1 Phylogeny Reconstruction

Arndt von Haeseler[1], *Heiko A. Schmidt*[2], *Ingo Ebersberger*[1]

[1]Bioinformatics, Düsseldorf University, Germany
[2]Bioinformatics, NIC, FZ-Jülich, Germany

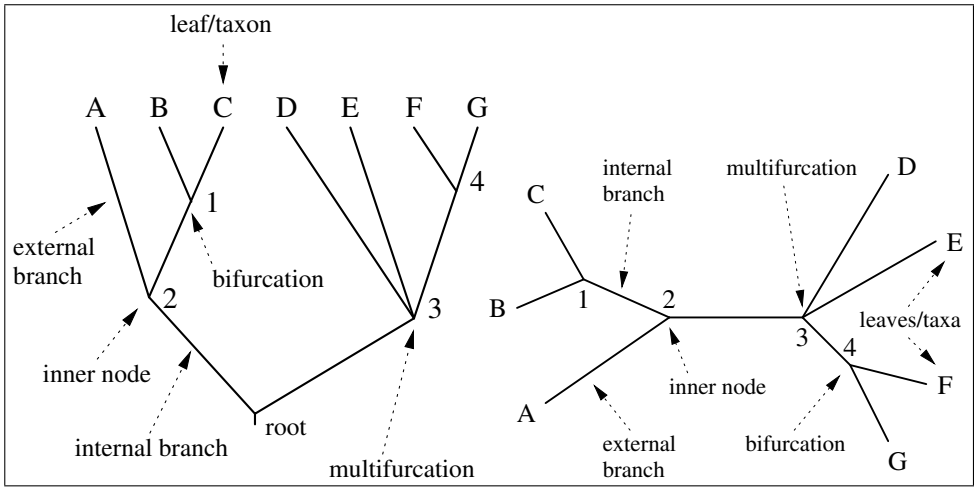
1.1 Introduction

1.1.1 The evolutionary process

Change of characters in time

Diversification and speciation

Tree-thinking in Phylogenetics



1.1.2 Reconstruction of phylogenetic relationships

It is maybe the most fundamental assumption in evolutionary biology that any two species on earth share a common ancestor at a certain time point in their history. One of the major tasks in evolutionary research is therefore the reconstruction of these phylogenetic relationships. Unfortunately, in most cases no direct evidences –such as a comprehensive fossil record– exists about when a particular pair of species split from their most recent common ancestor, nor about the succession of ancestors for more than two species. It remains then as the only option to reconstruct the phylogenetic tree from its leaves, comprised by the species, or more

general by the taxa we observe today.
Tree reconstruction can be divided into two sub-problems: First, what is the topology of the tree, i.e. what is the order the individual taxa branch off? Having inferred the tree topology, we will obtain valuable information about the relative relationships of the species to each other. Obviously, this is invaluable for a meaningful systematic classification concept. Second, what is the evolutionary time that has passed since any two taxa in the reconstructed tree last shared a common ancestor. Finding its reflection in the branch lengths of the tree, evolutionary time can be informative both about the absolute time in years (or generations) two taxa evolved independently and about the extent of evolutionary change that has accumulated on either lineage.

Reconstructing a tree from its leaves

Aims

- Reconstruction of the tree topology (branching order)
- Reconstruction of the tree shape/parameters? (branch lengths)

Means

- Comparison of evolved characters between taxa to trace the evolutionary signal
- Modelling the evolutionary process to dissect the evolutionary signal from noise
- Account for heterogeneous evolutionary signals in the data
- Find the appropriate character class (candidates range from body plan to molecules) as well as the appropriate character whithin the chosen class to reconstruct phylogenies.

Molecular sequence data as the relevant character class

1.2 Modelling DNA sequence evolution

It is well-known that a variety of evolutionary forces act on DNA sequences (see Chapter 1). As a result, sequences change in the course of time. Therefore, any two sequences derived from a common ancestor that evolve independently of each other eventually diverge. The pattern and extent of divergence between the two sequences can then be used to reconstruct their evolutionary history. The substitution of nucleotides or amino acids in a sequence is usually considered a random event. As a consequence, an important prerequisite for the reconstruction of phylogenetic relationships among species is the prior specification of a *model of substitution* which provides a statistical description of this stochastic process. Once a mathematical model of substitution is assumed, then straightforward procedures exist to infer genetic relationships from the data.
To count the number of mutations X(t) that occurred during the time t we introduce the so-called Poisson process: at any point in time an event, i.e. a mutation, can take place. That is to say, per unit of time a mutation occurs with intensity or rate μ. Let $P_n(t)$ denote the

probability that exactly n substitutions occurred during the time t. Then we can say that the number of substitutions up to time t is Poisson-distributed with parameter μt:

$$P_n(t) = [(\mu t)^n exp(-\mu t)]/n! \tag{1.1}$$

On average, μt substitutions with variance μt are expected. Note that the parameters μ (nucleotide substitutions per site per unit time) and t (the time) are confounded. That is, they cannot be estimated separately but only through their product μt (number of substitutions per site up to time t.

