4. Species Tree Estimation Using Site Pattern Frequencies

David L. Swofford and Laura S. Kubatko

1 Introduction

Methods for estimating species trees can be divided into several classes, as described in Chapter 1. The advantages and disadvantages of each class of methods are widely appreciated. For example, summary methods are known to be computationally efficient, often requiring only seconds or minutes of computation once gene trees have been estimated for each of the individual loci. However, these methods fail to account for uncertainty in the estimated gene trees, and thus their performance is dependent on the quality of the gene tree estimates used as input. On the other hand, Bayesian co-estimation methods are based on probabilistic models linking the observed sequence data directly to the species tree and associated parameters, typically by explicitly incorporating the multispecies coalescent (MSC) model. These have the advantage of being fully model-based, thus enabling estimation of associated model parameters. However, due to the complexity of the underlying models, co-estimation methods typically rely on Markov chain Monte Carlo (MCMC) methods for statistical inference, and thus require extensive, and sometimes prohibitive, computation in order to carry out inference.

An alternative to summary methods and MCMC-based co-estimation methods are methods that utilize site pattern frequencies as input to directly estimate the species tree under the MSC. In this chapter, we describe two such methods. The first, SVDQuartets, is a method for estimation of the species tree topology that does not rely on likelihood computation for estimation, thus enabling coalescent-based inference to be carried out on large-scale genomic data in a computationally efficient manner. We describe the theory underlying SVDQuartets, give some details concerning the algorithms used for estimation, and provide examples of its implementation in PAUP*. The second method we describe is designed to estimate parameters associated with a fixed species tree topology, such as the speciation times, using a composite likelihood approach. We describe how the composite likelihood is computed, as well as its use in a Bayesian context to derive statistically consistent estimators of the speciation times. Both procedures are implemented in the PAUP* software, allowing the analyses to be carried out with user-friendly software. We provide recommendations for use of these methods, and carefully describe their strengths and weaknesses. Finally, we use the genome-scale data on gibbon species analyzed by Carbone et al. (2014) and Shi and Yang (2018) to demonstrate the performance of the methods.

2 Estimation of the species tree topology using SVDQuartets

2.1 Theoretical basis

As its name implies, SVDQuartets is a method for species tree inference that is based on the examination of quartet relationships. Thus, we begin by describing the model that SVDQuartets assumes for the relationships among quartets (collections of four taxa) under the coalescent model. Consider four species numbered 1 through 4, and note that there are three possible unrooted phylogenies relating these four species, as shown in Figure 1. To derive the theory underlying SVDQuartets, we make the following assumptions. First, we assume that our data consist of a collection of aligned sites from the genomes of the four species under consideration. We assume that, conditional on the species tree, each site evolves independently of the other sites in the data set. Thus, each site has its own underlying gene genealogy that arises under the MSC model on

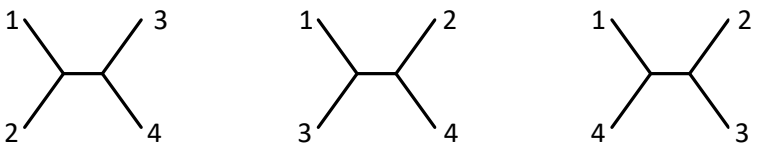


Figure 1: The three possible quartet trees for four taxa.

the species tree. We call this data type *coalescent independent sites (CIS)*. Though we argue later that SVDQuartets can be properly applied to linked sites sharing the same underlying gene tree, the theory underlying the method is motivated by the setting of CIS data and we focus on that data type in this initial description.

Next, we assume that sequence data arise along these gene trees according to one of a set of commonly-used nucleotide substitution models that includes the GTR+I+G model or any submodel thereof. We further assume that no gene flow occurs following speciation, and that no processes other than the coalescent contribute to variation among gene trees, i.e., we assume that no horizontal transfer, gene duplication and loss, or gene conversion has occurred. The model underlying SVDQuartets *does* allow for species trees that violate the molecular clock, and it allows for variation in the effective population sizes along the branches of the species tree.

Given species 1 through 4, we can consider the probability of observing a particular configuration of nucleotides at the tips of the tree. Let X_H denote the nucleotide observed at a particular site for species H, and let p_{ijkl} be the probability that species 1 has nucleotide i, species 2 has nucleotide j, species 3 has nucleotide k, and species 4 has nucleotide l, i.e.,

$$p_{ijkl} = P(X_1 = i, X_2 = j, X_3 = k, X_4 = l). \tag{1}$$

We call $ijkl$ a site pattern, and we refer to the collection of probabilities $\{p_{ijkl} : i, j, k, l \in \{A, C, G, T\}\}$ as the site pattern probability distribution. For any particular unrooted quartet tree, we can arrange these $4^4 = 256$ site pattern probabilities in the form of a 16×16 matrix, called a *flattening matrix*, for which the rows correspond to possible nucleotides for two of the species that form a cherry in the tree, and the columns correspond to possible nucleotides for the other two species. As an example, the flattening matrix below corresponds to the tree in Figure 1(a):

$$Flat_{12|34} = \begin{pmatrix} & [AA] & [AC] & [AG] & [AT] & [CA] & \cdots & [TT] \\ [AA] & p_{AAAA} & p_{AAAC} & p_{AAAG} & p_{AAAT} & p_{AACA} & \cdots & p_{AATT} \\ [AC] & p_{ACAA} & p_{ACAC} & p_{ACAG} & p_{ACAT} & p_{ACCA} & \cdots & p_{ACTT} \\ [AG] & p_{AGAA} & p_{AGAC} & p_{AGAG} & p_{AGAT} & p_{AGCA} & \cdots & p_{AGTT} \\ [AT] & p_{ATAA} & p_{ATAC} & p_{ATAG} & p_{ATAT} & p_{ATCA} & \cdots & p_{ATTT} \\ [CA] & p_{CAAA} & p_{CAAC} & p_{CAAG} & p_{CAAT} & p_{CACA} & \cdots & p_{CATT} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ [TT] & p_{TTAA} & p_{TTAC} & p_{TTAG} & p_{TTAT} & p_{TTCA} & \cdots & p_{TTTT} \end{pmatrix}$$

In the above matrix, the $(3, 2)$ entry, p_{AGAC}, is the probability of observing nucleotide A for species 1, nucleotide G for species 2, nucleotide A for species 3, and nucleotide C for species 4, for example.

Chifman and Kubatko (2015) showed that when the flattening matrix corresponds to the true species tree (i.e., when the species used for the rows and columns of the matrix reflect a true split in the species tree), then the rank of the flattening matrix is 10 when the molecular clock holds for the GTR+I+G model and all submodels. When the flattening matrix does not correspond to a split in the true tree, then the rank of this matrix is 16. Long and Kubatko (2019) extended

this result to the case in which the molecular clock is no longer required and for which effective population sizes are allowed to vary along the tree under the GTR model and all submodels. These results are important in two ways. First, they establish that the species tree is identifiable from sequence data under the MSC, which is necessary to prove consistency of likelihood-based inference methods (see [27]). Second, they provide the theoretical basis for the SVDQuartets method, which we now describe.

Given an empirical data set consisting of either a collection of SNPs or of alignments for multiple loci, we use the observed site pattern frequencies in the data to estimate the flattening matrices corresponding to each of the three unrooted four-taxon trees in Figure 1. For each of these matrices, we then compute a measure, which we call the SVDScore, that gives the distance to the nearest rank 10 matrix. As a specific example, consider again the tree in Figure 1(a) and suppose that the $Flat_{12|34}$ matrix has been estimated by substituting the observed frequency of each site pattern for the true probabilities. For this matrix, we then carry out singular value decomposition. Letting $\hat{\sigma}_j$ represent the j^{th} singular value, we define the SVDScore as

$$SVDScore := \sqrt{\sum_{i=11}^{16} \hat{\sigma}_i^2}, \tag{2}$$

i.e., the SVDScore is the square root of the sum of squares of the 11^{th} through 16^{th} singular values. The Eckhart-Young Theorem [10] establishes that this is the distance to the nearest rank 10 matrix under the Frobenius norm. When the flattening matrix corresponds to the true quartet relationships found in the species tree, the true values of the 11^{th} to the 16^{th} singular values are 0, and thus their estimated values are expected to be small. What this means in practical terms is that small values of the SVDScore indicate that the quartet relationship under consideration is likely to be that found on the true species tree, while larger values indicate lack of support for that particular quartet relationship on the true species tree.

