

Supertree methods for building the tree of life: divide-and-conquer approaches to large phylogenetic problems

Mark Wilkinson and James A. Cotton

Both authors are at:

Department of Zoology, The Natural History Museum, London, UK.

(MW) Tel: +44 (0)207 942 5164, Fax: +44 (0)207 942 5054, mw@bmnh.org.

(JAC) Tel: +44 (0)207 942 5306, Fax: +44 (0)207 942 5054, james.cotton@nhm.ac.uk

1. INTRODUCTION

2. DIVIDE-AND-CONQUER METHODS

3. EFFECTIVE OVERLAP

3.1 THE IMPORTANCE OF PHYLOGENY

3.2 IMPORTANCE FOR EXPERIMENTAL DESIGN AND FUTURE SAMPLING

4. FAST QUARTET-BASED SUPERTREE CONSTRUCTION

4.1 VOTING SYSTEMS

4.2 USING FEWER QUARTETS

4.3 QUARTET JOINING

5. CONCLUSION

6. REFERENCES

Abstract

Reconstructing the tree of life will require fast methods for building very large phylogenetic trees from patchy data. The leading candidates for such an approach employ supertree methods as part of a divide-and-conquer strategy. Here, we discuss two aspects of a phylogenetic divide-and-conquer method – the decomposition of the tree into subproblems, and the recombining of these into an overall solution. In particular we highlight and explore the issue of effective taxon overlap, how it might be achieved via suitable decomposition and how it might be used to guide the setting of priorities for additional data acquisition, and we show how some knowledge of phylogeny is vital in both contexts. Lastly, we show that quartet puzzling, the best-known phylogenetic divide-and-conquer method, can perform poorly when not all quartets are available and we present a new, fast, supertree method designed to perform better in this context. While a great deal of work remains, such an approach has great potential as part of a divide-and-conquer method for reconstructing large phylogenies on the scale of the tree of life or for large subsets of species-rich taxa.

1. Introduction

Reconstructing the complete history of life is an ultimate goal of biology¹ and recent interest in constructing the phylogenetic “tree of life” reflects the central role of phylogenetic trees in understanding evolutionary history^{2,3}. Notwithstanding that most living species may be as yet undescribed, there are major methodological challenges in realising the tree of life. While there has been some debate⁴⁻⁸ we generally expect accurate reconstruction of phylogenetic trees to become more difficult as trees become larger. The main reason for this is computational complexity. As is well known, the number of possible trees grows more than exponentially with the number of taxa on the tree, so as we seek to identify optimal trees under some objective function the size of tree space (the set of trees for the relevant set of taxa) in which we hope to locate them becomes impossibly large. Exact methods such as exhaustive searches or branch-and-bound algorithms are prohibitively time consuming for all but the smallest phylogenetic problems, and for any substantial problem we are forced to rely on heuristics to guide a limited search of tree space in the hope of finding good (optimal or near optimal) trees. As the size of tree space grows, such searches will become more and more difficult, as they search for one or more needles in an ever-expanding haystack. Building trees as large as the tree of life (of the order of millions of taxa) using any known heuristic will be unfeasible, and new approaches will be needed.

A second difficulty in reconstructing the tree of life is the patchy availability of data for different leaves. DNA sequences have become the principal source of data for phylogenetic reconstruction and are accumulating at a rapid rate. As the number of leaves increases, however, it becomes increasingly unlikely that a single gene or single source of data is available for all the taxa, or that a single gene will be effective in

reconstructing their relationships. Thus we can expect some information to be unavailable for some taxa (“missing data”). Extensive non-random missing data may complicate or compromise analyses⁹. Furthermore, if different markers are needed for different taxa, then accurate analysis will probably need to account for heterogeneity between these markers^{10,11}. Modelling this heterogeneity can be complex and will tend to make methods for analysing such data slow.

A solution to both of these problems may be to use divide-and-conquer approaches in which large phylogenetic problems are decomposed into subproblems and the solutions of these subproblems combined to give a global solution. Such approaches reflect the expectation that subproblems can be more easily analysed separately because they are smaller in size and because they can include just those taxa for which a particular type of data is available, reducing the problem of missing data and allowing the process of evolution for particular data to be more accurately modelled. A decomposition might be a natural one, such as dividing a large molecular dataset into data from individual orthologs, or could be designed to yield subproblems that should be more easy to solve accurately and that are readily recombined. The problem of combining a set of phylogenetic trees into a single estimate of phylogeny is addressed by supertree methods, which are therefore integral to any divide-and-conquer approach to building large phylogenetic trees. For example, quartet puzzling¹² is perhaps the best known divide-and-conquer approach to phylogenetic inference and the puzzling step is a heuristic supertree method.

In this chapter our main aims are to draw attention to the problem of achieving effective overlap between subproblems and to outline a new fast supertree method. Speed is an obvious important consideration in building large phylogenetic trees but the importance

of effective overlap and how it might most efficiently be achieved has been less widely appreciated. We begin with an overview of divide-and-conquer methods

2. Divide-and-conquer methods

Divide-and-conquer is a standard approach to solving difficult computational problems by splitting them into smaller, easier (often trivial) subproblems which can be independently solved, and then combined to give a global solution¹³. A number of well-known algorithms use a divide-and-conquer approach, such as merge sort and quick sort and the fast fourier transform. The efficiency of solving the subproblems and the efficiency of this merging process will determine how effective a divide-and-conquer approach can be. While classical divide-and-conquer algorithms provide globally optimal solutions to problems, this is probably an unrealistic aim for phylogenetic methods, given that most optimisation-based phylogenetic problems are known to be, or likely to be, NP-complete¹⁴ so there is very unlikely to be a polynomial-time algorithm to solve them (NP-completeness has been shown for parsimony¹⁵, compatibility¹⁶, distance metrics¹⁷ and at least one likelihood problem¹⁸). Thus we should expect phylogenetic divide-and-conquer strategies to be heuristic rather than exact algorithms. There is no guarantee that solutions to subproblems will be accurate and thus no guarantee that they will all be compatible or readily combinable. Even apparently disjoint subproblems may be incompatible (while quartets with less than three leaves in common must be pairwise compatible, three such quartets can be incompatible). In the phylogenetic context, there may be choices to be made in the order subproblems are combined and even between which sets of subproblems to consider at all.

Most uses of supertree methods have been to build larger phylogenies from sets of previously published trees. While this is divide-and-conquer analysis of a sort (the published trees can be thought of as the results of a given decomposition of the overall problem) we prefer to view divide-and-conquer approaches more narrowly as those in which a designed decomposition of the problem is integral to the analysis. Here, divide-and-conquer analyses offer methods for inferring trees from large datasets rather than from sets of previously inferred trees, and the criterion for choosing among alternative inferences, be it parsimony, likelihood or something else, need be no different from that used by other methods of analysing large datasets.