The nucleotide substitution process of DNA sequences described by the Poisson process can be generalized to a so-called Markov process that uses a Q matrix –that is a matrix, which specifies the realtive rates of change of each nucleotide along the sequence. The most general form of the Q matrix is shown in figure 1.1. Rows follow the order A, C, G and T so that, for example, the second term of the first row is the instantaneous rate of change from base A to base C. This rate is given by te frequency of base C (π_C) times a relative rate parameter, describing (in this case) how often the substitution A to C occurrs during evolution with respect to all other possible substitutions. Thus, each non-diagonal entry in the matrix represents the flow from nucleotide i to nucleotide j, while the diagonal elements are chosen to make the sum of each row equal to zero. They represent the the total flow that leaves nucleotide i. Accordingly, we can write the total number of substitutions per unit time (i.e. the total substitution rate μ) as

$$\mu = -\sum_{i=1}^{n} Q_{ii}\pi_i \tag{1.2}$$

Nucleotide substitution models like the one summarized by the Q matrix in figure 1.1 belong to a general class of models known as time-homogenous time-continuous stationary Markov models. When applied to modelling the nucleotide substitution process these modells share the following set of underlying assumptions:

- The rate of change from i to j at any nucleotide position in a sequence is independent of the nucleotide that that occupied this position prior to i (Markov property)
- Substitution rates do not change over time (homogeneity).
- The relative frequencies of A, C, G, and T ($\pi_A, \pi_C, \pi_G, \pi_T$) are at equilibrium (stationarity).

. Obviously, these assumptions are not necessarily biologically plausible. However, they are the consequence of modelling nucleotide substitutions as a stochastic process. As soon as the evolutionary model – and thus the Q matrix – is specified, it is possible to calculate the probabilities of change from any nucleotide to any other during the evolutionary time t, $P(t)$ by computing the matrix exponential:

any newer approaches to mention where these assumptions are relaxed?

$$P(t) = exp^{Qt} \tag{1.3}$$

This equation resembles the equation under the Poisson distribution with the sole difference that the mean substitution rate μ in the exponential is replaced by the Q matrix, specifying

$$Q = \begin{pmatrix} -(a\pi_C + b\pi_G + c\pi_T) & a\pi_C & b\pi_G & c\pi_T \\ g\pi_A & -(g\pi_A + d\pi_G + e\pi_T) & d\pi_G & e\pi_T \\ h\pi_A & i\pi_C & -(h\pi_A + i\pi_C + f\pi_T) & f\pi_T \\ j\pi_A & k\pi_C & l\pi_G & -(j\pi_A + k\pi_C + l\pi_G \end{pmatrix}$$

Figure 1.1: Instantaneous rate matrix Q. Each entry in the matrix represents the instantaneous substitution rate from nucleotide **i** to nucleotide **j** (rows and columns follow the order **A, C, G, T**. a to l are rate parameters describing the relative rate one nucleotide is substituted by any other nucleotide. $\pi_A, \pi_C, \pi_G, \pi_T$ correspond to the nucleotide frequencies. Diagonal elements are chosen such that each row sums up to zero.

the relative rates for each possible $i \rightarrow j$ substitution according to the evolutionary model represented by the Q matrix.

Nucleotide substitution models From the instantaneous substitution rate matrix Q depicted in figure 1.1 various sub-models of the nucleotide substitution process can be obtained. Among these, the so-called time-reversible models are the ones most commonly used. These models assume that for any two nucleotides i and j the rate of change from i to j is the same as from j to i ($a = g, b = h, c = j, d = i, e = l, f = l$ in figure 1.1).If all of the remaining eight free parameters of a reversible substitution rate matrix Q are specified, the general time reversible (GTR) is derived. At the other extreme it can be assumed that the equlibrium frequencies of the four nucleotides are 0.25 each, and that any nucleotide has the same rate to be replaced by any other. These assumptions correspond to a Q matrix with $\pi_A = \pi_C = \pi_G = \pi_T = 1/4$, and $a = b = c = d = e = f = 1$ reflecting the simplest nucleotide substitution model, namely the Jukes and Cantor (JC69) model. An overview of the hierarchy of the most common substitution models is shown in figure XXX.