Example: Rank reductions for flattening matrices

To make the ideas introduced in this section more concrete, we provide a specific example that provides some intuition for the matrix rank results discussed above. To simplify the problem, we consider the gene tree (rather than the species tree) setting, and we suppose that there are only two possible nucleotides that could be observed at the tips of the tree, A and G. We assume that mutations between these two nucleotides occur at equal rates and that the nucleotides are equally frequent. Under this model, there are four distinct classes of site patterns as depicted in Figure 2, such that all patterns in the same class have the same probability of occurring on the tree under this model. The flattening matrix that results is shown in the bottom right panel of Figure 2. It is easy to see that the middle two columns and middle two rows of this matrix are identical. Thus, although the flattening matrix has four rows and four columns (and thus its rank could be as large as four), one of those columns is redundant, leading to a reduction in the rank of the matrix by at least one.

We now consider a numerical version of this example to show that the rank is reduced even further. In Figure 3, the tree considered in this example is assigned branch lengths, which are then used to compute the probabilities of each of the 16 possible site patterns, as shown in the left column. These site patterns can be arranged into the flattening matrix, as demonstrated in Figure 2, and then the redundant third row and third column can be removed. It can additionally be noted that the probabilities given in the last column can be obtained from those in the first two columns (and similarly for the rows, though this is not shown in the figure), and so the rank of the matrix is again reduced by one. Thus, although the maximum possible rank of the flattening

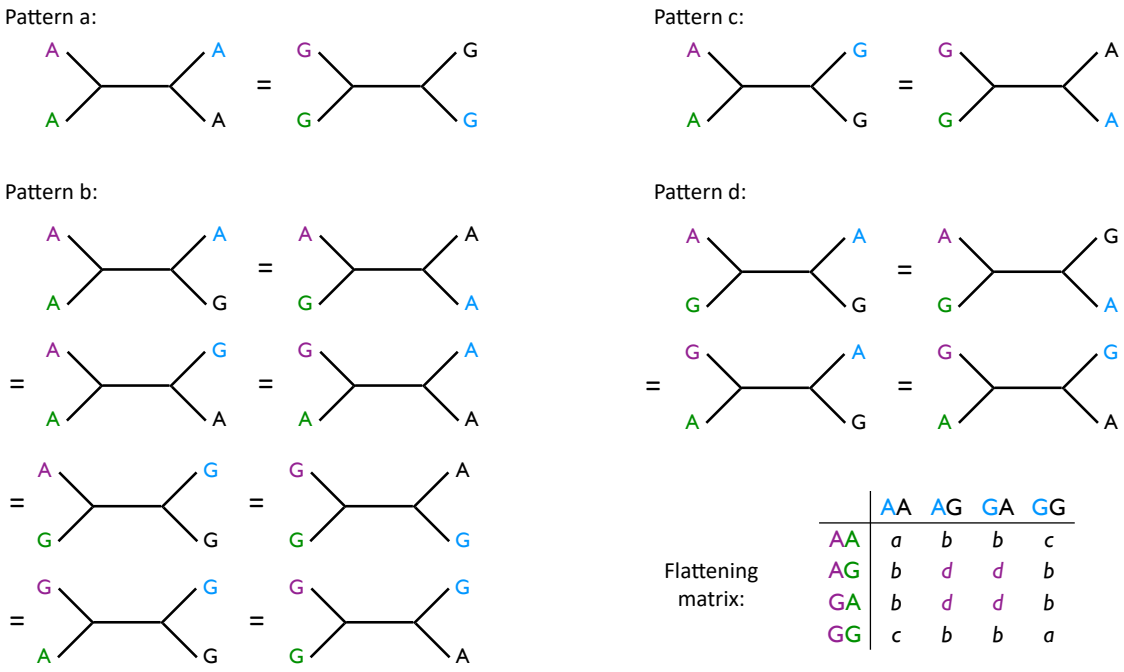
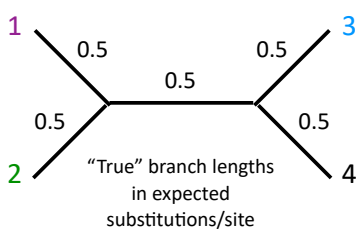


Figure 2: Possible site patterns under the simplified model considered in our example, for which only nucleotides A and G are possible. In each tree shown, colors indicate species (purple = species 1, green = species 2, black = species 3, blue = species 4) and nucleotides at the tips of tree indicate a possible site pattern. For this simplified model, all possible site patterns can be reduced to four classes in such a way that patterns within a class have the same probability. For example, "Pattern a" (top left) includes the case where all species have either nucleotide A or nucleotide G. These two patterns are equally probable under the model assumed here. The flattening matrix corresponding to this tree is shown in the bottom right panel. Note that the middle two columns and middle two rows in the flattening matrix are identical to one another.

matrix is four, the actual rank is two, because the entire matrix can be reproduced once the first two columns of the matrix are known.

Thus far, we have provided some intuition for why the matrix rank is reduced for flattening matrices that correspond to the true tree. However, the question about whether similar rank reductions occur for the two alternative flattening matrices has not been addressed. To examine this, consider the tree that groups species 1 and 3 and species 2 and 4 together (see Figure 4). Note that the flattening matrix for this tree can be obtained by rearranging the entries of the matrix for the original tree, as shown in part (b) of Figure 4. With these rearrangements, no columns (or rows) are duplicates of one another, and it can be shown that none are linear combinations of the others, either. Thus, the rank of the flattening matrix for this alternative tree is 4.

This is a significant result, and one that forms the basis of the theory underlying SVDQuartets: flattening matrices corresponding to the true tree show reductions in their ranks, while flattening matrices corresponding to incorrect trees (i.e., trees that did not give rise to the observed data) do not. Though our example here was simplified to the case of gene trees for a model with only two observed nucleotides and equal mutation rates, the principles hold directly for the more complicated case of data arising under the MSC. The only changes are that the full set of nucleotides are considered, leading to 256 site patterns and flattening matrices that are 16×16. For a quartet that reflects the true species-level relationships, the rank of the corresponding flattening matrix is



Expected site-pattern frequencies

p_{AAAA}	0.09300841
p_{AAAG}	0.06135527
p_{AAGA}	0.06135527
p_{AAGG}	0.06811487
p_{AGAA}	0.06135527
p_{AGAG}	0.04672782
p_{AGGA}	0.04672782
p_{AGGG}	0.06135527
p_{GAAA}	0.06135527
p_{GAAG}	0.04672782
p_{GAGA}	0.04672782
p_{GAGG}	0.06135527
p_{GGAA}	0.06811487
p_{GGAG}	0.06135527
p_{GGGA}	0.06135527
p_{GGGG}	0.09300841

Expected flattening matrix for 1,2|3,4

	AA	AG	GA	GG
AA	0.093008	0.061355	0.061355	0.068115
AG	0.061355	0.046728	0.046728	0.061355
GA	0.061355	0.046728	0.046728	0.061355
GG	0.068115	0.061355	0.061355	0.093008

Delete redundant 3rd row and column...

	AA	AG	GG
AA	0.093008	0.061355	0.068115
AG	0.061355	0.046728	0.061355
GG	0.068115	0.061355	0.093008

Note that we can now obtain the last column of the above matrix as a linear combination of the first two columns:

$$f_{AA,GG} = -f_{AA,AA} + 2.62617\, f_{AA,AG} = 0.068115$$
$$f_{AG,GG} = -f_{AG,AA} + 2.62617\, f_{AG,AG} = 0.061355$$
$$f_{GG,GG} = -f_{GG,AA} + 2.62617\, f_{GG,AG} = 0.093008$$

$\therefore$ ***matrix* has *only two linearly independent rows and columns; rank is 2***

Figure 3: Continuation of the example in Figure 2 to consider a specific numerical example. The example tree with branch lengths shown in the top left can be used to compute the 16 possible site pattern probabilities listed in the left column. Arranging these in the flattening matrix and removing the redundant third column shows that the last column can be expressed as a linear combination of the first two, leading to a further reductions in the rank of the flattening matrix to rank 2.