An important advantage of divide-and-conquer approaches is that they may be relatively computationally efficient¹⁹. While only two supertree methods have been studied in the divide-and-conquer setting²⁰, the structure of divide-and-conquer algorithms suggest that many existing supertree methods are perhaps unsuitable for such use. In particular, optimisation supertree methods such as MRP (Matrix Representation with Parsimony) require time-consuming heuristic searches of trees, and combining subproblems for large sets of taxa using these methods will take just as long as solving the problem in a single step (it may or may not be more accurate). These optimisation methods will not be suitable for the amalgamation step in a divide-and-conquer strategy that hopes to be quicker than a conventional analysis: we need faster supertree methods that take a length of time proportional to some polynomial in the number of input taxa. MinCut²¹ and modified MinCut²² supertree methods are both polynomial time, as is the strict consensus merger (SCM²³). However, while saving time is important, accuracy is paramount. MinCut supertrees have been shown to be less accurate than trees constructed using other methods in simulation studies²⁴ and show a significant bias with respect to shape²⁵ that might be correlated with poor accuracy. There is clearly scope for

new, fast supertree methods in the context of divide-and-conquer approaches (see below).

Exact divide-and-conquer algorithms generally break a problem down into the smallest, trivial subproblems. Similarly, quartet puzzling breaks a phylogeny problem into the smallest meaningful (unrooted) problem of quartets of taxa. Solving quartets is possible very quickly, as only 3 different quartets need to be compared for a four taxon subproblem. However, some quartets may be difficult to accurately infer and a heuristic analysis of larger subproblems might be quicker or more accurate, leading to an optimal “granularity” of the decomposition for particular problems.

3. Effective overlap

It is widely recognised that the efficacy of supertree construction is contingent upon which taxa are shared between different input trees (their overlap), but what distinguishes effective and ineffective overlap? To simplify matters we consider the special case of compatible input trees. If two (or more) trees are compatible then there exists at least one supertree that displays both (or all) the input trees. The set of supertrees that display all the input trees is termed the span of the input trees and denoted $\langle S \rangle$. The strict component consensus of $\langle S \rangle$ is referred to here as the consensus supertree. Figure 1 gives an example of two compatible input trees and their consensus supertree in which there is a mixture of effective and ineffective overlap²⁶. $\langle S \rangle$ includes seven fully resolved supertrees that differ only in the placement of leaf 10 with respect to leaves 1, 2, 6, 7, and 8. Although it mostly does a good job of combining the information in the two input trees the consensus supertree conveys much less information about the relationships of these leaves than does tree 1. Dealing with

compatible trees allows a natural definition of effective and ineffective overlap: effective overlap occurs when the consensus supertree displays all of the input trees, while overlap is ineffective to the extent that information present in the input trees is not present in their consensus supertree. In this example it is easy to see that there is mostly good overlap but that there is not sufficient information in the input trees to determine the relationships of leaves 6-8 from tree 1 to leaf 10 from tree 2.

3.1 *The importance of phylogeny*

Overlap has mostly been considered only in terms of the number of leaves in common (e.g. [27]). Existing approaches to designing meta-analyses of sequence data have focused on maximising this overlap, either by identifying datasets for which all genes are available for all taxa²⁸ so that there are no missing data, or by minimising the number of gene-taxon pairs that are missing from a dataset, with no consideration of the phylogenetic relationship between taxa²⁹. The first approach leads to very conservative datasets in which most of the available data needs to be discarded, while neither approach directly addresses the effectiveness of the overlap. Given a pair of unrooted trees the minimum requirement for effective overlap is three common leaves (two leaves plus the root in rooted trees). However, as our example shows, effective overlap depends upon both the number of common leaves and their relationships to each other (see also [30]).

We can show the importance of the phylogeny of the input leaves with some simple examples. The three simple cases shown in figure 2 each consist of a pair of trees with two leaves in common, but differ greatly in the relative positions of the common leaves. These examples show that the common taxa occurring as sister taxa in both trees gives

no effective overlap between the trees, and neither does the two taxa occurring widely separated on both trees. The optimal situation appears to be when a small clade in one tree spans a deep split in the second tree, as in the second example. Figure 3 shows this result more generally. We can define the separation of any pair of taxa (and mean separations of any sets of taxa), on a given tree by counting the number of internal edges separating them. This quantity might be helpful in weighting simple co-occurrence metrics used to assess overlap – if the above result holds in more complicated cases, then it should be optimal to have taxa with highly different mean separations across different input trees. Measures developed in different contexts also might prove useful here, such as the proportion of phylogenetic history sampled by a set of taxa³¹.

3.2 Importance for experimental design and future sampling

Achieving effective overlap has been an important practical issue in supertree construction. With the exception of quartet puzzling trees, most published supertrees have been constructed from input trees harvested from the literature. Although such use of existing phylogenetic inferences has allowed the assembly of several large scale phylogenies without the need to reanalyse primary data³²⁻³⁴, achieving effective overlap has sometimes required the use of comprehensive taxonomic hierarchies, interpreted as phylogenies³⁴. Faced with needing to rely upon taxonomies or confronted with ineffective overlap, the obvious question is how best to improve overlap. Answering this question has obvious importance for guiding the prioritisation and targeting of additional data acquisition so as to most efficiently bridge the gaps. Our aim here is more to highlight the importance of this question than to address it, but a few preliminary comments may be worthwhile.

If we have non-overlapping data and trees for genes X and Y with some ineffective overlap identified or suggested by a poorly resolved supertree, then we can ask how the tree for X impinges upon the choice of which leaves in Y should be sequenced for gene X. Consider again the example from Gordon²⁶ shown in figure 1, which we can think of as showing two trees constructed from two different genes. There are a number of possible choices of additional sampling of genes for particular taxa that might improve overlap sufficiently to allow the consensus supertree to display both input trees, and some selections that would be less likely to help. For example, assuming that the new sequences introduce no conflict in the input trees then obtaining additional data for leaves 11-14 so as to include them in tree 1 would obviously produce no practical improvement in overlap. An obvious choice that would provide completely effective overlap would be to sample 10 to include it in tree 1, but we could alternatively sample from 6, 7, and 8 for the tree 2 gene. Other things being equal we would target whichever (10 or 6, 7, 8) were most convenient or least expensive to obtain. If we are constrained by the availability of samples or other resources (e.g. if we had sufficient funding to sample a single gene for a single taxon and gene 1 cannot be sampled for taxon 10), we can ask which of the remaining targets should be our priority. Thus we would ask whether the available phylogenetic information suggests that additional data for one of 6, 7, and 8 would be most likely to provide effective overlap. Sampling taxon 6 and adding it to tree 2 on branches A, B, or C, gives a fully resolved supertree if on branch A or B but not on C, while sampling 7 (or 8) on this tree gives a fully resolved supertree on branch C, but not on either A or B. Sampling taxon 6 is more likely to resolve the problem, and so might be an optimal choice for this gene.