add figure 4.7 from Strimmer and Haeseler

1.2.1 Modelling rate heterogeneity

The nucleotide substitution models, as we have described them so far, implicitly assume that the rate of nucleotide substitution is the same for any position in the DNA sequence, i.e. rate homogeneity applies. However, it is a well-known fact that this is a due oversimplification. For example, substitutions are about 10 times more frequently accumulated at C and G nucleotides when the C is followed by a G. Similarly, constraints in maintaining functional DNA sequences can result in substitution rates varying along a DNA sequence. To account for such a site-dependent rate variation, a plausible model for distribution of rates over sites is required. Most commonly, a Γ-distribution with expectation 1 and variance $1/\alpha$ is used for this purpose. By adjusting the shape parameter α, the Γ-distribution allows varying degrees of rate heterogeneity. For $\alpha > 1$, the distribution is bell-shaped and models weak rate heterogeneity among sites. For $\alpha < 1$, the Γ-distribution takes on its characteristic L-shape, which describes situations of strong rate heterogeneity, i.e. some positions have very high substitution rates, but most other sites are practically invariable.

$$q_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ differ at two or three nt posititions,} \\ \pi_j & \text{if } i \text{ and } j \text{ differ by one synonymous transversion,} \\ \kappa\pi_j & \text{if } i \text{ and } j \text{ differ by one synonymous transition,} \\ \omega^{(h)}\pi_j & \text{if } i \text{ and } j \text{ differ by one non-synonymous transverion} \\ \omega^{(h)}\kappa\pi_j & \text{if } i \text{ and } j \text{ differ by one non-synonymous transition.} \end{cases}$$

Figure 1.2: Instantaneous rate that codon i at site h is replaced by codon j. pi_j representes the marginal frequency of codon j, κ denotes the transition/transversion rate ratio, and ω the nonsynonymous/synonymous substitution rate ratio.

1.2.2 Codon models

Heterogeneous substitution rates become a particular issue for such DNA sequences that code for proteins. Amino acid sites in a protein sequence are expected to be under different selective constraints, depending on their relevance for the protein function. Accordingly, nucleotide substitutions causing the encoded amino acid to change (replacement mutations) will have probabilities to become fixed in a population varying with the selective constraint imposed on the encoded amino acid position. In contrast, silent substitutional changes – that is the change in the DNA sequence has no effect on the protein sequence level – are invisible to selective forces acting on the protein sequence. As a result, the nonsynonymous/synonymous substitution rate ratio ($\omega = d_N/d_S$) will vary among sites in a DNA sequence, with $\omega = 1$ indicating no selection, $\omega < 1$ purifying selection by removing replacement mutations, and $\omega > 1$ diversifying positive selection/adaptive evolution. Codon models of DNA sequence evolution have been specifically designed to model the evolution of protein coding DNA sequences. An example is shown in figure 1.2 based on a extension of the F84 model (Figure XXX). Note, that in contrast to the conventional substitution models, codon models consider the replacement of one nucleotide triplet (codon) by another. By that, the number of letters in the alphabet of the model increases to sixty four possible codons, rendering codon models computationally highly intensive. Obviously, the assignment of a distinct $\omega^(h)$ to each codon position would lead to a vast over-parameterization of the model. Therefore, distinct ω-classes or statistical distributions –both discrete and continuous– are used to account for heterogeneous ω-values among sites.

1.3 Tracing the evolutionary signal

note that we assume that those taxa are related that share the same character state.

We have briefly outlined above that any two DNA sequences tracing back to a shared ancestor will – mediated by the nucleotid substitution process – eventually diverge. Given a set of related DNA sequence, we can – assuming their genealogy is known – make inferences about the evolutionary forces molding the contemporary DNA sequences from their shared ancestral sequence. Conversely, with an assumption/model at hand of how DNA sequences evolve, we can aim to reconstruct the phylogenetic tree along which the DNA sequences in our data set have evolved. In both cases, however, a representation of the data set is needed, which allows a comparison of such positions in the individual sequences that trace back to a single position in

the common ancestor of all sequences. In an alignment, each of these sequences is represented by a row and is written such that positionally homologous nucleotides form a single column. To account for the insertion and deletion of positions during sequence evolution, gaps can be introduced to achieve positional homology. However, we will not go into details of alignment reconstruction and assume in the following that an alignment for a given set of sequences is available. Based on this alignment several criteria exist to find the tree that best reflects the evolutionary relationships of the sequences in the alignment.