10 (rather than 2, as in our example), while the rank for an incorrect tree is 16 (rather than 4, as in our example). These results have been proven mathematically, and hold for any substitution model that fits the assumptions of the GTR+I+G model when the molecular-clock assumption is satisfied. Even if the clock assumption is violated, the result still holds for all submodels of GTR that assume equal rates among sites. For GTR submodels including invariable sites or other mixtures of rates, the rank is still reduced for the true tree, but by a lesser amount.

One reason that this result is so important is that calculating the rank of a small matrix is trivial in terms of computation time, and can be completed in fractions of a second. In practice, however, the flattening matrix must first be estimated from data, and the resulting numerical calculation of the matrix will always give a rank of 16, even for the true tree. This is due to the fact that sampling error in the estimates of the entries of the flattening matrix will result in the middle two columns in our example being *similar*, but not *identical*. This is why the SVDScore in Equation 2 is used; it allows us to measure how similar a matrix is to being of a certain rank, even when it is not precisely the desired rank. This score has well-developed mathematical theory behind it, and has been used in other phylogenetic contexts, as well [1, 2]. It can also be rapidly computed, and has been shown to be useful in differentiating between trees [7, 11, 15].

(a) Flattening matrix for 1,2|3,4

	AA	AG	GA	GG
AA	*a*	*b*	*b*	*c*
AG	*b*	*d*	*d*	*b*
GA	*b*	*d*	*d*	*b*
GG	*c*	*b*	*b*	*a*

(b) Flattening matrix for 1,3|2,4

	AA	AG	GA	GG
AA	*a*	*b*	*b*	*d*
AG	*b*	*c*	*d*	*b*
GA	*b*	*d*	*c*	*b*
GG	*d*	*b*	*b*	*a*

Figure 4: Flattening matrices for the original example tree (a) and for the tree in which species 1 and 3 and species 2 and 4 are grouped together (b). Colors refer to species labels as defined in Figure 2 and matrix entries refer to the site pattern probabilities from Figure 2. Note that the matrix in (b) has no duplicate rows or columns.

2.2 Accounting for incomplete lineage sorting in SVDQuartets

In the example above, gene trees were used to gain intuition about how the reduction in matrix rank arises for the flattening matrix corresponding to the true tree. In the full SVDQuartets method, however, the reduction in matrix rank is achieved for a species tree under a data model based on the multispecies coalescent. In this section, we provide a short, non-technical description of how the coalescent process is incorporated into SVDQuartets to accommodate incomplete lineage sorting (ILS; see Chapter 1 for a description of ILS and how it is related to gene trees).

To illustrate the model, consider a species tree with three taxa, as shown in the left panel of Figure 5. Recall from Chapter 1 that the coalescent process can be used to specify the probability of each of the four possible gene tree histories that can arise from this species tree. These are shown in the middle panel of Figure 5 with their probabilities labeled on the arrows that lead from the species tree. The coalescent model also specifies a probability distribution on the lengths of the branches within these gene tree histories. These probabilities distributions are not shown here, as they don't provide any insight into the model (the interested reader can consult [13, 19]). Along each gene tree history, sequence data arise in the same way as described in the gene tree example above to generate site patterns (third panel in Figure 5). When a single site pattern is assumed to have arisen along each gene tree, a CIS data set is generated. When multiple site patterns arise from a fixed gene tree and data from many such gene trees are observed, the data are referred to as multilocus data.

The previous paragraph described the mechanism by which data are generated, but how is this data generation mechanism incorporated into SVDQuartets? The key idea is that the entries in the theoretical flattening matrix (i.e., the one that contains the true site pattern probabilities rather than those estimated from the data) are computed under the model in Figure 5. For example, consider the probability associated with the observed site pattern GTT. We can see from Figure 5 that there are several "paths" by which we could observe this site pattern: it could have arisen from any of the four gene tree histories in the second panel of the figure. Thus, to compute its probability, we need to sum over the probability that it was generated from each of the gene trees histories, weighting by the probability that the particular gene tree history was generated by the species tree. Branch lengths are handled similarly, except that we must integrate over the branch length distribution because the branch lengths are continuous. Chifman and Kubatko (2015) used the Mathematica software to compute these integrals for the two possible rooted four-taxon species trees. They obtained analytical expressions that could then be used to prove mathematically that the rank of the flattening matrix is reduced for the true tree under the coalescent model. Thus, although the data used for SVDQuartets are often presented as a concatenated data matrix, it is

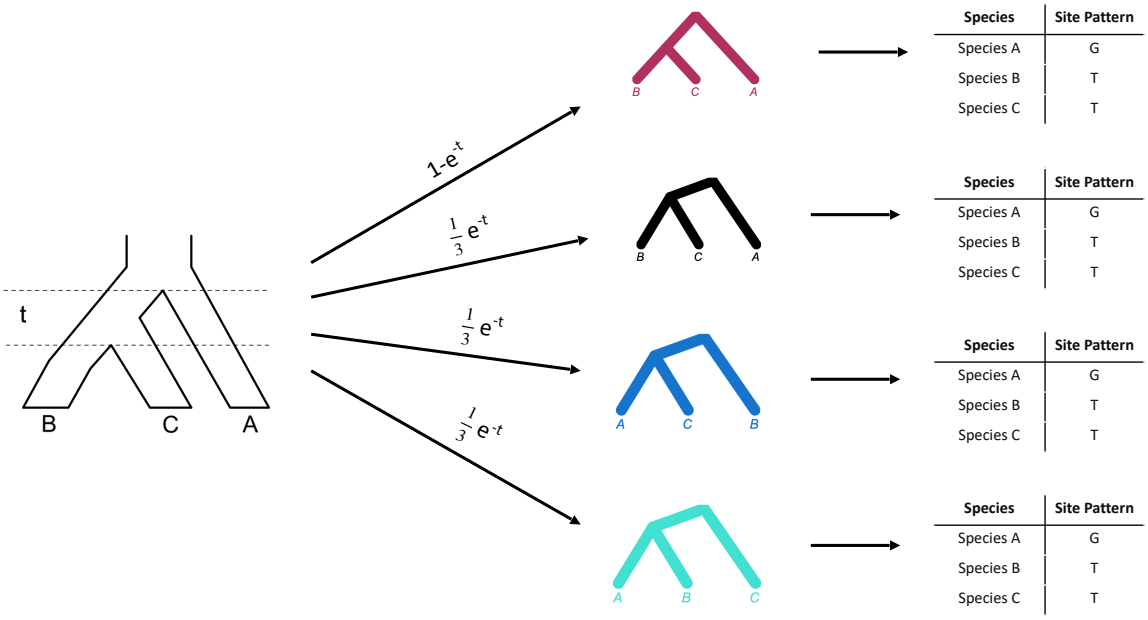


Figure 5: The data generation process for a species-level phylogeny with three species. The species tree is shown in the leftmost panel. The length of the branch between speciation events is t. The second panel shows the four possible gene tree histories that are compatible with the species trees, along with their probabilities (labeled on the arrows). Note that the first two gene tree histories share the same topology but differ in the timing of the coalescent event uniting species B and C (see Chapter 1 for details). The third panel shows that site pattern GTT can arise from any of the four histories.

important to remember that SVDQuartets does formally model ILS arising from the multispecies coalescent.

2.3 Species tree inference: quartet sampling and assembly

In the previous section, we described the analysis used by SVDQuartets for trees with only four taxa. We now describe the method used by SVDQuartets as implemented in PAUP* (`http://paup.phylosolutions.com`) for inferring the species tree. To begin, the data must be input into PAUP* in Nexus format. Optionally, the Nexus file also contains a `taxpartition` command that can be used to map sequences to species in the case in which multiple individuals are sampled for one or more of the species under consideration. The analysis is run using the `svdq` command in PAUP*.

By default, the software will consider all possible combinations of four sequences, each selected from a distinct species, if this number is not too large. If this number is too large, then samples of four sequences may be randomly selected. PAUP* allows this step to be parallelized via multi-threading, as each quartet can be evaluated independently of the others. SVDQuartets can easily evaluate all possible quartets for up to 100 taxa, and this number can be pushed to 200 or more taxa depending on the speed of the computer, the number of processor cores available, and the length of time the user is willing to wait. Unfortunately, there is not a straightforward answer to the question of how many random quartets should be sampled for problems that are too large for exhaustive quartet sampling. In general, the more the better, but the number needed to obtain an accurate estimate of the species tree topology will depend on several data-specific characteristics, such as the informativeness of the inferred quartet trees and the extent to which they conflict with

one another. We recommend that users experiment with increasingly large numbers of sampled quartets and assess stability in the inferring species tree across these runs.