Similar issues need to be addressed when designing a divide-and-conquer algorithm. Different subproblems need to overlap to some extent if they are to be combined in a global solution, but too much overlap will lead to the same relationships being inferred many times, making the algorithm inefficient. The problem of designing a decomposition of a particular tree so as to provide effective overlap is trivial. Simultaneously providing effective overlap and easily solvable subproblems is more challenging, particularly without knowledge of the tree. Previous workers have designed decompositions that give subproblems that are easily solved, and are even provably easy to solve. Huson et al. created the original disk-covering method to produce subproblems of minimal “evolutionary diameter” in that taxa in a particular subproblem have small pairwise sequence divergence²³. Distance-based methods such as neighbour-joining are known to be accurate for such data, and this allowed Huson et al. to prove a number of theorems about the accuracy of analysis using their decomposition together with these methods. This “disk-covering method” or DCM decomposition did not attempt to control the degree of overlap, however, and so performs poorly in the more general supertree context^{19,20}. A second method, DCM2, identifies a single set of taxa which have bounded diameter and which produces subproblems of bounded diameters such that the largest subproblem is as small as possible³⁵. Both DCM2 and a related recursive alternative (Rec-I-DCM3) have been shown to be remarkably effective divide-and-conquer algorithms^{20,36} (see also Bininda-Emonds and Stamatakis, this volume), but it remains unclear whether a strategy in which all subproblems share a set of common taxa is preferable to one in which pairs of subproblems have different shared taxa. We note in passing that consensus efficiency³⁷ which is the ratio of the cladistic information content³⁸ of a consensus to that of a set of trees (such as the span) provides a potential measure of the efficacy of overlap which could be used to compare different decompositions of the same dataset.

If different taxonomic groups are studied using different molecular markers, as they inevitably will be to some extent, then the tree of life can only be inferred by combining individual studies. It would be helpful to be able to give some guidance to molecular systematists as to how to design such studies (focusing on their particular taxonomic group of interest) to be easily combinable in this context – for example, it might be best to sequence a marker for a few closely related taxa (as is currently done for e.g. outgroup rooting) or it might be better for every study to include a few of a selected set of systematic “model organisms” which might, but need not, coincide with the model organisms of molecular biologists, for many of which complete sequence data is already available. Tentatively, it seems that the first solution is likely to be better, and we might encourage systematists to sample closely related sequential sister-groups to their clade of interest in molecular studies (figure 4). Further work is needed and our limited discussion and exploration of the simplest examples is intended simply to highlight this need.

4. Fast quartet-based supertree construction

Perhaps the best-known and most widely used divide-and-conquer approach in phylogenetics is the quartet puzzling (QP) method of Strimmer and von Haeseler¹². This method has three steps. Firstly, trees are inferred for all quartets of leaves using some objective function. Secondly, 'puzzling' is used to combine the quartet inferences to produce a tree for all the leaves. In puzzling, an initial quartet is selected and additional leaves are added sequentially, with the position at which they are grafted to the growing tree determined as a function of the votes cast by the relevant quartets – those that include the new leaf and any three leaves already in the tree and thus convey

information on the position of the new leaf on the growing tree. The result may be contingent on the choice of starting quartet and the order of addition of leaves. Consequently the puzzling step is usually repeated many times, and the third step of QP is then the construction of a majority-rule consensus of the resulting trees. The frequencies of relationships found in these trees can be taken as an indication of their relative support.

4.1 Voting Systems

The puzzling step is a supertree method and it can take also as input any set of weighted quartets, including those displayed by a set of input trees³⁹. However, QP differs from other supertree methods in having been designed for the analysis of a single data set from which inferences about all or most quartets can be made. In contrast, supertree construction is more typically based on input trees that display relationships among only a fraction of the quartets. It has been suggested that to be well-suited to the latter, the voting method used in puzzling requires some modification^{40,41}. Consider the twelve quartets in Fig. 5a, which are all compatible and jointly entail a single tree (Fig 5b). Taking these quartets as input, we would expect a supertree method to yield just this tree, as does, for example, parsimony analysis of character encodings of the quartets. In contrast, applying the puzzling step (1,000 times) with these quartets as input does not yield the expected tree and all support values are low (Fig. 5c), a quite unsatisfactory result.

In the original puzzling voting procedure of Strimmer and von Haeseler¹² each relevant quartet is taken in turn. In each of these, the new leaf is paired with one of the three leaves already in the growing tree. We find the path between the other two leaves in the

growing tree and give a score of +1 to every edge on that path (Fig. 6a). This is a vote against the grafting of the new leaf to the tree on any of those edges. The votes of all relevant quartets are counted and the new leaf attached to an edge with the smallest vote against, ties being broken randomly (Fig. 6c). Strimmer et al. subsequently developed an approximate system for weighting the votes of quartets according to their posterior probabilities that is employed in TREE-PUZZLE^{42,43}.

The inadequacy of the QP voting system in the more general supertree case, i.e., where we do not have the luxury of votes from all possible relevant quartets and have to rely upon a subset of them, is readily demonstrated and diagnosed. Consider in our example (Fig. 6) that we have only the single quartet AE/BC to vote on the position of the new leaf E. The puzzling voting system leaves a tie between two branches, which are thus equally likely inferred placements of E. However, only one of these placements (with A) is consistent with what the quartets actually entail about the relationships of E. The other (with D) actually contradicts the information in the relevant quartet because it entails AB/CE. That the puzzling voting system does not provide a vote against this illogical position may not matter when all or nearly all quartets are available, because other quartets (e.g. AE/BD) may vote directly against this position, but it is expected to compromise its performance, as we have already seen, when not all relevant quartets are available.

An alternative voting procedure, used by Vinh and von Haeseler⁴⁴ in the somewhat different context of an algorithm which efficiently elucidates the landscape of possible optimal trees, seems much better suited for the supertree context. For any three leaves, A, B, and C, in a tree there is a unique node or vertex where the paths connecting each pair of these leaves intersect and which is subtended by three subtrees (one containing

A, one containing B and one containing C) which we shall call the subtrees of the node. The resolution of a quartet on A, B, C and a new leaf tells us to which subtree the new leaf must be grafted in order for the quartet to be displayed by the tree. Other positions contradict the quartet. Thus instead of voting against branches lying on a particular path, we can vote either for all branches in the subtree in which any grafting of the new taxon would display the quartet, or against all branches in the subtrees in which grafting would contradict the quartet. What is entailed by a pair of quartets is governed by dyadic inference rules of which there are just two⁴⁵⁻⁴⁷. Vinh and von Haeseler's⁴⁴ voting system reflects these simple inference rules extended to the case where one of the quartets being compared is embedded in a larger tree.

Although not suggested by Vinh and von Haeseler⁴⁴, we could use their voting system in place of the original puzzling step of QP. Taking the twelve quartets in figure 5a as input, a QP-type analysis using this alternative voting system would return the unique tree defined by the quartets (Fig. 5b; Fig. 7) with maximum support for all splits, a far more satisfactory result than unmodified QP (Fig. 5c). The comparative performance of this alternative voting system in more typical QP-type analysis merits further investigation. Certainly we would expect it to offer improvements if QP is used to analyse very incomplete or patchy supermatrices which do not support resolutions of many quartets and might therefore be expected to expose the limitations of the current QP voting system.

