$
- 11-taxon strongILS 10-gt: (CA-ML vs. MP-EST): $p < 10^{-3}$
- 11-taxon strongILS 25-gt: (CA-ML vs. MP-EST): $p = 10^{-6}$
- 11-taxon strongILS 50-gt: (CA-ML vs. MP-EST): $p < 10^{-5}$
- 11-taxon strongILS 100-gt: (CA-ML vs. MP-EST): $p = 0.003$
- 17-taxon 8-gt: (CA-ML vs. MP-EST): $p = 0.013$
- 17-taxon 32-gt: (CA-ML vs. MP-EST): $p = 0.043$

Thus, the improvement of CA-ML over the next best method is statistically significant in all these cases.

3.2 Experiment 2: Evaluating species tree estimation methods on 20 replicate datasets

**BEAST vs. fast methods on RAxML gene trees:* We compared *BEAST to fast methods on RAxML gene trees on 20 replicates of all model conditions. With the exception of the 17-taxon 32-gene case, the differences were statistically significant. On 11-taxon strongILS datasets, *BEAST is significantly better than the fast methods ($p < 10^{-3}$). The difference is also significant on 17-taxon 8-gene datasets (p-values are within the range $0.02 \sim 0.03$). On 11-taxon weakILS datasets, *BEAST is significantly better than the fast methods on 5 and 10 genes ($p < 10^{-2}$), but not significantly better on 25 or 50 genes ($p > 0.1$).

*CA-ML vs. *BEAST:* As *BEAST is computationally intensive to run (tens to hundreds of hours for each analysis for some datasets), we compared CA-ML to *BEAST on only 20 replicate datasets of each model condition. The relative performance between the two methods was mixed, with CA-ML being more accurate in some cases and less accurate in others. However, the only statistically significant differences were for two conditions: 11-taxon 25-gene strongILS and 11-taxon 5-gene weakILS, in which CA-ML was more accurate than *BEAST ($p = 0.05$ and $p = 0.03$, respectively).

BUCKY-con vs. BUCKy-pop: The difference is statistically significant only on the 11-taxon strongILS 25-gene ($p = 0.003$) and 50-gene ($p = 0.035$) cases.

3.3 Experiment 3: Evaluating gene tree estimation error

Here we discuss the accuracy of gene trees estimated by maximum likelihood (by RAxML or FastTree-2) and *BEAST. Results for the 11-taxon strongILS conditions are provided in Figure 1 and Table 1; results for the 11-taxon weakILS conditions are provided in Figure 2 and Table 2. In Table 3 we present results for the 17-taxon datasets; the figure for these data are in the main document. Note that *BEAST gives a dramatic improvement in gene tree estimation accuracy, and that the smallest improvement is on the 17-taxon datasets. However, even on these data, the improvement is at least 50%.

Table 1. Average missing branch rates (over 20 replicates) of gene trees estimated by different methods on 11-taxon strongILS datasets. *BEAST could not be run on 100-gene datasets. Experiment 3.

Method	Error 5 genes	Error 10 genes	Error 25 genes	Error 50 genes	Error 100 genes
*BEAST	0.224	0.162	0.155	0.141	-
FastTree	0.430	0.440	0.407	0.418	0.424
RAxML	0.405	0.424	0.401	0.399	0.413

Table 2. Average missing branch rates (over 20 replicates) of gene trees estimated by different methods on 11-taxon weakILS datasets. Experiment 3.

Method	Error 5 genes	Error 10 genes	Error 25 genes	Error 50 genes
*BEAST	0.095	0.039	0.033	0.033
FastTree	0.314	0.299	0.338	0.334
RAxML	0.311	0.283	0.321	0.319

Table 3. Average missing branch rates over 20 replicates of gene trees estimated by different methods on 17-taxon datasets. Experiment 3.

Method	Error 8 genes	Error 32 genes
*BEAST	0.195	0.176
FastTree	0.399	0.400
RAxML	0.393	0.389

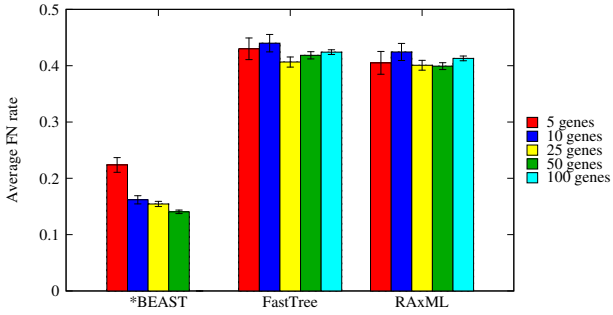

Fig. 1. Gene tree estimation error rates on 11-taxon strongILS datasets. Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. Experiment 3.