1.3.1 The parsimony principle of evolution

Parsimony methods share as an optimality criterion that among various alternative hypotheses the one, which requires the minimum number of assumptions should be chosen. In the context of DNA sequence evolution this results in a model of Maximum Parsimony proposing that contemporary DNA sequences evolved from their shared ancestor via a minimal number of nucleotide substitutions. Consequently, that sequence tree will be chosen which explains the sequence variability in the corresponding alignment by a minimum number of substitutions.

cite some of them

cite: Sankoff 1975, Sankoff and Rousseau 1975, Sankoff and Cedergren 1983, Swofford et al. Textbook

Generalized Parsimony To date a vast number of modifications of the initial criterion of maximum parsimony exist. Instead of referring to each and every modification separately, we would like to give the idea of the generalized parsimony from which the individual modifications can be easily derived. In a mathematical terminology, one aims to identify those trees in the space of all possible trees which minimize the following equation:

$$L(\tau) = \sum_{k=1}^{B} \sum_{j=1}^{N} \omega_j \times \mathrm{diff}(x_{k'j}, x_{k''j}) \tag{1.4}$$

where $L(\tau)$ id the length of the tree τ , B is the total number of branches in the tree, N is the number of characters analyzed, k' and k'' are two nodes on a branch k having the character states $x_{k'j}$ and $x_{k''J}$. These can be either the observed character states present in the data matrix or in the case of internal nodes the optimal character state assignments. Eventually, $\mathrm{diff}(x, y)$ is the cost for the transition from character state x to state y, and ω_j allows to assign a weight to each character in the tree reconstruction. The transition costs can be summarized in a $m \times m$ matrix S, with m as the number of different character states. The entry S_{ij} then represents the increase in tree length associated with a transition from character state x to y. Depending on the choice of the cost matrix S and of the weight parameter ω_j different extents of prior information about the evolutionary process can be introduced into the parsimony model. In the simplest case, corresponding to no prior knowledge, all characters have equal weights ($\omega_j = 1 for j = 1...N$) and all changes between the character states invoke the same cost (Figure 1.3A). A slightly more specialized model considers that nucleotide substitutions that leave the base type unchanged (Transitions: Purine↔Purine and Pyrimidine$leftrightarrow$Pyrimidine) occur more frequently than nucleotide substitutions altering the type of the base (Transversions: Purin↔Pyrimidine). As a consequence, the phylogenetic signal might decay more quickly into noise for transitional changes than for transversional changes. To account for the resulting difference in the information content, it

A =

	A	C	G	T
A	–	1	1	1
C	1	–	1	1
G	1	1	–	1
T	1	1	1	–

B =

	A	C	G	T
A	–	5	1	5
C	5	–	5	1
G	1	5	–	5
T	5	1	5	–

Figure 1.3: Cost matrices for generalized parsimony. In matrix **A** substitutions between all four nucleotides invoke the same cost. In matrix **B** a slightly more sophisitcated model is presented that lays more weight on transversions in the process of tree reconstruction than on transitions.

might be justified to a assign a smaller weight to transitional changes than to transversional changes in the phylogeny reconstruction procedure. An example for the corresponding S-matrix is shown in figure 1.3B.

Multiple/parallel hits Parsimony principles rely on the assumption that a group of related taxa share a certain character state due to its inheritance from their common ancestor. However, this approximates the true evolutionary events only then in a realistic way when the overall amount of sequence changes is low. Thus, multiple changes of the same character in the same taxon or parallel independent changes of the same character in different taxa – both of which are incompatible with the parsimony principle –, are sufficiently infrequent not to cause any problems. However, when considerably diverged sequences are used for tree reconstruction, or marked substitution rate heterogeneity among sites exists, character inconsistencies can cause severe problems in both assessing the correct number of character changes along the phylogenetic tree and in inferring the correct tree topology.

really incompatible or just requires additional hypotheses such a parallel evolution, reversal etc