4.2 Using fewer quartets

One potential problem with quartet methods is that the number of possible quartets increases substantially (n^4) with the number of leaves (n). For example, for 500 taxa

there are more than 2.5×10^9 quartets. While the number of quartets is far fewer than the number of possible trees, solving all quartets seriously limits the efficiency that may be obtained from breaking a large problem into more tractable quartet problems. In addition the number of quartets whose votes must be counted to determine the placement of each new taxon in the puzzling step also increases polynomially (n^3) with the number of leaves in the growing tree, giving a complexity for the puzzling step of $O(n^4)$.

In contrast, the minimum number of quartets needed to uniquely specify a tree increases only linearly with the number of leaves⁴⁸. This suggests that considerable improvements in speed might be obtained by focussing only upon privileged subsets of quartets that are sufficient to specify a tree. Knowing the tree it is easy to find minimal sets of quartets that fully specify the tree (e.g. the quartets in Fig. 5a are one such set). More typically in phylogenetics we are trying to infer an unknown tree, which would make the selection of appropriate quartets rather more difficult. Fortunately, our precise problem is to find a privileged set of quartets sufficient to efficiently place a single leaf on an otherwise known tree, which is considerably more tractable.

Vinh and von Haeseler⁴⁴ defined a natural ranking of the leaves of the subtrees of a node in terms of their 'distance' (no. of edges) from that node and used this to define subsets of leaves called *k-representative sets* comprising the k leaves closest to the node (with random breaking of ties). Motivated by the desire to speed puzzling by relying upon the votes of fewer quartets (hence their unexplained modified voting system), while using quartets likely to provide the most accurate placements, they defined the k^3 important quartets with respect to a node as those including a new leaf and one leaf from each of the k -representative sets of each of the three subtrees of the node. The

important quartets of a tree are all those that are important quartets of any node in the tree and time can be saved by permitting only these quartets to vote.

Important quartets were used by Vinh and von Haeseler⁴⁴ to vote on the reattachment of leaves that have been deleted from a tree, as part of a method for exploring tree space. They note that using only important quartets in QP would yield a decrease in the complexity of puzzling, from $O(n^4)$ to $O(n^2)$ but because of poor performance in simulations they did not pursue this further. Better performance might be expected using their alternative voting procedure, given the frailty of unmodified QP when not all relevant quartets are available. In their study, Vinh and von Haeseler set k to four, but in the extreme k could be set to one, so that there are just $N-2$ important quartets for the tree. This is the minimum number needed to uniquely specify a tree for $N+1$ leaves, but there is no guarantee that minimal sets of important quartets will specify a tree: there may be conflict.

4.3 Quartet Joining

Concentrating on speed, we suggest an alternative approach which carries the divide-and-conquer approach a step further. Each relevant quartet provides information on the placement of a new leaf with respect to one internal node, by indicating to which subtree the leaf must be added so as to display the quartet. Thus, if we accept the subtree placement implied by one or more relevant quartets the problem then becomes that of placement of the leaf within that subtree, which can be further addressed with one or more quartets relevant to its position with respect to a node in the subtree, and so on until the position is uniquely specified or there are no more relevant quartets (in which case a placement in the remaining subtree is chosen at random or its addition delayed in

favour of another leaf). The number of relevant quartets that need to be consulted is thus bounded by the number of nodes and increases linearly with the number of leaves giving a complexity of $O(n^2)$. The one or more quartets used in each step of this iterative divide-and-conquer procedure could be important quartets in the sense of Vinh and von Haeseler⁴⁴ but need not be.

The basic idea of this approach, which we call *quartet joining* is to grow a tree through a series of refinements of the problem of adding leaves that make use of dyadic inference rules. Importantly, the order in which refinements of the problem are sought can further affect the speed of tree construction. We can rank the nodes in a growing tree or any subtree by the number of possible relevant quartets (which is equal to the product of the numbers of leaves in the three subtrees of the node). If we always resolve first the position of the new leaf with respect to the nodes with the highest number of possible relevant quartets, this (at least) halves the problem at each step leading to a complexity of only $O(n \log n)$.

Quartet joining requires only a single relevant quartet to resolve the placement of a leaf with respect to a particular node (Fig. 7). With the 12 quartets of Fig. 5a as input, quartet joining returns the single tree jointly entailed by the quartets with maximal support. More generally, for any compatible set of quartets the method will return one or more trees that display or extend the dyadic closure of the quartets. Thus to maximise speed quartet joining would consult a single quartet when considering the placement of a new leaf with respect to a node. Compared to the democracy of QP, and the oligarchy of Vinh and von Haeseler's important quartet puzzling, each quartet consulted in this extreme form of quartet joining dictates the placement of the new leaf, its vote is decisive, and the consulted quartets never conflict. However the method can also accommodate the votes

of multiple relevant quartets at each step (e.g. those of important quartets) in order to improve accuracy with only linear increase in complexity. Further speed-ups could be obtained by using the placement of a new leaf as an opportunity to graft a larger piece of an input tree onto the supertree. For example, if the position of leaf X is finalised by quartet XA/BC drawn from input tree T, the subtree of T at the node defined by X, A and either of B or C that includes leaf X can be grafted to the growing supertree. Note that the method does not demand the starting trees be quartets, they could be the trees inferred using any designed decomposition, and this obviates any concern that quartet trees are difficult to infer accurately because of poor taxon sampling. As with QP, this approach may be sensitive to the starting quartet and to the order in which new leaves are added with the extent of any variation reflecting conflict and/or ineffective overlap.

5. Conclusion

Supertree methods provide ways of combining phylogenetic information in diverse trees. As such they can be used to produce large-scale phylogenies from sets of trees that are culled from the literature or produced anew through mining of genomic data, and they are essential to any more formal divide-and-conquer analysis of single data sets. To date, supertree methods have mostly been used to produce composite phylogenies from previously published trees, but there has been a recent increase in their application to the phylogenetic analysis of genomic data^{49,50}. Molecular data is still available for relatively few genes from relatively few taxa⁵¹ but this is rapidly improving as more complete genomes are sequenced and as “shallow genomics” projects such as EST (Expressed Sequence Tag) surveys⁵² and organelle genome sequences are completed⁵³ and we expect the use of supertree methods to increase along with the available genomic data.

Much has been made of the potential for supertree methods to combine 'data' that are otherwise difficult to combine in a single phylogenetic analysis⁵⁴. Increasingly, however, most phylogenetic work will be based on molecular data that could, in principle, be combined so that this justification for supertree methods will become less important^{55,56}. While there has been a debate between advocates of supertree methods and those who prefer simultaneous analysis of data, we agree with others in not seeing a stark choice between mutually exclusive alternatives^{57,58}. This is perhaps most clear in the use of supertree methods as part of divide-and-conquer approaches to finding best-fitting trees for a given set of data, i.e. to efficiently and accurately perform simultaneous analysis of large datasets.