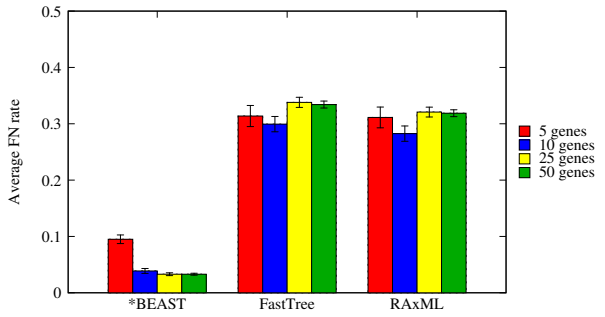

Fig. 2. Gene tree estimation error rates on 11-taxon weakILS datasets. Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. Experiment 3.

3.4 Experiment 4: Evaluating summary methods on gene trees estimated by *BEAST

The figures below show results of using summary methods on gene trees estimated using *BEAST, and compares them to the species trees estimated by *BEAST. There were no statistically significant differences in the accuracy of trees estimated using *BEAST as compared to using summary methods on gene trees estimated using *BEAST ($p > 0.2$ for all pairwise comparisons).

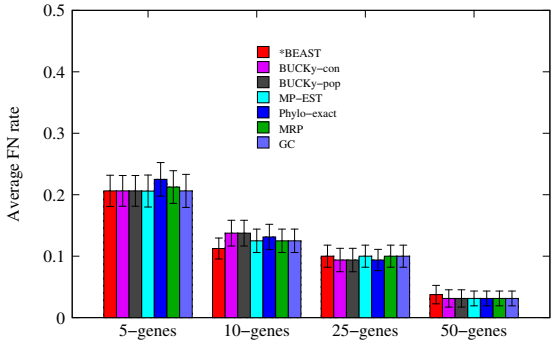

Fig. 3. Results for summary methods on gene trees estimated using *BEAST on 11-taxon weakILS model conditions with up to 50 genes; n=20 for each data point. Experiment 4.

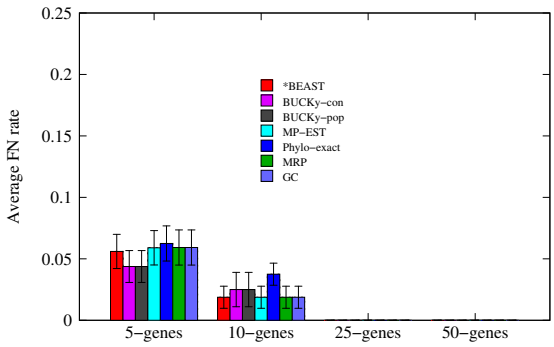

Fig. 4. Results for summary methods with input gene tree distributions estimated using *BEAST on 11-taxon weakILS model conditions with up to 50 genes; n=20 for each data point. Every method returns the true tree on the 25- and 50-gene datasets. Experiment 4.

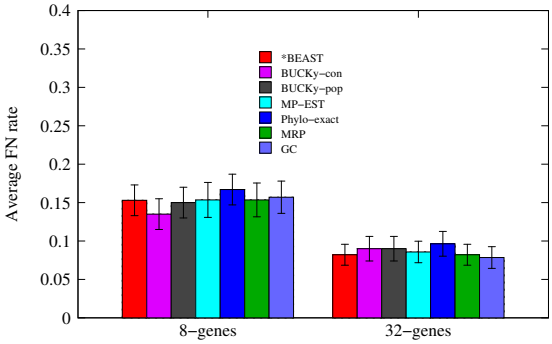

Fig. 5. Results for methods with input gene tree distributions estimated using *BEAST on 17-taxon model conditions; n=20 for each data point. Experiment 4.

3.5 Experiment 5: Evaluating the impact of naive binning on fast methods - 100 replicate datasets

We divide Experiment 5 into two parts: a comparison on 100 replicate datasets of the fast methods (all methods other than *BEAST and BUCKy), and then a comparison on 20 replicate datasets of all methods. See this subsection for results on fast methods, and the next subsection for results on all methods. Note that the impact of binning on the fast methods is best evaluated in the experiments on 100 replicate datasets, rather than on the 20 replicate datasets, especially in terms of statistical significance.

Because CA-ML is an unpartitioned analysis, it is not impacted by binning. Binning can impact all the other methods, but we do not have results for the unbinned Bayesian methods (*BEAST and BUCKy) on these 100 replicate datasets because they are too computationally expensive.