For each quartet, three flattening matrices, one corresponding to each of the unrooted four-taxon trees in Figure 1, are estimated, and the SVDScore defined in Equation 2 is computed for each. The three scores are compared, and the tree that produces the smallest score is retained as that inferred for the quartet under consideration. The result after considering all quartets (or a random sample of quartets) is a list of quartet trees that is then used as input to a quartet assembly algorithm. More information on the process of quartet assembly is given in Section 2.4 below. The result of the assembly process is an estimate of the unrooted species tree topology. The procedure used by SVDQuartets is shown in Figure 6.

2.4 Algorithmic details

A complication that arises when calculating the SVDScore (Equation 2) involves the numerical accuracy and stability of the singular value decomposition. A number of algorithms exist for computing the SVD of a matrix, and care must be taken to select an appropriate one; e.g., accuracy of SVD algorithms is tied, in part, to minute details of floating-point arithmetic characteristics on the processor being used. Our usage in SVDQuartets differs from many applications of SVD in that accurate estimation of the smallest entries in the vector of singular values is important.

We use routines from the highly regarded LAPACK Fortran library [3], which are the only ones we evaluated that do not crash or exhibit other numerical problems for some inputs. The primary method used for singular value decomposition in PAUP* is the DGESDD routine from LAPACK (which is also used by R, MATLAB, and other numerical software). However, we have discovered that the singular values returned by this method can be inaccurate for sparse flattening matrices containing many zero entries (i.e., when many of the 256 possible site patterns for 4 taxa are not observed). Very small singular values can occur in this case, and sometimes values that should be exactly zero are instead returned as small positive numbers. The numerical inaccuracy can be great enough to cause the wrong quartet topology to be preferred, or one of the topologies to be chosen as best when in fact all three topologies should have equal scores.

Several modifications to the code have been made to improve accuracy of computing the singular values of the flattening matrix under these conditions. First, rows and columns containing all-zero entries are removed prior to performing the SVD (which does not affect the estimated rank of the matrix). In addition to improving the accuracy of DGESDD, this matrix reduction sometimes makes SVD calculation entirely unnecessary, as the rank of the matrix cannot be greater than min(number of rows, number of columns). Second, when all three of the quartet topologies have very similar SVDScores, an additional SVD is performed using the slower, but more accurate, LAPACK routine DGEJSV; the singular values returned by DGEJSV are then used to recalculate the scores for the quartet. Third, if the scores for all three topologies for a quartet are equal (within a tolerance for floating-point roundoff error), the quartet is simply discarded. In earlier versions of PAUP*, one of the three resolutions was kept, and the decision as to which one was chosen was impacted by the numerical inaccuracy described above.

A second critical issue is the method used to assemble the results from individual quartets into an overall tree. This assembly requires solving the NP-complete Maximum Quartet Consistency (MQC) problem [12]. Many heuristics have been proposed to approximate solutions to MQC. We tested several of them, and found the Quartets MaxCut (QMC) [24] and Quartet FM (QFM) [20] to outperform the alternatives, and chose to implement QFM for PAUP*. We developed a novel implementation of QFM based on the algorithmic descriptions in Reaz et al. (2014) [20] that runs much faster than their original implementation. We also modified certain aspects of the algorithm

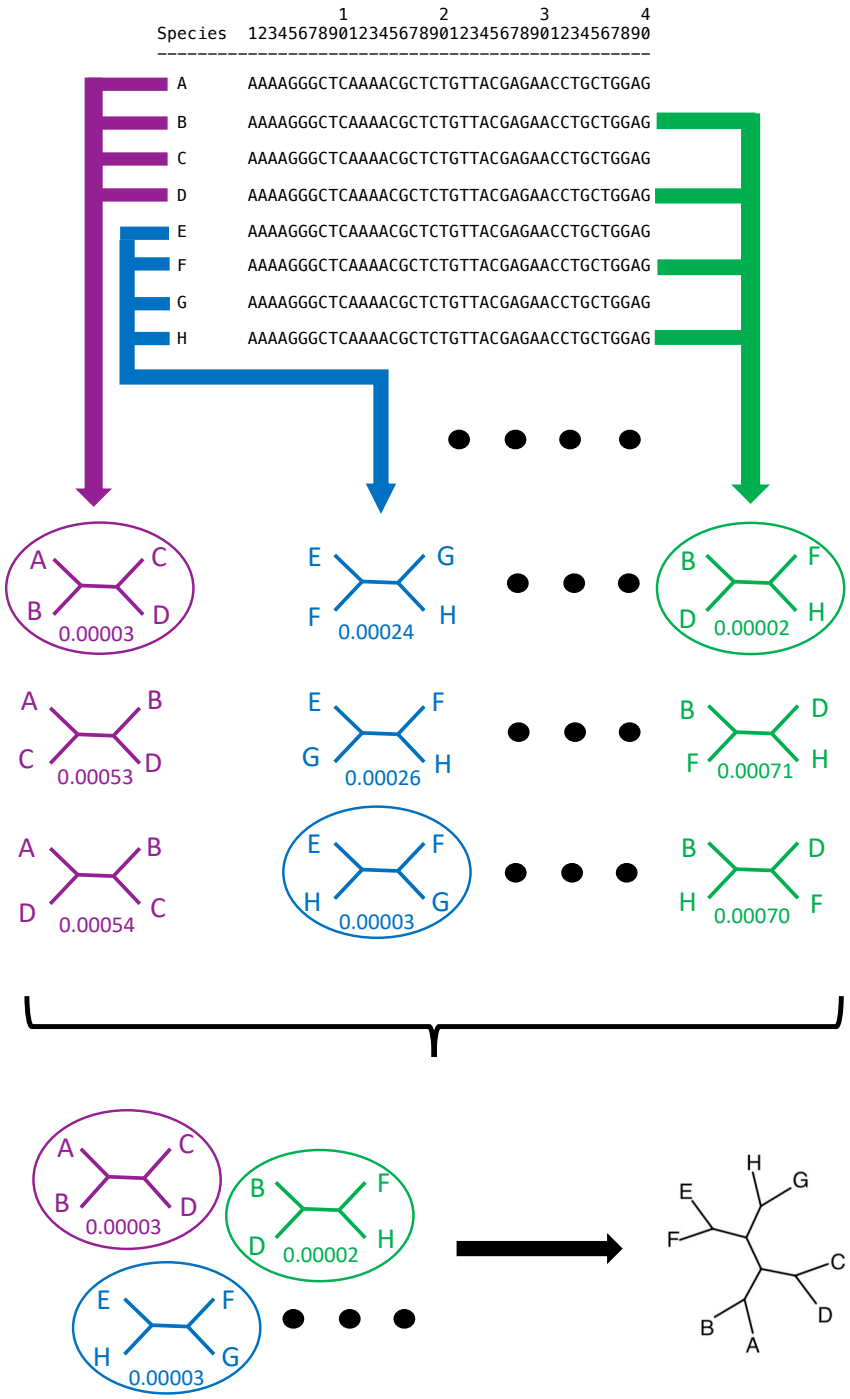


Figure 6: Schematic of the estimation procedure used by SVDQuartets. The input data set consists of eight species, labeled A through H. The first step is to consider quartets of species and compute the SVDScore for each of the three possible unrooted topologies for these quartets. The figure shows three example quartets, indicated by arrows of different colors. For example, the purple arrow indicates selection of species A, B, C, and D, and of the three possible unrooted trees for this quartet, the one that places species A and B together (circled in purple) has the lowest SVDScore. This tree is then retained and used as input for the quartet assembly step (bottom of the figure). A similar process is shown for quartets consisting of species E, F, G, and H (blue arrow) and B, D, F, and H (green arrows). The black dots indicate that all possible quartet samples, including those not shown here, would need to be evaluated in this way. The bottom of the figure indicates that the inferred quartet trees from the first step are used as input for a quartet assembly procedure that then produces the estimate of the species tree (bottom right).

that improved its effectiveness (to be published elsewhere).