Whereas we do not know whether supertrees constructed from published trees are particularly accurate, we do know that supertree methods embedded in a rationally designed divide-and-conquer strategy can improve heuristic searches, producing better trees faster^{19,20,59}. We also know that some sort of supertree analysis will be needed to join together disparate parts of the tree of life inferred using different markers. We consider the question of how to best achieve effective overlap to be an extremely important one because good answers have the potential to help us target our future research efforts to build the tree of life as efficiently as possible. Efficient supertree construction also requires polynomial time algorithms. The quartet joining method we have outlined is a very fast method of supertree construction that should work well in the absence of conflict. However its accuracy when confronted with real inference problems is unknown. A priori, one might anticipate some trade off between speed and accuracy, given that speed is achieved partly by considering less evidence. Hence, accuracy might be improved by considering the evidence from multiple relevant quartets should they be

available. We are currently developing an implementation of quartet joining that will allow the performance of the method to be investigated when input trees conflict.

6. References

1. Haldane, J.B.S., *Possible Worlds and other essays*, Chatto and Windus, London, 1927.
2. Cracraft, J. and Donoghue, M.J., *Assembling the Tree of Life*, Oxford University Press, New York, 2004.
3. Soltis, P.S. and Soltis, D.E., Molecular systematics: assembling and using the tree of life, *Taxon*, 50, 663, 2004.
4. Hillis, D.M., Inferring complex phylogenies, *Nature*, 383, 130, 1996.
5. Kim, J., General inconsistency conditions for maximum parsimony: effects of branch lengths and increasing numbers of taxa, *Syst. Biol.*, 45, 363, 1996.
6. Purvis, A. and Quicke, D.L.J., Building phylogenies: are the big easy?, *Trends Ecol. Evol.*, 12, 49, 1997.
7. Pollock, D.D. et al., Increased taxon sampling is advantageous for phylogenetic Inference, *Syst. Biol.*, 51, 664, 2002.
8. Rosenberg, M.S. and Kumar, S., Taxon sampling, bioinformatics and phylogenomics, *Syst. Biol.*, 52, 119, 2003.
9. Wilkinson, M., Missing data and multiple trees: stability and support, *J. Vertebr. Paleontol.*, 23, 311, 2003.
10. Nylander, J.A.A. et al., Bayesian phylogenetic analysis of combined data, *Syst. Biol.*, 53, 47, 2004.
11. Pagel, M. and Meade, A., A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data, *Syst. Biol.*, 53, 571, 2004.

12. Strimmer, K. and von Haeseler, A., Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies, *Mol. Biol. Evol.*, 13, 964, 1996.
13. Cormen, T.H. et al., *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.
14. Garvey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, NY, 1979.
15. Graham, R.L. and Foulds, L.R., Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computation time, *Math. Biosci.*, 60, 133, 1982.
16. Day, W.H.E. and Sankoff, D., Computational complexity of inferring phylogenies from dissimilarity matrices, *Syst. Zool.*, 35, 224, 1986.
17. Day, W.H.E., Computational complexity of inferring phylogenies from dissimilarity matrices, *B. Math. Biol.*, 49, 461, 1987.
18. Addario-Berry, L. et al., Ancestral maximum likelihood of evolutionary trees is hard, *Journal of Bioinformatics Comput. Biol.*, 2, 257, 2004.
19. Nakleh, L. et al., Designing fast converging phylogenetic methods, *Bioinformatics* 17, S190, 2001.
20. Roshan, U. et al., Performance of supertree methods on various data set decompositions, in *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, Bininda-Emonds, O.R.P., Ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004, chap. 15.
21. Semple, C. and Steel, M., A supertree method for rooted trees, *Discrete Appl. Math.*, 105, 147, 2000.
22. Page, R.D.M., Modified mincut supertrees, in *Proceedings of WABI 2002*, Gusfield, D. and Guigó, R., Eds., 2002, 537-551.

23. Huson, D.H., Nettles, S., and Warnow, T., Disk-covering, a fast-converging method for phylogenetic tree reconstruction, *J. Comp. Biol.*, 6, 369, 1999.
24. Eulenstein, O. et al., Performance of flip supertree construction with a heuristic algorithm, *Syst. Biol.*, 53, 299, 2004.
25. Wilkinson, M. et al., The shape of supertrees to come: tree shape related properties of fourteen supertree methods, *Syst. Biol.* 54, 419, 2005.
26. Gordon, A.D., Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labelled leaves, *J. Classif.*, 3, 335, 1986.
27. Price, S.A., Bininda-Emonds, O.R.P., and Gittleman, J.L., A complete phylogeny of the whales, dolphins and even-toed hoofed mammals (Cetartiodactyla), *Biol. Rev.*, 80, 445, 2005.
28. Sanderson, M.J. et al., Obtaining maximal concatenated phylogenetic data sets from large sequence databases, *Mol. Biol. Evol.*, 20, 1036, 2003.
29. Yan, C., Burleigh, J.G., and Sanderson, M.J., Identifying optimal incomplete phylogenetic data sets from sequence databases, *Mol. Phylogenet. Evol.*, 35, 528-535, 2005.
30. Wilkinson, M. et al., Towards a phylogenetic supertree of Platyhelminthes?, in *Interrelationships of the Platyhelminthes*, Littlewood, D.T.J. and Bray, R.A., Eds., Taylor and Francis, London, 2001, pp. 292-301.
31. Faith, D.P., Conservation evaluation and phylogenetic diversity, *Biol. Conserv.*, 61, 1, 1992.
32. Kennedy, M. and Page, R.D.M., Seabird supertrees: combining partial estimates of procellariiform phylogeny, *Auk*, 119, 88-108, 2002.
33. Pisani, D. et al., A genus-level supertree of the Dinosauria, *P. Roy. Soc. B*, 269, 915-921, 2002.

34. Cardillo, M. et al., A species-level phylogenetic supertree of marsupials, *J. Zool.*, 264, 11-31, 2004.
35. Huson, D.H., Vawter, L., and Warnow, T., Solving large scale phylogenetic problems using DCM2, in *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, Lengauer, T. et al., Eds., AAAI Press, Menlo Park, CA, 1999, 118-129.
36. Roshan, U.W. et al., Rec-I-DCM3: a fast algorithmic technique for reconstructing large phylogenetic Trees, in *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, 2004, 98-109.
37. Wilkinson, M. and Thorley, J.L., Efficiency of strict consensus trees, *Syst. Biol.*, 50, 610, 2001.
38. Thorley, J.L., Wilkinson, M., and Charleston, M., The information content of consensus trees, in *Advances in Data Science and Classification*, Rizzi, A., Vichi, M., and Bock, H.H., Eds., Springer-Verlag, Berlin, 1998, 91-98.
39. Pentony, M.M., *Quartet puzzling supertrees*, PhD thesis, National University of Ireland, Maynooth, 2004.
40. Pisani, D. and Wilkinson, M., Matrix representation with parsimony, taxonomic congruence, and total evidence, *Syst. Biol.*, 51, 151, 2002.
41. Wilkinson, M. et al., Some desiderata for liberal supertrees, in *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, Bininda-Emonds, O.R.P., Ed., Kluwer Academic, Dordrecht, The Netherlands, 2004, chap. 11.
42. Strimmer, K., Goldman, N., and von Haeseler, A., Bayesian probabilities and quartet puzzling, *Mol. Biol. Evol.*, 14, 210, 1997.
43. Schmidt, H.A. et al., TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing, *Bioinformatics*, 18, 502, 2002.