These experiments show the following trends:

- MP-EST, MRP, Phylonet, and Greedy Consensus each improved for all numbers of genes on the 11-taxon strongILS condition and on the 25-gene 11-taxon weakILS condition. The improvements on the 11-taxon weakILS conditions with 25 genes were small (at most 0.5%), but this is because all unbinned methods were highly accurate to begin with – all had error between 0.4% and 1.4%. The improvements on the 11-taxon strongILS conditions ranged from 1% to 4.8% (Phylonet on 50 genes), but differences were generally less on the 100-gene case (ranging from 0.6% to 3%) and 25-gene case (ranging from 1.1% for Greedy to 3% for Phylonet) than on the 50-gene case (ranging from 1.6% for MP-EST to 4.2% for Greedy).
- Phylonet became 0.5% more accurate on the 17-taxon condition, but the change was not statistically significant ($p > 0.25$). All other methods (MP-EST, Greedy, and MRP) became less accurate on the 17-taxon conditions, but the difference in accuracy was small (at most 1%) and the changes were not statistically significant for any of these methods.
- On the 11-taxon models, the differences for Phylonet's performance were statistically significant for every case, and tended to be larger than for the other methods. They were statistically significant for Greedy Consensus only on the 11-taxon strongILS datasets with 50 and 100 genes (and hence not for 25 genes on either strongILS or weakILS). The results were statistically significant for MP-EST on the 25-gene datasets (both strongILS and weakILS), but not for the other cases. Finally, the results were statistically significant for MRP only on the 50-gene strongILS datasets.

Thus, methods differed in their response to binning, and binning on the 11-taxon datasets generally improved accuracy and sometimes substantially, while generally reducing accuracy (but only slightly) on the 17-taxon datasets. However, the only statistically significant differences were improvements in accuracy. Phylonet in particular benefited from binning, improving even on the 17-taxon datasets, and improvement was greatest in cases where there were enough genes (at least 50), and accuracy before binning was not too great.

7.

Table 4. Average missing branch rates for methods (unbinned and binned) on 11-taxon strongILS 25, 50 and 100-gene cases; n = 100. Each bin contains 5 genes. BUCKy (unbinned) was not run on 100 replicates. Experiment 5.

Method	Error 25 genes	Error 50 genes	Error 100 genes
CA-ML	0.053	0.031	0.018
BUCKy-con (binned)	0.070	0.045	0.034
BUCKy-pop (binned)	0.070	0.045	0.034
MP-EST	0.110	0.073	0.039
MP-EST (binned)	0.088	0.057	0.033
Phylonet-exact	0.126	0.089	0.054
Phylonet-exact (binned)	0.096	0.041	0.024
MRP	0.115	0.091	0.050
MRP (binned)	0.105	0.053	0.038
GC	0.114	0.096	0.054
GC (binned)	0.103	0.054	0.034

Table 5. Evaluating the statistical significance of using binning on fast methods, when analyzing 100 replicate 11-taxon strongILS datasets. We show p-values for the statistical significance of a difference between binned and unbinned analyses. Each bin has 5 genes. Experiment 5.

Method	p-values 25 genes	p-values 50 genes	p-values 100 genes
MP-EST	0.021	0.057	0.211
Phylonet	0.002	$< 10^{-5}$	$< 10^{-3}$
MRP	0.177	$< 10^{-4}$	0.079
GC	0.156	$< 10^{-4}$	0.007

Table 6. Average FN rates for methods (unbinned and binned) on 11-taxon weakILS 25-gene case; n = 100. Each bin contains 5 genes. We did not run *BEAST or BUCKy on 100 replicates. Experiment 5.

Method	Error
CA-ML	0.000
MP-EST	0.014
MP-EST (binned)	0.003
Phylonet	0.008
Phylonet (binned)	0.000
MRP	0.008
MRP (binned)	0.004
GC	0.009
GC (binned)	0.004

Table 7. Evaluating the impact of binning on fast methods on 100 replicate 11-taxon weakILS datasets with 25 genes. We show p-values for the statistical significance of a difference between binned and unbinned analyses. Each bin has 5 genes. Experiment 5.

Method	p-values
MP-EST	0.002
Phylonet	0.016
MRP	0.188
GC	0.109

Table 8. Average FN rates for methods (unbinned and binned) on 17-taxon 32-gene case; n = 100. Each bin contains 4 genes. We did not run unbinned BUCKy on 100 replicates. Experiment 5.