2.5 Uncertainty quantification

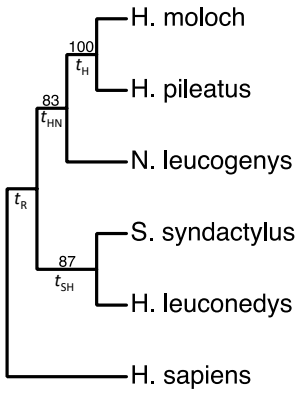
Thus far, we have described the process by which SVDQuartets can be used to estimate the species tree topology. When obtaining a phylogenetic estimate at any level, it is also important to obtain a measure of uncertainty in the phylogenetic estimate. For SVDQuartets, the most natural measure of uncertainty is obtained by bootstrapping, which is implemented in PAUP* in two different forms. If the input data to be used in SVDQuartets are SNPs, then the proper bootstrap procedure is to sample sites at random with replacement from the input aligned SNP data. For each bootstrap replicate, SVDQuartets is used to estimate a species tree as described in the previous section, and these bootstrap species trees are summarized with a consensus tree. The bootstrap support for partitions not appearing in the consensus tree is also reported, and the species trees estimated for each bootstrap replicate can optionally be written to an output file.

The procedure is similar in the case of multilocus data, with the exception that the process by which the bootstrap samples are drawn uses the method of Seo (2008) [22]. To illustrate the process in this case, consider a data set that consists of M loci with n_i sites for locus i, $i = 1, 2, \ldots, M$. To obtain a single bootstrap sample in this case, M loci are selected at random with replacement from the set of loci in the sample. For each sampled locus, say j, a sample of n_j sites is selected at random with replacement from the original data for that locus. These data then form a bootstrap data set from the original data. This process is repeated to obtain the desired number of bootstrap data sets, and the analysis and summary of the bootstrap data then proceeds as described above in the case of SNP data.

2.6 Application to species relationships among gibbons

As an example, we'll apply SVDQuartets to a data set consisting of five species of gibbons [6, 23]. The data set for coding regions used in [23] consists of 11,323 genes for a total of 2,264,600 sites for six species: *Hylobates moloch* (1 individual sampled); *H. pileatus* (1 individual sampled); *Symphalangus syndactylus* (2 individuals sampled); *Hoolock leuconedys* (2 individuals sampled); *Nomascus leucogenys* (2 individuals sampled); and *Homo sapiens* (1 sequence was used as the outgroup). Note that each individual contributes two sequences to the data set, and so the overall data set consists of 17 sequences.

SVDQuartets was run by sampling all possible quartets, with uncertainty measured using 100 bootstrap replicates in the multilocus bootstrap procedure. The entire analysis took 8.23 seconds on a desktop machine, and the estimated tree is shown in Figure 7. Shi and Yang (2018) analyzed these data using the MCMC-based method BPP, and found that, across 10 independent runs, two distinct trees were identified as the maximum a posteriori (MAP) tree (one was found in seven of the 10 runs, with the other found by the remaining three). Interestingly, the two trees found by BPP differ from the tree inferred by SVDQuartets in the placement of *N. leucogenys*, with it being placed sister to the clade containing *S. syndactylus* and *Ho. leuconedys* in the majority of the runs and sister to the entire group in the remaining runs. The somewhat low bootstrap support for the clade containing this species (i.e., bootstrap support of 83) indicates that the data are not strongly informative about placement of this species, which is reflected in the BPP analysis. Further analysis of these data (see Section 3.4 below) indicates very short intervals between speciation events in the evolutionary history of this species.



Branch	Estimate	Standard Error	95% HPD Interval
t_H	1.8892	0.0412	(1.8140, 1.9822)
t_{HN}	2.5×10^{-10}	4.86×10^{-9}	(0, 1.19×10^{-8})
t_{SH}	0.0822	0.0195	(0.0419, 0.1193)
t_R	10.60	0.2616	(10.11, 11.12)

Figure 7: Species tree estimated by SVDQuartets (left) and MAP_{CL} estimates of branch lengths in coalescent units (right) for the gibbon data of Carbone et al. (2014). Numbers above the internal nodes are bootstrap support values, and notation below the branches indicates the label of the branch used to report the estimated length in the table at the right. Note that SVDQuartets estimates an unrooted tree; the tree displayed here has been rooted using the human sequence as an outgroup.

2.7 Properties of SVDQuartets

In this section, we review the assumptions underlying SVDQuartets and describe its properties. In doing so, we describe scenarios under which SVDQuartets should and should not be applied.

2.7.1 Assumptions and misconceptions

SVDQuartets was introduced above for CIS data, i.e., data for which each site is generated from its own underlying gene tree which in turn has been generated from the species tree via the MSC, and the theoretical derivations described above assume this data structure. It is also directly applicable to SNP data sets, because constant site patterns (which are included in CIS data but would not be included in SNP data) do not impact the reduced rank results on which the method is based (e.g., in our example above, the two middle columns in the flattening matrix that were equal to one another did not involve any of the constant site patterns).

SVDQuartets can also be used to analyze multilocus data. To gain some intuition for why this works, recall again that for CIS data, each site is an independent observation arising from the species tree under the MSC. The method then works by using observed site pattern frequencies to estimate theoretical site pattern probabilities derived under the model. What changes when moving to multilocus data is that now each gene tree provides information about the distribution of site patterns under the MSC by contributing many *correlated* sites, i.e., sites within a given locus are not independent because they come from the same underlying gene tree. In one sense, this is an advantage: we get more information about each gene tree. But on the other hand, this is not quite as good as having an equivalent number of *independent* sites, as these will carry more information that is directly relevant about the species tree. Nonetheless, having increased information about each locus is helpful in obtaining the overall phylogenetic estimate. When the number of loci is large, so that we have sufficient data to capture the variation in the underlying gene trees, then the multilocus estimate will perform very well, and better, in fact, than an estimate obtained from using only one SNP from the same number of loci. For this reason, we recommend that multilocus data be used when they have been collected. For example, we do not suggest that RADseq data be filtered to provide only one SNP per locus; rather, all sequence data collected should be used in

the analysis.

The description above highlights a key requirement for SVDQuartets to perform well: the amount of data must be sufficient to enable accurate estimation of the flattening matrix corresponding to the true tree, so that the SVDScore will be able to "detect" the reduction in matrix rank. The number of sites needed depends to some extent on the underlying species tree, particularly on the probability with which sites that are informative about the reduced rank are generated. It also depends on whether CIS or multilocus data are used. Application of SVDQuartets to data collected for only a handful of loci is not likely to result in accurate species tree estimates, even if all of the loci are quite long, because such do not provide adequate information about variation among gene trees. On the other hand, several thousand CIS may be sufficient for some species trees, since each site is directly informative about the species tree. In both cases, however, it should be noted that trees with short internal nodes may require large sample sizes, as short species tree branches result in shallow gene trees which in turn produce data with a relatively low proportion of variable sites. Though this situation is challenging for all inference methods, it is particularly difficult for SVDQuartets when the number of sites is small.

A strength of SVDQuartets is that it is valid for a wide range of substitution models, although the model need not be specified explicitly. In addition, there is no requirement to estimate model parameters or to compute likelihoods. This has led some researchers to consider this a nonparametric or "model-free" method, but this is not accurate: the theory on which SVDQuartets is based assumes both the MSC and a standard substitution model, and features of the probability distributions arising from these models are precisely the information that is used for inference. Another misconception about SVDQuartets is that it is a "concatenation" method, an idea that appears to have originated from the fact that the input format for the analysis is a concatenated data matrix for which gene boundaries need not be specified. However, it is important not to confuse the data format with the underlying model. Recall that SVDQuartets assumes a model for which each site has its own underlying gene tree. Thus, the reason that gene boundaries need not be provided is because each site is assumed to be its own gene, independently sampled from the species tree. This is in direct contrast to concatenation methods that assume that all sites have a single underlying gene tree.

Finally, SVDQuartets has been referred to as a summary method for species tree inference [23] (p. 172) because the data used as inputs are counts of the 256 possible site patterns for each quartet. However, SVDQuartets is distinct from summary methods that estimate gene trees for each locus as a first step and subsequently use these estimated gene trees as input to estimate the species tree. Further, we do not view SVDQuartets as a summary method in any sense, as counts of the observed site patterns are a valid presentation of the input data, in the same way that a multiple sequence alignment (MSA) is a valid presentation of the input data. In fact, an MSA could also be viewed as a summary, in the sense that the raw read count data obtained in the sequencing reaction have been summarized to produce a single sequence to represent each taxon.