44. Vinh, L.S. and von Haeseler, A., IQPNNI: moving fast through tree space and stopping in time, *Mol. Biol. Evol.*, 21, 1565, 2004.
45. Bryant, D., *Building trees, hunting for trees and comparing trees*, PhD thesis, University of Canterbury, 1997.
46. Dekker, M.C.H., *Reconstruction methods for derivation trees*, Masters thesis, Vrije Universiteit, 1986.
47. Wilkinson, M., Cotton, J.A., and Thorley, J.L., The information content of trees and their matrix representations, *Syst. Biol.*, 53, 989, 2004.
48. Steel, M., The complexity of reconstructing trees from qualitative characters and subtrees, *J. Classif.*, 9, 91, 1992.
49. Creevey, C.J. et al., Does a tree-like phylogeny only exist at the tips in the prokaryotes?, *P. Roy. Soc. B*, 271, 2551, 2004.
50. Beiko, R.G., Harlow, T.J., and Ragan, M.A., Highways of gene sharing in prokaryotes, *P. Natl. Acad. Sci. USA*, 102, 14332, 2005.
51. Sanderson, M.J. and Driskell, A.C., The challenge of constructing large phylogenetic trees, *Trends Plant Sci.*, 8, 374, 2003.
52. Theodorides, K. et al., Comparison of EST libraries from seven beetle species: towards a framework for phylogenomics of the Coleoptera, *Insect Mol. Biol.*, 11, 467, 2002.
53. Miya, M., Kawaguchi, A., and Nishida, M., Mitogenomic exploration of higher teleostean phylogenies: a case study for moderate-scale evolutionary genomics with 38 newly determined complete mitochondrial DNA sequences, *Mol. Biol. Evol.*, 18, 1993, 2001.
54. Sanderson, M.J., Purvis, A., and Henze, C., Phylogenetic supertrees: assembling the trees of life, *Trends Ecol. Evol.*, 13, 105, 1998.

55. Rokas, A. et al., Genome-scale approaches to resolving incongruence in molecular phylogenies, *Nature*, 42, 798, 2003.
56. Scotland, R.W., Olmstead, R.G., and Bennett, J.R., Phylogeny Reconstruction: The Role of Morphology, *Syst. Biol.*, 52, 539, 2003.
57. Levausser, C. and Lapointe, F.-J., War and peace in phylogenetics: A rejoinder on total evidence and consensus, *Syst. Biol.*, 50, 881, 2001.
58. Holmes, S., Statistics for phylogenetic trees, *Theor. Popul. Biol.*, 63, 17, 2003.
59. Fuellen, G., Wagele, J.W., and Giegerich, R., Minimum conflict: a divide-and-conquer approach to phylogeny estimation, *Bioinformatics*, 17, 1168, 2001.

Figure 1. Two compatible input trees and their strict consensus supertree from Gordon (1986). Polytomies on the supertree show where there is no effective overlap between the input trees. Dots indicate the seven different positions in which leaf 10 could occur on input tree 1 while the two input trees remain compatible. Letters indicate the same positions for taxa 6, 7 and 8 on tree 2. Sampling leaf 10 for the tree 1 gene would produce a fully resolved supertree if it was placed in any of these positions. The improvement in overlap with sequencing leaves 6,7 or 8 for tree 2 depends on where the taxa appear on this tree.

Figure 2. Three pairs of four-taxon input trees together with their strict consensus supertrees. The phylogenetic position of the two shared leaves has a profound effect on their effectiveness of the overlap between the two trees.

Figure 3. The size of the span of supertrees inferred from two different input trees combining with two larger trees. The two small input trees overlap by two taxa (one of which is varied) with the larger trees, and have one unique leaf (X). The effectiveness of overlap between the two trees (measured by the size of the span) varies greatly with which leaves are shared and with the topology of the two trees.

Figure 4. Choosing new taxa for optimal supertree construction. Overlap might be maximised by sequencing a few model organisms (indicated by grey lines within large radiations) when sequencing a particular marker for a clade of interest (indicated by grey triangle), as many other markers will also be sequenced for these organisms. If some idea of the relationships of the sequenced organisms is known, more effective overlap might be obtained by sequencing closely related outgroups that form sequential sister

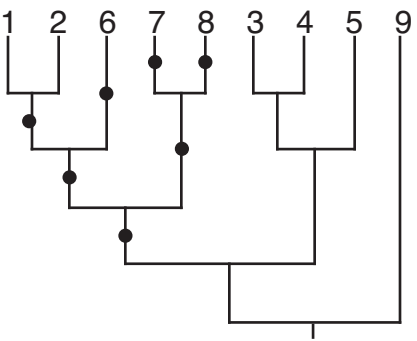
groups to the clade of interest (perhaps taxa A and B would be the best choices here). This suggests that existing taxonomic practice, in which closely related outgroups are chosen to root phylogenies, but care should be taken that the outgroups do not form a monophyletic group to the exclusion of the clade of interest (as taxa B and C would) as this would result in no effective overlap.

Figure 5. Performance of QP in the supertree setting. Twelve quartets (a), the unique tree displaying all 12 quartets (b) and the majority-rule component consensus of trees constructed from the 12 quartets using 1,000 replicates of the voting method of QP (c). Numbers indicate frequencies of occurrence of splits in the QP trees with those for splits entailed by the 12 quartets shown in bold.

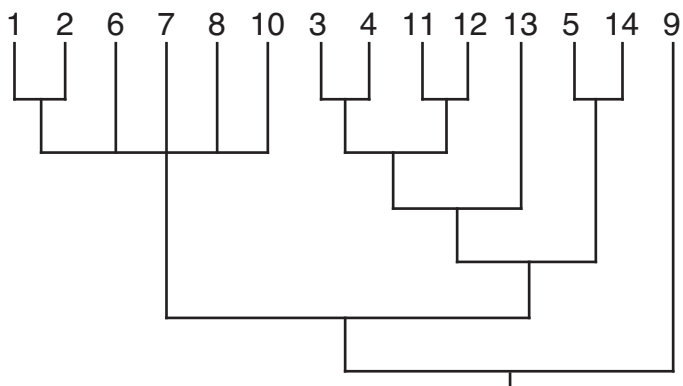
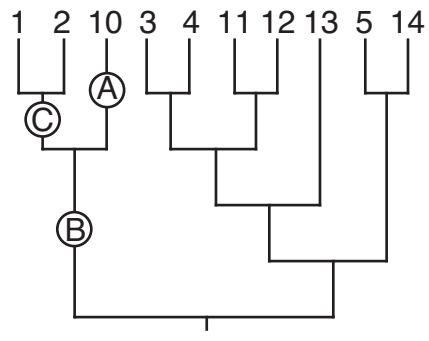
Figure 6. Quartet voting systems. (a) The QP voting system showing the votes cast by two quartets relevant to the placement of E (on the left) on the quartet AB/CD. (b) The Vinh and von Haeseler (2004) voting system showing votes cast by a single quartet relevant to the addition of E to AB/CD. (c) The fully resolved tree these quartets entail.