1.3.2 Distance based methods

In contrast to parsimony methods with their straightforward biologically interpretable approach to phylogenetic tree reconstruction, other methods chose a mathematical access to accomplish this task. Distance based methods utilize the principle of the least squares. Initially, a distance matrix D is calculated from all pairwise comparisons between the sequences in the data set. Thus, entry D_{ij} represents the pairwise distance between sequences i and sequences j. In a simple approach D_{ij} is computed as the edit distance (Levenstein distance) – that is the minimum number of substitutions required to transform sequence i into j. However, in order to account for superimposed changes occuring at a single sequence position, which have no effect on the edit distance, an appropriate model of sequence evolution can be used to correct the observed distance by estimating the expected number of 'unseen' changes given the model. These corrected distances reflect the number of changes that occurred in two sequences after their emergence from a common ancestor. Once the matrix D has been computed, a phylogenetic tree can be inferred such that the difference between the evolutionary distances d_{ij} of any sequence pair ij in the tree and the corresponding entries in D are minimized. The best tree under this criterion minimizes the following equation:

rephrase

still this requires rephrasing

$$R(T) = \sum_{i<j} (d_{ij} - D_{ij})^2 \tag{1.5}$$

Additive and ultrametric distances Under the hypothesis that biological sequences indeed evolve in a tree-like manner, evolutionary distances have the convenient property of tree additivity. That is, the evolutionary distance between any two sequences is equal to the total length of the branches connecting the two sequences in the phylogenetic tree. Under these conditions the four-point metric condition is satisfied for any four taxa A, B, C, and D connected by the tree show in figure **??** such that

cite Buneman 1971

$$d_{AB} + d_{CD} \leq max(d_{AC} + d_{BD}, d_{AD} + d_{BC}) \tag{1.6}$$

In reality, however, the critertion of exact tree additivity is frequently not met. Firstly, stochastic error in the sampling of phylogenetically informative positions can result in a deviation from perfect tree additivity, even when the underlying data has evolved exactly according to the model used for distance correction. Secondly, the evolutionary model for distance correction might be inappropriate. Lastly, additional evolutionary mechanisms –such as recombination– can have the effect that a sequence does not evolve according to a single evolutionary tree. Obviously, the first two problems can be approached by enlarging the sample size and by chosing more appropriate/realistic sequence evolution models for distance correction. Furthermore, although approaches to infer the phylogenetic tree along which a set of sequences have evolved from thei pair-wise distances are based on the assumption of tree additivity, they allow for a certain degree of deviation from additivity and still obtain the true evolutionary relationships. In contrast, the issue of sequences that do not evolve in a strictly tree like manner remains problematic. We will return to this problem in the section Ñetworks.. The concept of additive distances can be extended to the more constrained scenario of ultrametric distances. For any set of three taxa A, B, and C these can be summarized in the following three-point condition:

cite appropriate papers

be more specific?

$$d_{AC} \leq max(d_{AB}, dBC) \tag{1.7}$$

That is, two of the three genetic distances have to be equal and at least as large as the third distance. From this it can be easily followed that any two taxa in the tree are equally distant from their shared ancestral node. As a consequence, by finding that node in the tree which has the same distance to all terminal taxa, the root of the tree –that is the common ancestor of all taxa in the data set– can identified. In the same way, the evolutionary relationships of the taxa are put in a temporal order. However, for the inference of ultrametric distances the same limitations apply as mentioned for the inference of additive distances. Thus, it is frequently more convenient to refer to alternative concepts to determine the root of a tree (see below).

1.3.3 The criterion of likelihood

The third method of tree reconstruction is based on the principle of Maximum Likelihood (ML) [21] which was introduced into the field of sequence based tree reconstruction by Felsenstein in 1981. The general idea of ML is as simple as it is appealing: For a given model M and the corresponding parameter vector θ the probability or likelihood of observing data D can be calculated. That parameter vector set is then chosen that maximizes the likelihood of observing the data. For the specific problem of reconstructing a phylogenetic tree from biological sequence data, a further parameter τ representing the tree topology is introduced such

cite R.A. Fisher - DONE

that

$$(\tau_{ML} = argmax_{\tau} P(D|\tau, M, \theta). \tag{1.8}$$

Note the subtle but substantial difference to the principle of Maximum Parsimony methods described above. In Maximum Likelihood approaches, a general concept of sequence evolution, namely that one sequence is transformed into another via the smallest number of changes, is replaced by an explicit model of sequence evolution, which specifies the rates with which evolutionary changes occur. From this the most significant advantage of Maximum Likelihood becomes apparent: It allows the incorporation of any model of biological sequence evolution into the tree reconstruction process. By that, it opens access to the full use of statistical approaches to e.g. compare alternative phylogenetic hypotheses, as well as to test fit and robustness of individual models of sequence evolution. A further advantage compared to the previous two approaches is the possibility to make full use of the sequence information, including those sites in the data set where no change is observed.