2.7.2 Statistical consistency

One property of a phylogenetic estimator that is often used to compare methods is that of statistical consistency. A statistically consistent estimator is one that becomes increasingly likely to be equal to the truth as the amount of data increases. Recent work [27] has shown that SVDQuartets provides a statistically consistent method for estimating the species tree. This result builds on two key ideas. First, entries of the flattening matrix are consistently estimated, leading to consistent estimation of the SVDScore. Consistent estimation of the SVDScore ensures that the quartet trees used as input to the quartet assembly algorithm are all correct when the data set size is sufficiently

large. Second, it is well-known that quartet relationships uniquely identify a tree [21], and thus an input data set consisting of true quartet relationships will result in the correct species tree estimate. Importantly, this consistency result holds for **both** multilocus and CIS data, and thus SVDQuartets is an accurate method of species tree estimation for large data sets, whether multilocus or CIS data are used. A detailed simulation study is included in [27], including comparisons with the Bayesian method BPP, for four-taxon problems.

2.7.3 Missing data and data filtering

Phylogenomic data commonly contain a very high proportion of missing data. SVDQuartets has two options for handling missing data. The first option is to ignore the missing data. If this option is selected, then the complete data set is considered for each set of four taxa that are sampled. If there are sites for which some data are missing for one of the four taxa under consideration, then those sites are not used in computing the SVDScore for that quartet. In this way, each quartet evaluation uses the maximum amount of non-missing data possible. The second option is to impute the missing data in the following way. Each site in the sequences contributes a count of 1 to the relevant site pattern frequency estimate, so if data are missing for one or more sites, then this count could be distributed over possible site patterns. For example, consider a site at which taxa A, B, and C have nucleotide T, but data at this site for taxon D is missing. Since the missing nucleotide could be either A, C, G, or T, we increment the counts of the patterns $TTTA$, $TTTC$, $TTTG$, and $TTTT$ each by some fractional amount. This amount could be chosen based on the observed proportion of each nucleotide in the input data, or based on some assumptions about the relative proportions of nucleotides. For example, if all nucleotides are assumed to be equally frequent, then each of the four site patterns listed above could be given a count of 0.25. In this way, missing data is accommodated by using the information available at this site (i.e., taxa A, B, and C have nucleotide T), while incorporating the information about the uncertainty in the nucleotide at that site for taxon D.

Large phylogenomic data sets are often fairly informative about the species tree, even when the proportion of missing data is large. In other words, what often matters is how much data you *have*, rather than how much data you *don't* have. Large data sets will necessarily often have a high percentage of missing data, yet still lead to relatively certain estimates for many relationships in the species tree. For this reason, we do not recommend that users filter their data prior to analysis with SVDQuartets. Subjective filtering can lead to biases in the retained sites, and the computational efficiency of SVDQuartets, coupled with its intuitive approach for handling missing data, make its application to all sampled data the recommended approach.

2.8 Recommendations for using SVDQuartets

When selecting a method for analyzing data with the goal of estimating the species tree, it is important to consider the characteristics of the data in relation to the requirements of possible inference methods. At the core of the SVDQuartets method is the need to estimate the site pattern probabilities accurately for each of the quartet trees, so the primary requirement in terms of data is that the data are sufficient to obtain reliable estimates of these probabilities. In general, this means that the performance of the method will improve as the amount of data increases. It also means that when the overall data set size is small (either in the number of loci or the number of sites or both), the method may not perform as well. We suggest that users exercise extreme caution when applying the method to fewer than 20 loci or fewer than 5,000 SNPs. On the other hand, a strong advantage of SVDQuartets is that it is computationally efficient for very large data sets. We have

successfully applied the method for millions of sites (see, e.g., the gibbon data analysis in Section 2.6) and hundreds of taxa.

SVDQuartets should always be run in conjunction with an appropriate bootstrapping approach (i.e., either the standard or the multilocus bootstrap). It is also important to note that SVDQuartets returns an unrooted estimate of the species tree, and hence rooting with an outgroup or some other appropriate method must be applied if a rooted estimate of the species tree is desired. An advantage of SVDQuartets is that it is not necessary to specify a substitution model, and the method remains valid if there is variation in the substitution model across the input sites when the molecular clock holds, provided that the models governing all partitions are submodels of GTR+I+G. In practice, settings in which there is extensive heterogeneity in the models that apply to different data partitions may require large sample sizes for accurate estimation of the species tree. SVDQuartets can also be applied in settings where the molecular clock is violated, and when there is variation in effective population sizes throughout the tree [16]. SVDQuartets is also robust to the presence of gene flow between sister taxa [17].

3 Estimation of speciation times

In this section, we describe a method that uses site pattern frequencies to estimate speciation times for a fixed species phylogeny. The method uses the likelihood for four-taxon trees to construct a composite likelihood for species trees of arbitrary size. Maximization of this composite likelihood leads to estimators that are statistically consistent and asymptotically efficient, and that can be computed efficiently in practice. In the sections below, we describe the theoretical basis for computing these estimators, methods for quantifying uncertainty in the estimates, and recommendations for their use. We note that although the estimates are based on quartets and use site pattern frequencies as input data, they are distinct from the SVDQuartets method described in the previous section. In particular, they can be applied to a species tree topology that has been estimated by **any** method.

3.1 Theoretical basis

As mentioned in the previous section, there are 256 possible site patterns that can arise on a four-taxon species tree. Chifman and Kubatko (2015) showed that for the rooted symmetric and asymmetric species trees, these 256 site patterns can be reduced to 9 and 11 classes, respectively, in such a way that site patterns within each class occur with equal probability under the MSC with the JC69 substitution model. For example, it is clear that for the JC69 model, $p_{AAAA} = p_{CCCC} = p_{GGGG} = p_{TTTT}$ for any tree, while for the tree in Figure 1(a), $p_{AAGG} = p_{AACC} = \ldots = p_{GGTT}$. In addition to defining these classes for both symmetric and asymmetric trees, [8] derived analytic expressions for the probability of a site pattern in each class. These expressions can be used to compute the likelihood for site pattern frequency data on a fixed species phylogeny using the multinomial distribution, as described below.

We consider the symmetric species tree shown in Figure 8 in defining the relevant notation; the asymmetric case is analogous. This species tree includes three speciation times, labeled τ_1, τ_2, and τ_3. For this tree, the nine distinct classes of site patterns can be specified by

$$p_{xxxx}, p_{xxxy} = p_{xxyx}, p_{xyxx} = p_{yxxx}, p_{xyxy} = p_{yxxy}, p_{xxyy},$$

$$p_{xyxz} = p_{yxxz} = p_{xyzx} = p_{yxzx}, p_{xxyz}, p_{yzxx}, p_{xyzw}. \tag{3}$$

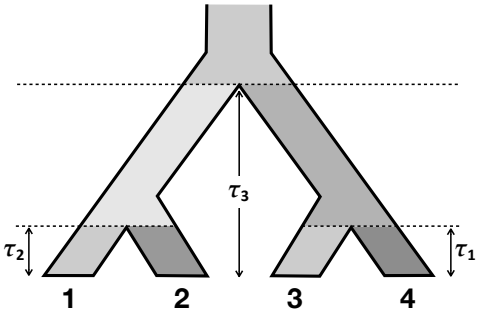


Figure 8: Example symmetric species tree with speciation times τ_1, τ_2, and τ_3.

where x, y, z, and w are states $\in \{A, C, G, T\}$ observed in each of four distinct taxa. For example p_{xxxx} includes the patterns $p_{AAAA} = p_{CCCC} = p_{GGGG} = p_{TTTT}$. We label the classes above 1-9, and denote the vector of 9 probabilities corresponding to these classes by $\mathbf{p} = (p_1, p_2, \ldots, p_9)$. We note that each of these probabilities is a function of both the speciation times and the effective population size, which is assumed to be constant throughout the tree and is denoted here by θ. The precise expressions for these probabilities are given in [8], and we refer the reader there for further details.