Figure 7. Quartet joining of the quartet trees shown in figure 5. Choosing a starting quartet at random, a supertree is built up by sequentially adding a single taxon using the information from relevant quartets. For the minimal set of quartets used here, there is always only one relevant quartet and the order in which leaves are selected does not matter, but it will matter in general.

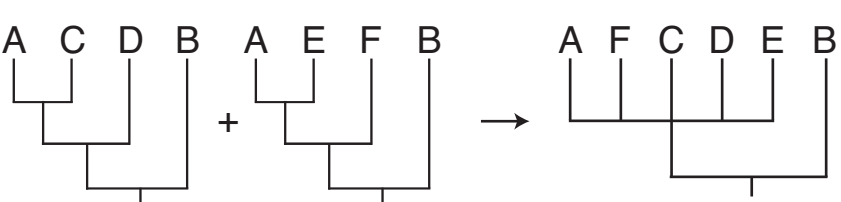
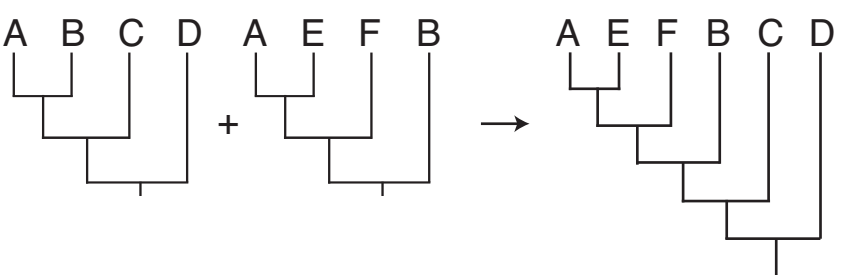
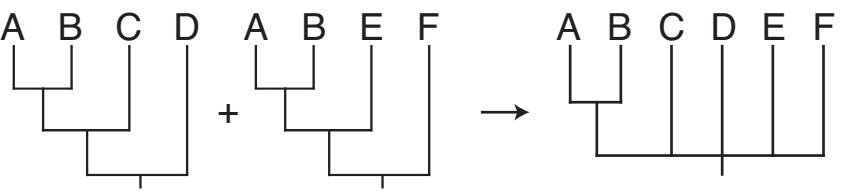
Input tree 1

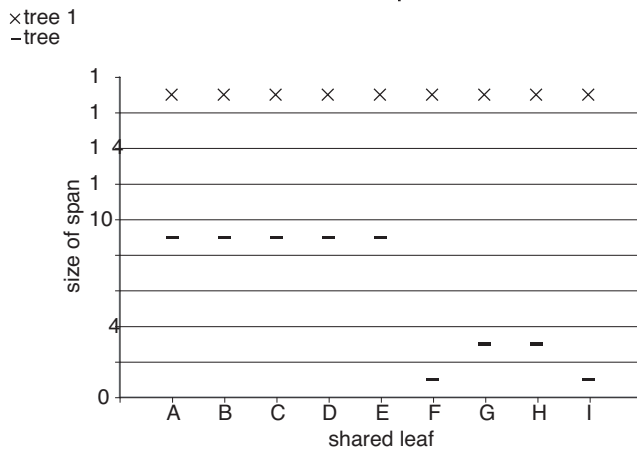
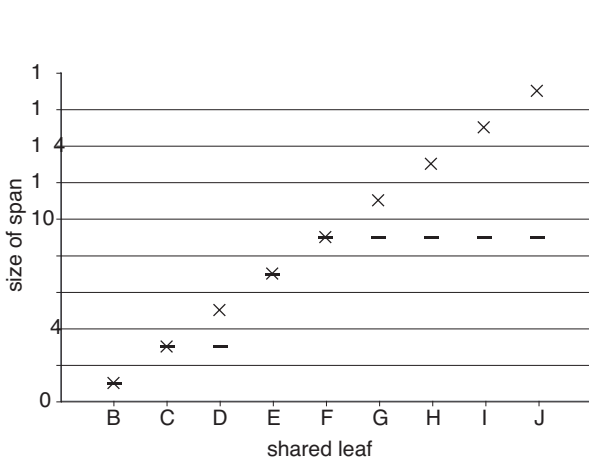
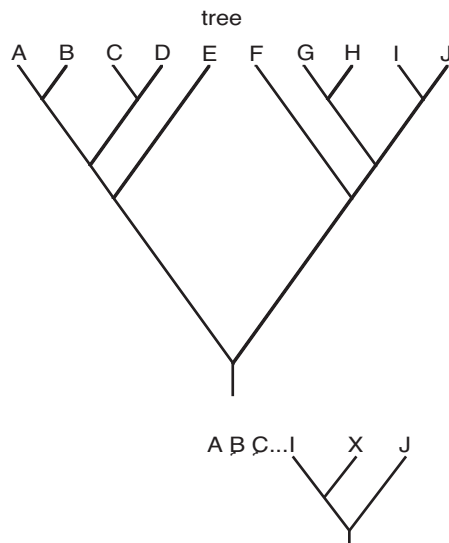
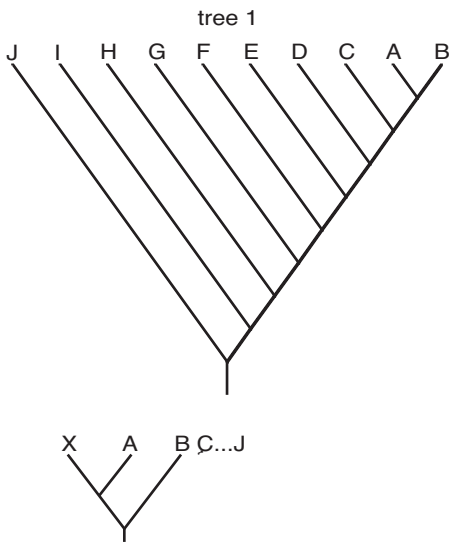