1.3.4 Calculating the likelihood of a tree

We have described above how to calculate the probability of observing a difference at a given site in two sequences. We now extend this to the question of how to compute the probability of finding a certain nucleotide pattern (D_s) in a column s in a set of n aligned DNA sequences. we should stay with DNA sequences in the whole manuscript Obviously this probability depends on the model of DNA sequence evolution and on the tree relating the n nucleotides in the alignment column: $P(D_s|\tau, M, \theta)$. An efficient way how to efficiently compute this probability has been described by Felsenstein (1981). Superimposing the constraints that all positions in an alignment of length L evolve according to the same evolutionary model but independently from each other, the probability of the alignment given a tree and a model is

$$P(D|\tau, M, \theta) = \prod_{s=1}^{L} P(D_s|\tau, M, \theta) \tag{1.9}$$

To avoid underflow errors during the calculation, probabilities are always $\leq$1, the likelihood of the data is usually calculated in log-scale, such that

$$log[P(D|\tau, M, \theta)] = \sum_{s=1}^{L} log[P(D_s|\tau, M, \theta)] \tag{1.10}$$

This equation allows us to compute the likelihood of an alignment, if we knew the evolutionary model, the topology of the tree connecting the sequences in the alignment as well as its branch lengths. In reality, however, we face the reverse situation. Starting from a given alignment, we aim to infer the underlying phylogenetic tree together with its branch lengths. In order to do so we regard these parameters as variables. Once we have decided for an evolutionary model and have specified its parameter values, we can adjust the tree topology and the branch lengths such that equation 1.10 is maximized. While straightforward and efficient ways exist to obtain Maximum Likelihood branch lengths for a specific tree topology, it is both a computationally as well as intellectually demanding problem to obtain an optimal tree topology. Section "Finding the optimal tree" deals in detail with this problem. rephrase last sentence!

1.3.5 Rooting trees / Molecular clock

So far we have introduced various methods to infer the relative relationships of sequences (or taxa) to each other. In many cases, however, it is the temporal order in which the individual taxa in a tree branch off one is mainly interested in. Unfortunately, most of the methods described above lack an inherent criterion to assign directionality to the evolutionary process. As a consequence they are unable to identify the root –that is the internal node dating farthest back in time– of a phylogenetic tree. To nevertheless arrive at a rooted tree, it is required (and possible) to add supplementary information into the tree reconstruction procedure.

Outgroup rooting Among the various methods to root a tree, it is certainly most intuitive to divide the set of taxa into two subgroups: a monophyletic ingroup and an outgroup, whose more distant relationship to any member of the ingroup is either known or at least reasonable to assume. It is then straightforward to conclude that the node where the outgroup joins the ingroup taxa represents the root of the tree [55, 41]. Although of tempting simplicity, this approach requires some considerations in order to be applied in a meaningful way. Outgroup taxa should be, despite their clear position outside the ingroup, as closely related to the ingroup taxa as possible. This will increase the probability to reliably identify homologous sequence positions using standard alignment procedures. Furthermore, it minimizes the risk that outgroup taxa are represented by little more than randomized, fully saturated sequences with respect to the ingroup [67, 33]. In such cases it is impossible to extract the evolutionary signal from the sequence comparisons, and thus outgroup/root placement lacks support by the data[51]. In addition to these more general requirements, some additional guidelines exist to root phylogenetic trees by the use of an outgroup. First, where applicable more than one taxon should be included into the outgroup [34]. Furthermore, different outgroup taxa should be used to check whether the root placement changes with the chosen outgroup [62]. Finally, hyper-variable sites in the sequences should be identified and excluded from the analysis where applicable since they tend to blur the phylogenetic signal in distantly related sequences [24].

still rephrase

Midpoint rooting and Molecular clock As we have seen, the choice of a meaningful outgroup to root a phylogenetic tree can extend to an unsolvable problem, especially when analyzing groups whose phylogenetic relationship is largely unclear. In such cases additional assumptions about the evolutionary process can be imposed that help to root the tree without any prior knowledge about the phylogenetic relationships of the taxa under study.
Phylogenetic trees can be rooted under the assumption of a molecular clock. That is, per unit of time any lineage accumulates the same amount of sequence changes. In other words, the branch lengths in the tree represent the evolutionary time since two taxa split from their common ancestor. Under these conditions, the point in the tree that is equally distant from all terminal taxa will be assigned as the root. In reality, however, the assumption of a molecular clock is frequently violated. If this is neglected, rooting under the clock assumption will tend to place the root on that part of the tree that is evolving with a higher evolutionary rate. Midpoint rooting relaxes the constraints imposed by the molecular clock assumption slightly. It places the root on the midpoint of the path connecting the two most distantly related taxa in the phylogenetic tree. Compared to the molecular clock assumption this retains only the constraint that the evolutionary rate has to be the same on the two most divergent lineages in

the data set. However, when this criterion is met midpoint rooting identifies the localization of the root correctly [61].