Let $\mathbf{Y} = (y_1, y_2, \ldots, y_9)$ denote the number of site patterns observed in each of the classes. Then for CIS data, the likelihood of the speciation times and the effective population size is given by

$$L(\tau_1, \tau_2, \tau_3, \theta|\mathbf{Y}) \propto \prod_{j=1}^{9} p_j^{y_j} \,. \tag{4}$$

We note again that the p_j are functions of the τ's and θ, though we have suppressed that notation here for ease of exposition. This likelihood is used to form the basis of our inference procedure. For example, this likelihood (or equivalently, its logarithm) can be maximized to obtain maximum likelihood estimates (MLEs) of the parameters in the case of a fixed four-taxon species tree.

While this procedure is straightforward in the four-taxon case, expressions analogous to the p_is cannot be efficiently computed for five or more taxa, prohibiting the use of this likelihood framework directly for species trees with more than four tips. Therefore, as an alternative to computing the true likelihood, we use instead the *composite likelihood* for estimation of the parameters for species trees of arbitrary size. In order to form the composite likelihood, we first decompose the set of taxa into all possible quartets. We then compute the four-taxon likelihood given by Equation 4 for each of the quartets. Finally, we combine the four-taxon likelihoods by taking their product. Let Q be the number of unique quartets that can be obtained by sampling one lineage from each of four distinct species. Any quartet i induces a subtree on the full tree containing three internal nodes u_i, v_i, and w_i corresponding to nodes 1–3 in Equation 4. This subtree may be either symmetric or asymmetric, with either $n(i) = 9$ or $n(i) = 11$ distinct site patterns, respectively. Rewriting the likelihood for quartet i as $L_i(\tau_{u_i}, \tau_{v_i}, \tau_{w_i}, \theta|\mathbf{Y})$, corresponding to Equation 4, the composite likelihood is given by

$$\ell(\tau_1, \tau_2, \ldots, \tau_R, \theta|\mathbf{Y}) \propto \prod_{i=1}^{Q} L_i(\tau_{u_i}, \tau_{v_i}, \tau_{w_i}, \theta|\mathbf{Y}) = \prod_{i=1}^{Q} \prod_{j=1}^{n(i)} p_j^{y_j} \,. \tag{5}$$

The parameter values for which this function is maximized are called maximum composite likelihood estimates (*MCLEs*). See Figure 9 for a depiction of the process of computing the composite likelihood for a 5-taxon tree.

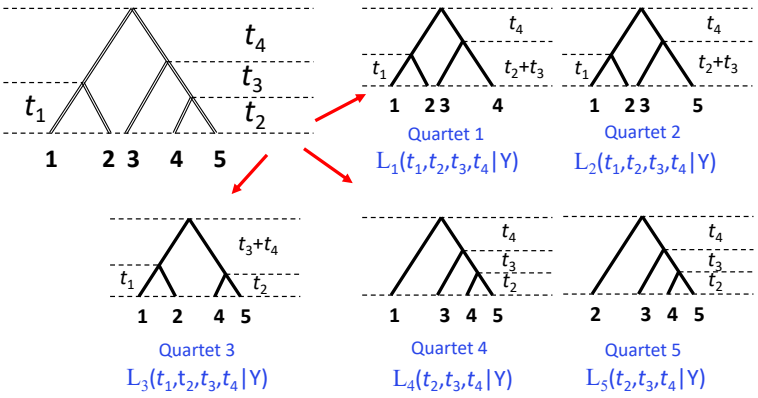


Figure 9: Schematic representation of the computations of the composite likelihood for a 5-taxon species tree (top left) with branch lengths t_1, t_2, t_3, and t_4. Red arrows point to the five possible quartet trees that are obtained by removing one taxon from the species tree. Branches for each quartet tree are re-labeled appropriately, and the likelihood based on the multinomial likelihood is computed for each, as indicated by the terms below each tree. To compute the composite likelihood, the terms below the five quartet trees are multiplied.

Note that the procedure above entails multiplication of the quartet likelihoods, even though they are not independent. If the quartet likelihoods were independent of one another, then multiplying them together would lead to computation of their joint probability. However, this will not be the case when there is dependence among the quartets, which is why a likelihood formed in this way is termed a *composite likelihood* (or *pseudolikelihood*). To see the lack of independence, refer to Figure 9, which shows the five quartet trees that can be obtained from the species tree in the top left. Comparing Quartets 1 and 2, for example, we observe that they share three taxa in common and that both include the same set of branch lengths. Thus, they use overlapping sets of data to compute their quartet likelihoods, and they share a set of parameters (the three branch lengths on their quartet trees), so they are clearly not independent of one another. Similar comparisons can be made among all pairs of quartets.

Intuitively, the composite likelihood is useful for inference because values of the parameters that maximize the true likelihood are likely to lead to large values for each of the quartet likelihoods as well. Thus the composite likelihood contains information about which values of the parameters are likely to provide reasonable explanations of the data. Indeed, there is a rich literature on the use of composite likelihood approaches for inference in situations where computation of the full likelihood is infeasible, including theoretical results about the statistical properties of estimators obtained using composite likelihood approaches (see, e.g., [26] for a recent review). Peng et al. (2021) have shown that the values of the speciation times that maximize the composite likelihood in Equation 5 are statistically consistent and asymptotically normally distributed, by proving that the results of [4] can be applied in this case. This enables implementation of a computationally efficient method for obtaining estimates of these parameters with good statistical properties.

Instead of using composite likelihood directly, however, we have found it preferable to estimate parameter values via Bayesian maximum *a posteriori* (MAP) estimation. In addition to allowing incorporation of prior knowledge into the estimate, weighting the (composite) likelihood by the priors improves the computational efficiency and stability of the optimization algorithms by reducing the flatness of the optimality surface in regions of the parameter space that have very low likelihood (see below). Prior distributions are placed on both θ and the total tree height $h{=}\tau_R$ (R is the index

of the root node). If $f_\theta(\theta)$ and $f_{\mathrm{h}}(\tau_R)$ represent the probability density functions for these prior distributions, then the log of the posterior density is proportional to

$$\log g(\tau_1, \tau_2, \ldots, \tau_R, \theta|\mathbf{Y}) = \log f_\theta(\theta) + \log f_{\mathrm{h}}(\tau_R) + \sum_{i=1}^{Q} \log L_i(\tau_{u_i}, \tau_{v_i}, \tau_{w_i}, \theta|\mathbf{Y}) \tag{6}$$

(g is an unnormalized posterior density because it lacks a marginal likelihood term). We refer to the parameter values that optimize this function as MAP_{CL} estimates, with the subscript "CL" signifying that a composite-likelihood term is used in Equation 6 rather than a true likelihood. Note that PAUP* allows frequentists to omit the prior terms in Equation 6 to obtain MCLEs instead, although for reasons explained below, larger amounts of data may be needed to avoid numerical difficulties during the optimization.

3.2 Algorithmic details

The goal is to obtain estimates of the speciation times and the effective population size that maximize the posterior density function g in Equation 6. Several algorithmic issues arise in the development of a computationally efficient estimator that performs well, and we briefly describe those here.

A key concern is that the value of the effective population size, θ, has a large impact on the estimates of the speciation times, but the composite likelihood surface is relatively flat with respect to the effective population size parameter. Thus routines for numerical optimization are extremely sensitive to the starting point chosen for θ, making it important to choose a good initial value for this parameter. To address this complication, a grid search is first carried out in conjunction with the moment-based estimators for the speciation times given in [14] to determine an interval that provides loose lower and upper bounds on the optimal θ value. The starting θ value is then optimized numerically using a two-step procedure in which the bracket is first narrowed via a golden section search, then polished using a derivative-free one-dimensional optimizer. Initial values for the speciation times are determined using the moment-based estimators computed at the starting value chosen for θ.

We then use multidimensional optimization techniques to search numerically for values of the speciation times and effective population size parameters that maximize the log posterior density. This optimization involves reparameterization of the speciation-time parameters in order to enforce the constraint that a node cannot be older than its parent, as well as variable transformations to eliminate bounds, facilitating efficient unconstrained optimization. Partial derivatives of function g with respect to the transformed parameters can be computed efficiently, allowing use of gradient-based optimizers; we use the quasi-Newton L-BFGS algorithm.

Full details of the PAUP* implementation are provided in the Supplemental Material to [18], which interested readers can consult for more information.