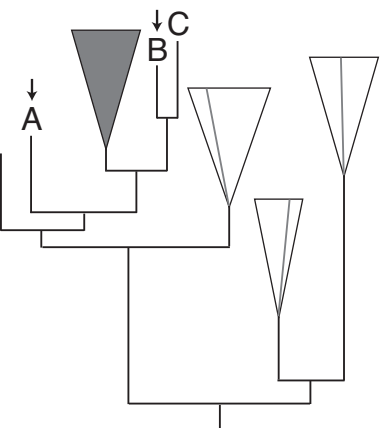
Input tree 2



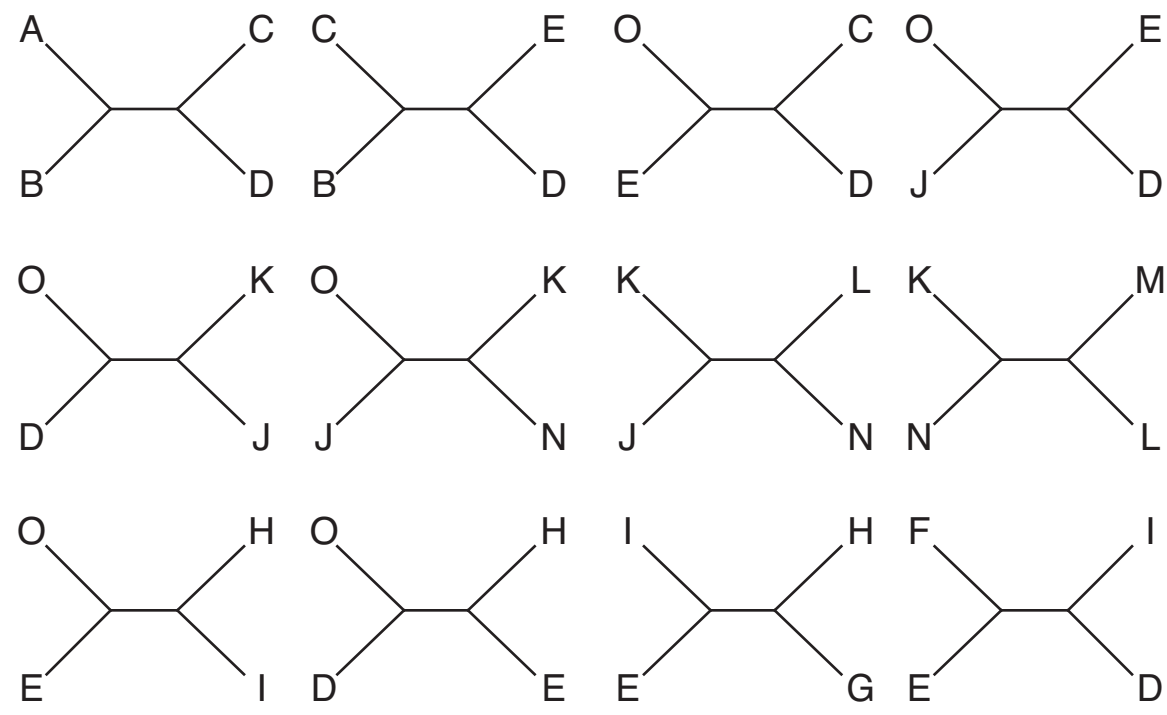
Supertree



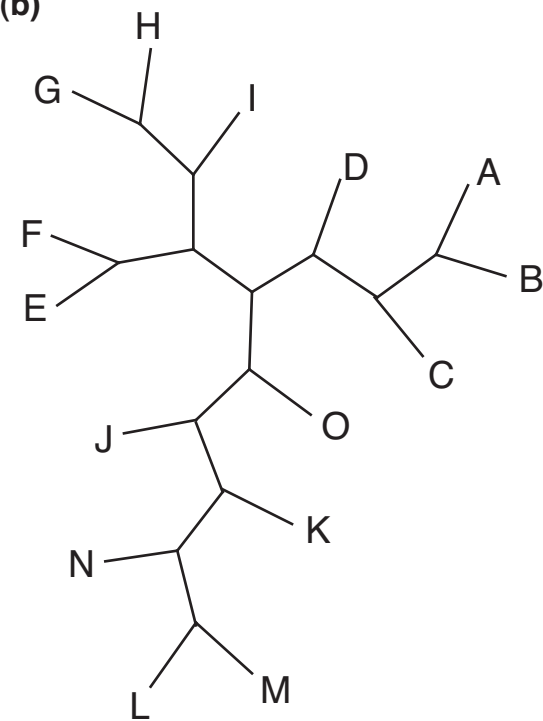




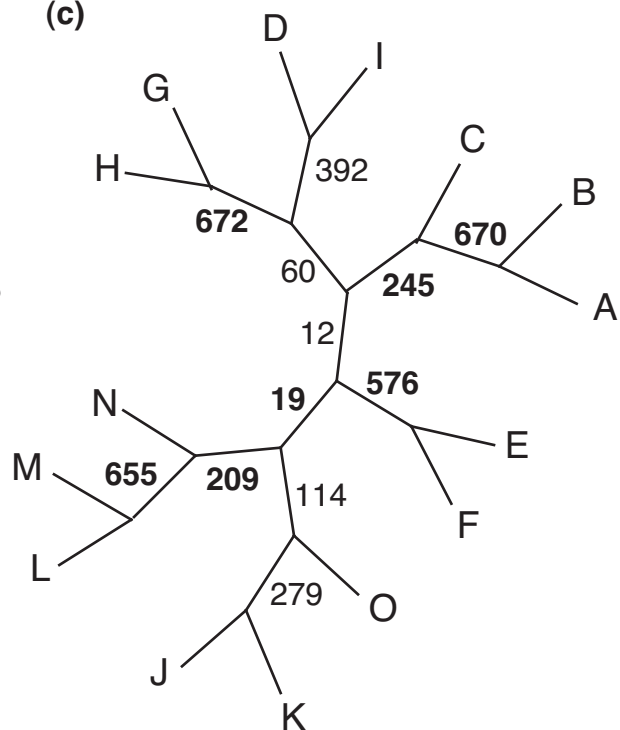
(a)



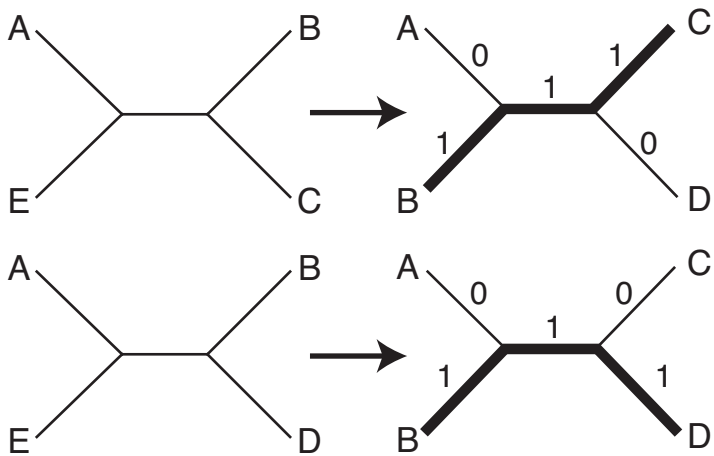
(b)



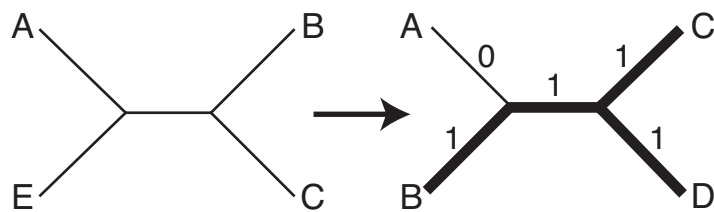
(c)



(a)



(b)



(c)