1.4 Finding the optimal tree

Up to now we dealt mainly with the principles to evaluate a phylogenetic tree under a certain evolutionary model. However, people are usually only moderatly interested in a statement how likely a particular tree reflects the true evolutionary relationships among a set of taxa. Rather, they require the tree that most likely reflects these relationships. We can differentiate between two general concepts to search the tree space, that is the set of all possible trees for a set of n taxa, for the desired optimal tree(s): 1) exact searches which guarantee the identification of the optimal tree, and 2) the computationally less demanding heuristic searches which, however, not necessarily obtain the optimal tree.

maybe replace *most likely* with *best*?

1.4.1 Exact searches

In the conceptually simplest approach, the exaustive search, each and every possible bifurcating tree in the tree space is evaluated under the selected optimality criterion. The identification of the optimal tree is then straightforward and the computationally challenge in this search limits to an algorithm that assures a walk through the entire tree space. To accomplish this, one usually starts with a selection of any three taxa from the data set since they can be connected only by a single unrooted tree. Subsequently, the remaining taxa are added in a step-wise fashion such that the ith taxon is added separately to each of the $2i - 5$ branches of every possible tree for the $i - 1$ previous taxa. Going through this approach on a piece of paper with a set of five taxa immediately clarifies the procedure. Obviously, the addition of every taxon increases the number of possible trees by the number of branches the new taxon can be connected to. Thus, the total number of unrooted trees for a set of n taxa calculates as:

$$B(n) = \prod_{k=3}^{n}(2k-5) = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$

[20] The limitations of the exaustive search are perspicuous. Already a compilation of 20 taxa, a data set that is nowadays easily exceeded, requires the evaluation of over 2×10^{20} different trees. Not much imagination is needed to comprehend that this is –and presumably will remain– computationally infeasible.
However, an alternative approach exists that assures a globally optimal solution to the problem of tree search without the need to evaluate every possbile tree for a set of taxa. Branch-and-Bound methods perform an exact but guided search in the tree space omitting those subspaces where the optimal tree cannot be contained in [redacted]. The rationale is simple and depends only on the requirement that the criterion of tree evaluation, e.g. total tree length, is non-decreasing with the addition of new taxa to a particular subtree. For the following we assume that we aim to minimize the value L of this evaluation criterion. We start with the definition of an upper bound for L. This can be obtained for example by evaluating any arbitrary n-taxon tree as a

cite Hendy and Penny 1982

reference. Subsequently, using again a three taxon tree as a primer we recursively reconstruct the possible n-taxon trees. However, as we move on in our reconstruction procedure, i.e. with the addition of more and more taxa into the trees, we compare L of the resulting subtrees with L_upper. As soon as $L_{subtree}$ exceeds the predefined upper limit of L we can be certain that our combined reconstruction/search path leads to a subspace where no tree equally good or better than the reference tree can be found. Thus, no further reconstruction is required and a new path has to be followed up. Alternatively, if we end up with a tree including all taxa that is equally good or better than the reference tree, we will store this as a candidate and update L_{upper} to the new value. Meanwhile a number of improvements have been added to increase the efficiency of Branch-and-Bound methods in the tree search process ▬. These refinements are mainly designed to facilitate earlier cut-offs in the tree search. They include methods to obtain a near optimal tree for an assessment of the initial upper bound, as well as considerations in the order taxa are added to the subtrees; e.g. to add divergent taxa first, thereby increasing the length of the initial subtrees.

cite Hendy and Penny 1982 and Swofford 1996

Despite these improvements exact searches eventually run into computational problems when data sets become large. In these cases, faster heuristic methods are required for tree reconstruction.

1.4.2 Heuristic methods

Summarizing heuristic methods for tree reconstruction with the attribute 'quick and dirty' is not entirely inappropriate. The jettison of a guaranteed globally optimal solution to the tree search problem –therefore 'dirty'– earns a substantial speed up in computating time. With contemporary software it is nowadays possible to reconstruct trees from data sets of more than thousand taxa, e.g. [**?**]. From the Bioinformaticists point of view, we are therefore for the first time in the pleasant situation that biological datasets hardly ever reach the computational limits of tree reconstruction software.