Method	Error
CA-ML	0.136
BUCKy-con (binned)	0.154
BUCKy-pop (binned)	0.154
MP-EST	0.149
MP-EST (binned)	0.159
Phylonet	0.176
Phylonet (binned)	0.171
MRP	0.146
MRP (binned)	0.153
GC	0.151
GC (binned)	0.161

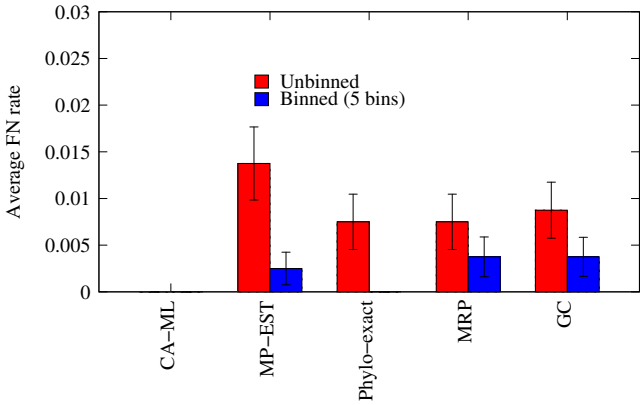

Fig. 7. Results of the binning experiment on 11-taxon 25-gene weakILS datasets. Each bin contains 5 genes. Average and standard error bars shown; n=100 for all datapoints. CA-ML returns the true tree on these data Experiment 5.

Table 9. Evaluating the impact of binning for fast methods (binned vs. unbinned) on 100 replicates of 17-taxon 32-gene dataset. We show p-values for the statistical significance of binned versus unbinned analyses. Each bin has 4 genes. Experiment 5.

Method	p-values
MP-EST	0.221
Phylonet	0.258
MRP	0.273
GC	0.245

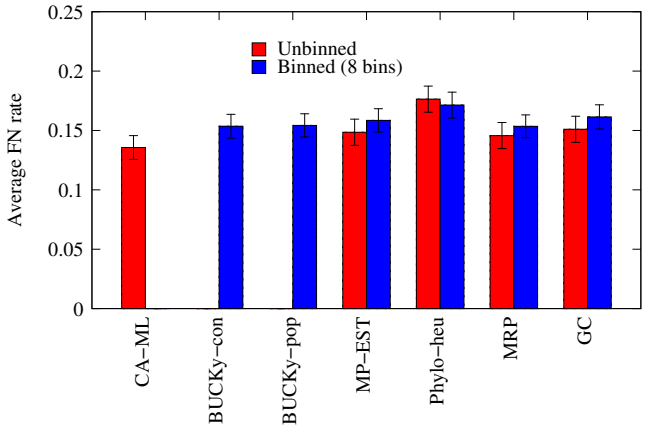

Fig. 6. Results of binning experiment on 17-taxon datasets with 32 genes. We show the performance (average and standard error bars) of methods other than BUCKy on unbinned genes and *BEAST. Each bin contains 4 genes; n=100 for all datapoints. Experiment 5.

3.6 Experiment 5: Evaluating the impact of naive binning on all methods - 20 replicate datasets

We now show results for naive binning on all methods (including BUCKy and *BEAST), but restricted to 20 replicate datasets. On these datasets, we were able to run the Bayesian methods (BUCKy and *BEAST), and so can explore the impact of binning on these methods. We do not show results for unbinned *BEAST on the 100-gene datasets, because these were too computationally intensive to run, but do show results obtained using *BEAST with binned datasets.

These results show the following trends:

- *BEAST has unchanged accuracy under all conditions where it can run in the unbinned and binned settings.
- On the 17-taxon datasets, no changes were statistically significant.
- BUCKy-con improved for the 11-taxon strongILS datasets (ranging from 3% on the 100-gene case to 7.5% on the 50-gene case) and by 2.5% on the 11-taxon weakILS 25-gene case. The changes were statistically significant for 25-genes and 50-genes, but not for 100-genes, on the strongILS datasets.
- With the exception of Phylonet (which was 100% accurate both with and without binning) all methods improved on the 11-taxon weakILS datasets as a result of binning, and the improvements ranged from 0.7% (for MRP) to 3.1% (for BUCKy-pop). However, only BUCKy-pop had a statistically significant improvement ($p = 0.031$).

These results are similar to those observed on the 100-replicate case, except that with only 20 replicates, we do not detect statistically significant changes.

Table 10. Average FN rates for methods (unbinned and binned) on 11-taxon strongILS 25, 50 and 100-gene cases; n = 20. We do not show results for unbinned *BEAST on 100 genes, because it was not run to convergence. Each bin contains 5 genes. Experiment 5.

Method	Error 25 genes	Error 50 genes	Error 100 genes
CA-ML	0.062	0.025	0
*BEAST	0.100	0.038	-
*BEAST (binned)	0.100	0.038	0.012
BUCKy-con	0.143	0.125	0.056
BUCKy-con (binned)	0.094	0.050	0.025
BUCKy-pop	0.088	0.088	0.056
BUCKy-pop (binned)	0.094	0.050	0.025
MP-EST	0.156	0.163	0.044
MP-EST (binned)	0.106	0.056	0.031
Phylonet-exact	0.106	0.094	0.025
Phylonet-exact (binned)	0.077	0.069	0.018
MRP	0.143	0.163	0.056
MRP (binned)	0.138	0.056	0.043
GC	0.150	0.160	0.063
GC (binned)	0.125	0.056	0.044

Table 11. Evaluating the impact of binning on all methods, applied to 20 replicates of the 11-taxon strongILS datasets. We show p-values. We were not able to run *BEAST (unbinned) on 100-gene datasets. Experiment 5.