3.3 Uncertainty quantification

As was the case when estimating the species tree topology using SVDQuartets, it is important to provide a measure of the uncertainty associated with the MAP_{CL} estimates. While the application of results from [4] provides an explicit expression for the asymptotic variance, we have found this variance estimator to be unstable in practice. Thus, we recommend instead that bootstrapping be used to estimate the variance. The implementation of the MAP_{CL} method in PAUP* includes options for both the standard and the multilocus bootstrap, and we have found that these methods perform well in practice [18].

3.4 Application to species relationships among gibbons

To demonstrate the performance of the MAP_{CL} estimators, we return to the gibbon data analyzed with SVDQuartets in the previous section. Using the tree estimated by SVDQuartets (Figure 7), we obtain estimates of the speciation times as well as their variances using the PAUP* command: `qage patProb=exactJC taxpartition=gibbonspecies loci=lociset bootstrap=multilocus;`. The estimates are shown in Figure 7. We note that the MAP_{CL} procedure provides estimates that are both fast and accurate – this analysis took 10.79 seconds on a desktop machine, and compares favorably to the estimates provided by BPP using a much longer run time, as reported in [18]. In addition, the MAP_{CL} estimates were found to be much more robust to the choice of prior distribution than those obtained by BPP [18].

3.5 Recommendations for using composite likelihood estimators of the speciation times

As described above, the MAP_{CL} estimators assume the JC69 model under the MSC. We have used simulations (not shown here) to demonstrate that the method can be somewhat sensitive to this assumption. To handle cases where more general substitution processes may be operating, PAUP* includes an option to compute the MAP_{CL} estimation procedure under more complex substitution models. The primary difficulty involved in using more general models is that closed form expressions for the site pattern probabilities are no longer available, and thus these site pattern probabilities must be approximated numerically. The implementation of this step in PAUP* is very accurate, but requires some additional computation time. Nonetheless, estimation using the MAP_{CL} procedure will generally be much more efficient than the corresponding Bayesian approach.

The MAP_{CL} estimates will benefit from larger data sets in the same way that SVDQuartets does. The fact that MAP_{CL} is statistically consistent means that these estimates become increasingly accurate as the sample size increases. As was the case for SVDQuartets, there is little computational cost associated with increasing the number of sites in the data, as counting the number of each type of site pattern is a rapid procedure that need be done only once since the species tree is fixed in this case. Thus, we recommend the MAP_{CL} estimates when the number of sites in the data set, whether multilocus or CIS, is large.

4 Conclusion and Future Work

Statistical methods for inferring species trees and associated parameters using site pattern frequencies provide computationally efficient, model-based approaches with provable statistical properties, including consistency. The methods described here, SVDQuartets and MAP_{CL} estimation of speciation times, have been implemented in PAUP* in a user-friendly interface, making them widely accessible. We have provided a tutorial that can be downloaded and/or viewed that replicates the analyses for the gibbon data that are included here. The tutorial is available at `https://phylosolutions.com/tutorials/svdq-qage/`.

Site pattern probabilities have also been applied to other problems in species-level phylogenetic inference. For instance, they are used to form both the HyDe [5] and ABBA-BABA [9] statistics that allow for identification of species that have arisen via hybridization. They have also been used to identify the root position of a species tree [25]. Because they enable rapid computations under the MSC model, methods based on site pattern frequencies are promising approaches for computationally efficient species tree inference for large-scale data sets.

We note, however, that good performance of methods based on site pattern frequencies depends crucially on having obtained a large sample. For CIS, this means that many independent sites are needed, while for multilocus data, it means that the number of loci should be reasonably large (certainly more than $\sim$50, but ideally several hundred to a thousand or more). Our view is that methods based on site pattern frequencies offer a strong alternative approach to Bayesian methods in cases where the data are too large to allow Bayesian inference in a reasonable time. When data set sizes are smaller, Bayesian approaches can be expected to perform better and offer the advantage of a full characterization of the posterior distributions of interest. However, as sequencing advances continue to outpace the development of computational tools that can efficiently analyze the wealth of available data, methods based on site pattern frequencies will be invaluable in carrying out efficient inference in a theoretically-justified framework.

References

[1] E. S. Allman, L. S. Kubatko, and J. A. Rhodes. Split scores: a tool to quantify phylogenetic signal in genome-scale data. *Systematic Biology*, 66(4):620–636, 2017.

[2] E. S. Allman and J. A. Rhodes. Phylogenetic Invariants. In O. Gascuel and M. Steel, editors, *Reconstructing Evolution: New Mathematical and Computational Advances*, chapter 4, pages 108–146. Oxford University Press, 2007.

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[4] B. C. Arnold and D. Strauss. Pseudolikelihood estimation: Some examples. *Sankhya: The Indian Journal of Statistics, Series B*, 53:233–243, 1991.

[5] P. Blischak, J. Chifman, A. D. Wolfe, and L. S. Kubatko. HyDe: a Python package for genome-scale hybridization detection. *Systematic Biology*, 67(5):821–829, 2018.

[6] L. Carbone, R. A. Harris, S. Gnerre, K. R. Veeramah, B. Lorente-Galdos, J. Huddleston, T. J. Meyer, J. Herrero, C. Roos, and B. Aken et al. Gibbon genome and the fast karyotype evolution of small apes. *Nature*, 513:195–201, 2014.

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

[8] J. Chifman and L. Kubatko. Identifiability of the unrooted species tree topology under the coalescent model with time-reversible substitution processes, site-specific rate variation, and invariable sites. *Journal of Theoretical Biology*, 374:35–47, 2015.

[9] E. Y. Durand, N. Patterson, D. Reich, and M. Slatkin. Testing for ancient admixture between closely related populations. *Molecular Biology and Evolution*, 28(8):2239–2252, 2011.

[10] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[11] J. Gaither and L. Kubatko. Hypothesis tests for phylogenetic quartets, with applications to coalescent-based species tree inference. *Journal of Theoretical Biology*, 408:179–186, 2016.

[12] T. Jiang, P. E. Kearney, and M. Li. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM Journal on Computing*, 30:1942–1961, 2000.

[13] L. Kubatko. The multispecies coalescent. In D. J. Balding, I. Moltke, and J. Marioni, editors, *Handbook of Statistical Genomics, Fourth Edition, Volume 1*, pages 219–245. John Wiley & Sons Ltd., 2019.

[14] L. Kubatko and J. Chifman. Identifiability of speciation times under the multispecies coalescent. *preprint bioRxiv: doi:10.1101/2020.11.24.396424*, 2020.

[15] C. Long and L. Kubatko. Hypothesis testing with rank conditions in phylogenetics. *Frontiers in Genetics (special issue on Algebraic and Geometric Phylogenetics)*, page in preparation, 2021.

[16] C. L. Long and L. Kubatko. Identifiability and reconstructibility of species phylogenies under a modified coalescent. *Bulletin of Mathematical Biology*, 81:408–430, 2019.

[17] C. L. Long and L. S. Kubatko. The effect of gene flow on coalescent-based species tree inference. *Systematic Biology*, 67(5):770–785, 2018.

[18] J. Peng, D. Swofford, and L. Kubatko. Estimation of speciation times under the multispecies coalesent. *in revision*, 2021.

[19] B. Rannala and Z. Yang. Likelihood and Bayes estimation of ancestral population sizes in hominoids using data from multiple loci. *Genetics*, 164:1645–1656, 2003.

[20] R. Reaz, Md S. Bayzid, and M. S. Rahman. Accurate phylogenetic tree reconstruction from quartets: A heuristic approach. *PloS One*, 9(8):e104008, 2014.

[21] C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, 2003.

[22] T.-K. Seo. Calculating bootstrap probabilities of phylogeny using multilocus sequence data. *Molecular Biology and Evolution*, 25(5):960–971, 2008.

[23] C.-M. Shi and Z. Yang. Coalescent-based analyses of genomic sequence data provide a robust resolution of phylogenetic relationships among major groups of gibbons. *Molecular Biology and Evolution*, 35:159–179, 2018.

[24] S. Snir and S. Rao. Quartets maxcut: a divide and conquer quartets algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 704–718, 2010.

[25] Y. Tian and L. Kubatko. Rooting phylogenetic trees under the coalescent model using site pattern probabilities. *BMC Evolutionary Biology*, 17:263, 2017.

[26] C. Varin, N. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21:5–42, 2011.

[27] M. Wascher and L. Kubatko. Consistency of SVDQuartets and maximum likelihood for coalescent-based species tree estimation. *Systematic Biology*, 70(1):33–48, 2021.