Method	p-values for 25 genes	p-values for 50 genes	p-values for 100 genes
*BEAST	0.500	0.500	-
BUCKy-con	0.018	0.005	0.089
BUCKy-pop	0.441	0.227	0.062
MP-EST	0.011	$< 10^{-4}$	0.363
Phylonet	0.113	0.179	0.500
MRP	0.307	$< 10^{-3}$	0.291
GC	0.230	$< 10^{-4}$	0.290

Table 12. Average FN rates for methods (unbinned and binned) on 17-taxon 32-gene case; n = 20. Each bin contains 4 genes. Experiment 5.

Method	Error
CA-ML	0.100
*BEAST	0.082
*BEAST (binned)	0.082
BUCKy-con	0.107
BUCKy-con (binned)	0.111
BUCKy-pop	0.119
BUCKy-pop (binned)	0.114
MP-EST	0.114
MP-EST (binned)	0.125
Phylonet	0.139
Phylonet (binned)	0.132
MRP	0.104
MRP (binned)	0.114
GC	0.104
GC (binned)	0.121

Table 13. Evaluating the impact of binning on species tree estimation methods on 20 replicates of the 11-taxon weakILS datasets with 25 genes. We show p-values for methods (binned vs. unbinned methods). Each bin has 5 genes. Experiment 5.

Method	p-values
BUCKy-con	0.063
BUCKy-pop	0.031
MP-EST	0.250
Phylonet	0.500
MRP	0.500
GC	0.250

Table 14. Average FN rates for methods (unbinned and binned) on 11-taxon weakILS 25-gene case; n = 20. Each bin contains 5 genes. Experiment 5.

Method	Error
CA-ML	0.000
*BEAST	0.000
*BEAST (binned)	0.000
BUCKY-con	0.025
BUCKy-con (binned)	0.000
BUCKy-pop	0.031
BUCKy-pop (binned)	0.000
MP-EST	0.019
MP-EST (binned)	0.006
Phylonet	0.000
Phylonet (binned)	0.000
MRP	0.013
MRP (binned)	0.006
GC	0.019
GC (binned)	0.006

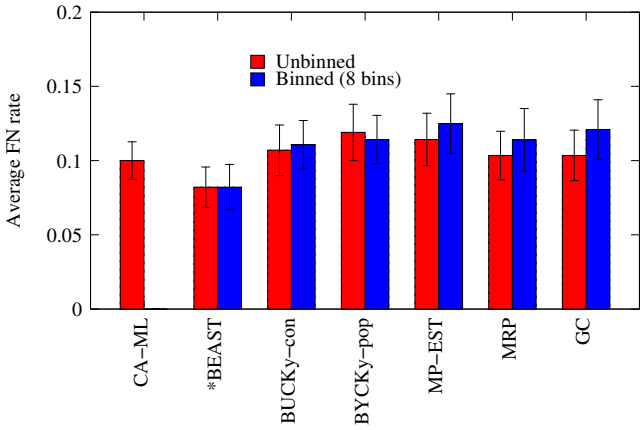

Fig. 9. Results of binning experiment of 17-taxon datasets with 32 genes. Average and standard error bars shown for all methods. Each bin has 4 genes; n=20 for all datapoints. No changes are statistically significant ($p = 0.053$ for MRP, $p = 0.082$ for GC, and $p > 0.2$ for all other methods). Experiment 5.

Table 15. p-values for methods (binned vs. unbinned) on 20 replicates of 17-taxon 32-gene dataset. Each bin has 4 genes. Experiment 5.

Method	p-values
*BEAST	0.500
BUCKy-con	0.444
BUCKy-pop	0.311
MP-EST	0.191
Phylonet	0.212
MRP	0.053
GC	0.082

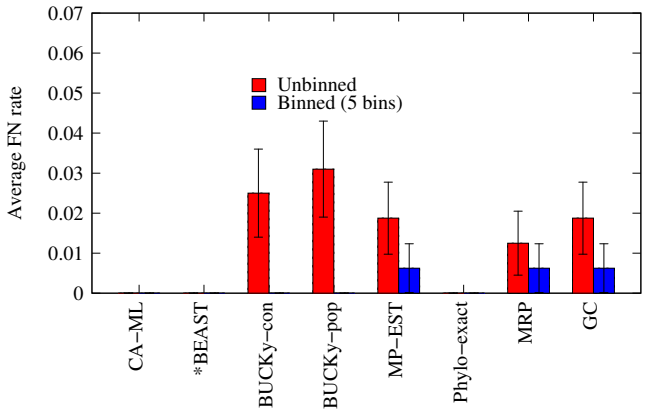

Fig. 8. Results of the binning experiment evaluating all methods on 20 replicates of the 11-taxon 25-gene weakILS datasets. Results are shown (average and standard error bars) for bins with 5 genes each. CA-ML, *BEAST (binned and unbinned), BUCKy-con (binned), BUCKy-pop (binned), and Phylonet-MDC (binned and unbinned) all return the true tree on these data.

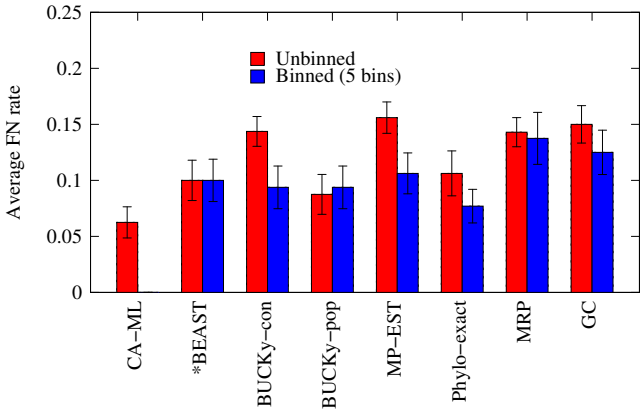

Fig. 10. Results of the binning experiment on 11-taxon 25-gene strongILS datasets. Each bin contains 5 genes. Average and standard error bars shown; n=20 for all datapoints. Experiment 5.

4 ADDITIONAL DISCUSSION

4.1 Previous studies comparing concatenation to coalescent-based estimation of species trees

One of the interesting results in this paper is that concatenation using maximum likelihood produced better results than the summary coalescent-based methods, and was often more accurate than *BEAST. Since this result seems to run counter to the literature about coalescent-based methods, we discuss this in some detail.

While many papers have used simulations to evaluate coalescent-based methods, most of these papers only compared coalescent-based methods to each other, rather than to concatenation. Thus, to the best of our knowledge, only Larget *et al.* (2010); Liu *et al.* (2010); Edwards *et al.* (2007); DeGiorgio and Degnan (2010); Kubatko and Degnan (2007); Leaché and Rannala (2011); Heled and Drummond (2010) present results of simulation studies that compare concatenated analysis (either based on a Bayesian or a maximum likelihood method) to coalescent-based methods. We discuss each of these in turn.

DeGiorgio and Degnan (2010): This study introduces Supermatrix Rooted Triplets (SMRT), a coalescent-based method that is statistically consistent under ILS when sequences evolve under the two-state CFN molecular clock model. They compare SMRT to maximum likelihood in an extensive simulation study with model trees having at most 6 taxa (most have only 4 or 5 taxa). Almost all of the simulations were performed under a strong molecular clock. In their simulations, concatenation was generally, but not always, outperformed by SMRT. However, the relative performance was clearly impacted by the amount of ILS (as determined by parameter settings), with concatenation performing as well (or better) when ILS was very low. The relative performance was also impacted by the number of genes, so that under some models where SMRT outperformed concatenation for large numbers of genes, concatenation outperformed SMRT for small numbers of genes. They also explored the impact of violating the molecular clock in the simulation, but inferring under the clock; this study showed that concatenation was less impacted by the model violation than SMRT.

The most interesting part of this analysis is that it showed that the relative performance of concatenation using maximum likelihood and SMRT depended on several conditions, including whether sequences evolved under a strong molecular clock, the amount of ILS, and the number of genes.

Leaché and Rannala (2011): This paper reports on a very extensive comparison several coalescent-based methods (STEM, BUCKy, and BEST) to two concatenation methods (one using MrBayes and one using maximum parsimony implemented in PAUP*) on 5-taxon model species trees. Sequence evolution on each gene was under Jukes-Cantor with a strong molecular clock, and produced sequences of length 1000 bp. They also report the percentage of time the true tree is returned by the given analysis.

One focus of their study was evaluating the impact of the the model tree topology (balanced vs. unbalanced) on the relative performance of methods; they observed that BEST generally had the highest accuracy on the asymmetric model species trees and BUCKy generally had the best accuracy on the symmetric model species trees. There were, however, some model conditions (reflecting the amount of ILS) in which MrBayes was either first or tied for first, and many conditions in which MrBayes was only slightly less accurate than BEST and BUCKy.

Larget et al. *(2010):* This paper presents a comparison of concatenated analysis using a consensus tree output by MrBayes (Huelsenbeck and Ronquist, 2001) to the BUCKy-pop and BUCKy-con trees, on three model conditions with rooted species trees and 5 taxa. Every model species tree has the strong molecular clock, and sequences with 500 bp evolve under the Jukes-Cantor model. They report only the percentage of times that each method recovers the true tree exactly. Two of the three models are in the anomaly zone, and one of these is in the "too greedy" zone. The analysis shows that BUCKy-pop generally had the best results of all three methods. Results on the easiest of the three model conditions show all methods had roughly the same accuracy (though BUCKy-pop does better at 10 and 30 genes than the other methods), and all methods converged to the true species tree at 100 genes. Results on the two trees in the anomaly zone distinctly show the improvement of BUCKy-pop over the other methods.

Liu et al. *(2010):* This paper presents the MP-EST method, and reports results for several simulation studies in which MP-EST is compared to other coalescent-based method. However, they also provide a simulation study comparing MP-EST and concatenation. The model tree here is a 5-taxon species tree in the anomaly zone, and sequences of length 500 evolve under the Jukes-Cantor model with the strong molecular clock. They report the frequency of returning the correct tree. Their study suggests that the two methods have roughly the same accuracy at the smallest number of genes they studied (100), but that MP-EST converges to the correct tree at 2500 genes, while Bayesian analysis (MrBayes) converges to the wrong tree at 500 genes.

Edwards et al. *(2007):* This paper introduced the coalescent-based method BEST, which co-estimates gene trees and species trees. They provide a simulation study comparing BEST to MrBayes from 30 genes that evolve within an 8-taxon model species tree. Sequence evolution on these genes is under the Jukes-Cantor model and a strong molecular clock and had 500 bp. For this analysis, they report that the species tree had 98% of the posterior probability under the BEST analysis, but that MrBayes converged to the wrong tree as the number of genes increased.

Heled and Drummond (2010): This paper introduced *BEAST, a method for co-estimating gene trees and species trees. They compared *BEAST to BEST (another coalescent-based co-estimation method) and also to BEAST, a Bayesian concatenation method for estimating species trees. They performed a simulation study using 7-taxon species trees with 4 genes that evolved under the Jukes-Cantor model and a strong molecular clock. The sequence alignments each had 1600 bp. They evaluated performance with respect to the how often the true species tree appeared in the 95% credible set of tree topologies. They observed that *BEAST had the best results, with BEST not too far below - but that BEAST had by far the worst accuracy.

Discussion: These studies clearly indicate that coalescent-based methods can be more likely to produce the true species tree than concatenation under some circumstances. However, all these studies shared some features: small numbers of taxa, generally large numbers of genes, and all genes evolving under a strong molecular

clock. Some of these studies also primarily focused on model species trees in the anomaly zone. These features are likely to make it easier for coalescent-based methods (possibly especially ones that combine estimated gene trees) to perform better than concatenation-based methods that do not take ILS into account. For example, DeGiorgio and Degnan (2010) observed that the presence of a strong molecular clock favors SMRT, a coalescent-based method that assumes the molecular clock; since many other coalescent-based methods assume the strong molecular clock, this would suggest that simulations under a strong molecular clock may be biased in favor of the coalescent-based methods. Also, summary methods (i.e., methods that combine estimated gene trees) are impacted by the accuracy of the estimated gene trees, and the simulation conditions in these studies may have all had sufficient sequence length and rates of evolution (relative to the number of taxa) to provide fairly accurate gene trees. Finally, most of these papers (though not all!) focused on accuracy on large numbers of genes, and the results in DeGiorgio and Degnan (2010) show that the relative accuracy concatenation and coalescent-based methods can change with the number of genes (with concatenation sometimes being as good or better on small numbers of genes, but coalescent-based methods being better than concatenation on larger numbers of genes).

Taken as a whole, these studies do show that coalescent-based methods can be more accurate than concatenation. However, these studies primarily explored performance only for very small numbers of taxa, large numbers of genes, high amounts of ILS, and a strong molecular clock, while also demonstrating that these model conditions can impact the relative accuracy of concatenation and coalescent-based methods. Like these studies, our study focuses on performance under high amounts of ILS (the 11-taxon strongILS and 17-taxon conditions both have high amounts of ILS), and we also use sequences that evolved under the Jukes-Cantor model. However, there are several key difference between these studies and our study. First, we explore performance on small numbers of genes (at most 100) rather than on large numbers of genes. Second, our conditions produce estimated gene trees that are generally not that accurate as a result of inadequate sequence length, and we conjecture that the other studies had more accurate gene trees than our study. Third, the 11-taxon model conditions do not evolve sequences under a strong molecular clock. Fourth, we use 11-taxon and 17-taxon datasets instead of smaller datasets.

These differences may be sufficient to explain the different conclusions between this study and the others, but additional research will be needed to understand the impact of these model conditions on the relative accuracy of concatenation and coalescent-based estimation. Finally, we note that the performance criterion used in our study is different from that used in these other studies; they explored the percentage of the datasets in which the true species tree was recovered by each method, while we reported the average False Negative (missing branch) rate. While these criteria are equal for very small trees (4-taxon unrooted trees or 3-taxon rooted trees), they are not identical for larger trees, and it is possible that relative performance between two methods could change depending on the choice of criterion.

4.2 Limitations on Binning

One of the findings of this study is that naive binning is helpful for coalescent-based methods. However, the conditions in which we explored the use of naive binning were either cases where concatenation was more accurate than binning (the 11-taxon datasets with not too many genes) or where the difference between concatenation and coalescent-based methods was very small (the 17-taxon datasets, and the 11-taxon datasets with sufficiently many genes so that all methods recovered the true tree). Therefore, it is possible that the naive binning technique we used is helping only because it creates a hybrid method that falls somewhere between concatenation and coalescent-based estimation, and therefore has accuracy that falls between these two.

In other words – does this naive binning technique help because it brings the coalescent-based method closer to concatenation, or does it help for some other reasons as well (such as addressing the vulnerability to poor signal gene trees)? Understanding the reasons that naive binning helps, and the conditions under which it helps, requires additional study.

4.3 Closing comments

We close with a basic question about phylogenetic estimation, suggested by this study. Given that summary methods are impacted by error in the estimated gene trees (resulting from inadequate phylogenetic signal in the sequence alignments), what is the optimal binning strategy? More generally, what is the best trade-off between data quantity (number of estimated gene trees) and quality (accuracy of estimated gene trees) for summary methods? Understanding the trade-off between data quantity and quality for each summary method will help inform binning strategies (e.g., how to pick the size of the bins), even if these strategies are statistically-based. This topic is subtle and statistically complex, and is only beginning to be studied, but see Huang *et al.* (2010) for further discussion.

REFERENCES

Ané, C., Larget, B., Baum, D. A., Smith, S. D., and Rokas, A. (2007). Bayesian estimation of concordance among gene trees. *Mol Biol Evol*, **24**, 412–426.

Bayzid, M. S. and Warnow, T. (2012). Estimating optimal species trees from incomplete gene trees under deep coalescence. *J. Comput. Biol.*, **19(6)**, 591–605.

Chung, Y. and Ané, C. (2011). Comparing two Bayesian methods for gene tree/species tree reconstruction: A simulation with incomplete lineage sorting and horizontal gene transfer. *Syst Biol*, **60**(3), 261–275.

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

Edwards, S. V., Liu, L., and Pearl, D. K. (2007). High-resolution species trees without concatenation. *Proceedings of the National Academy of Sciences*, **104**(14), 5936–5941.

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

Huang, H., He, Q., Kubatko, L., and 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.

Huelsenbeck, J. and Ronquist, R. (2001). MrBayes: Bayesian inference of phylogeny. *Bioinformatics*, **17**, 754–755.

Kubatko, L. S. and Degnan, J. H. (2007). Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Syst Biol*, **56**, 17.

Larget, B., Kotha, S. K., Dewey, C. N., and Ané, C. (2010). BUCKy: Gene tree/species tree reconciliation with the Bayesian concordance analysis. *Bioinf*, **26**(22), 2910–2911.

Leaché, A. D. and Rannala, B. (2011). The accuracy of species tree estimation under simulation: a comparison of methods. *Syst Biol*, **60**(2), 126–137.

Liu, L., Yu, L., and Edwards, S. V. (2010). A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC Evolutinary Biology*, **10:302**.

Maddison, W. P. and Maddison, D. R. (2011). Mesquite: a modular system for evolutionary analysis. Website. `http://mesquiteproject.org/mesquite/mesquite.html`.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2010). FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS ONE*, **5**(3), e9490.

Ragan, M. A. (1992). Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.*, **1**, 53–58.

Rambaut, A. and Grassly, N. C. (1997). Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 235–238.

Stamatakis, A. (2006). RAxML-NI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinf*, **22**, 2688–2690.

Swofford, D. (1996). *PAUP*: Phylogenetic analysis using parsimony (and other methods), version 4.0.* Sinauer Assoc., Sunderland, Mass.

Than, C. V., Ruths, D., and Nakhleh, L. (2008). PhyloNet: A software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinf*, **9**, 322.

Yu, Y., Warnow, T., and Nakhleh, L. (2011a). Algorithms for MDC-based multi-locus phylogeny inference. In *Proc RECOMB 2011*.

Yu, Y., Warnow, T., and Nakhleh, L. (2011b). Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles. *J Comp Biol*, **18**(11), 1543–1559.