Hill-climbing and the problem of local optimization

The problem of finding an optimal tree for a set of taxa can be illustrated by the metaphor of a night time hiker aiming to reach the point with the highest altitude in a hilly area. Due to the poor visibility, the highest peak cannot be identified a priori. Thus, the hiker remains with the only option to climb any slope he encounters first until he has reached its top. Up there he checks his altimeter and is either confident to have reached one of the highest points in this area and finishes his search, or he invests more effort and climbs another hill. The analogy to the tree search problem is obvious. Starting off with any tree we modify it in a stepwise fashion usually accepting only such modifications that lead, according to the chosen optimality criterion, to an improved tree. At a certain point no further improvement is possible, and thus we have reached the top of the hill. At this point of the search, however, we have no means to decide whether we have found the globally optimal tree, or merely a local optimum. Thus, these kinds of tree searches have to cope with three challenges: first, the identification of a reasonable tree to start the search with, second the implementation of a stepwise hill climbing algorithm for the tree search, and third the avoidance of local optima.

Reasonable starting trees are usually quickly obtained via so called "greedy" strategies. Here,

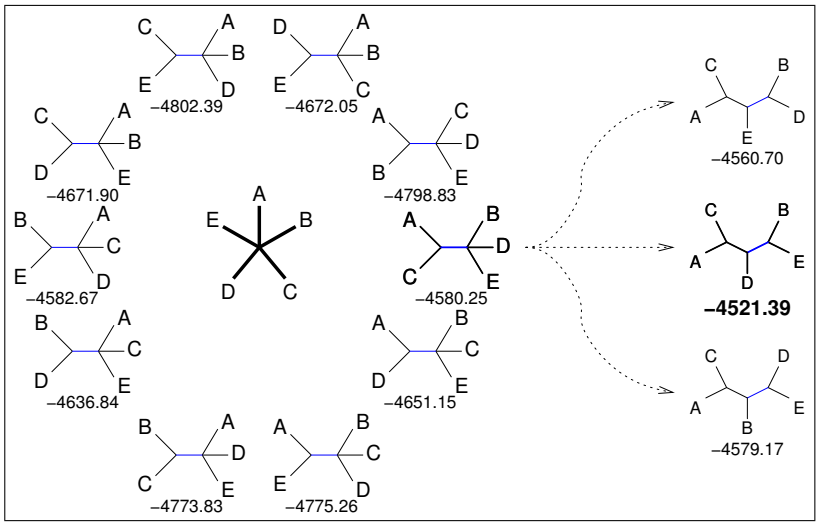


Figure 1.4: Star decomposition

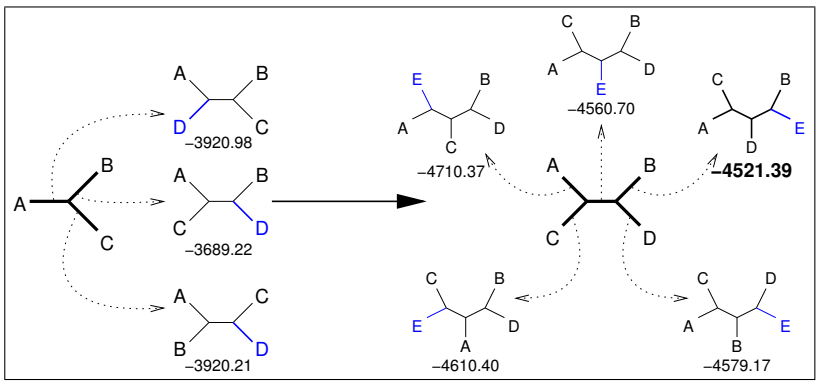


Figure 1.5: Stepwise insertion

the problem –i.e. finding an optimal tree– is divided into several subproblems. These are then sequencially solved by choosing always that solution that looks best given the current situation.

In star decomposition methods (Figure 1.4), e.g. Saitou and Nei's Neighbor Joining algorithm [**?**], we begin with an assignment of all taxa in the data set to the terminal nodes of a star like tree. Subsequently, all trees are evaluated that can be obtained by joining any two of the terminal taxa into a new group. The tree that scores best under the chosen optimality criterion forms then the basis for the next step. The iteration of pairwise joining and tree evaluation continues until the tree is fully resolved, that is an optimal binary tree is obtained.

Alternatively, we can directly construct a binary tree from scratch by inserting the taxa into a tree in a stepwise fashion (Figure 1.5). First, a set of three taxa is used to form a unique

Any suggestions for citation. E.g. Farris 1970?

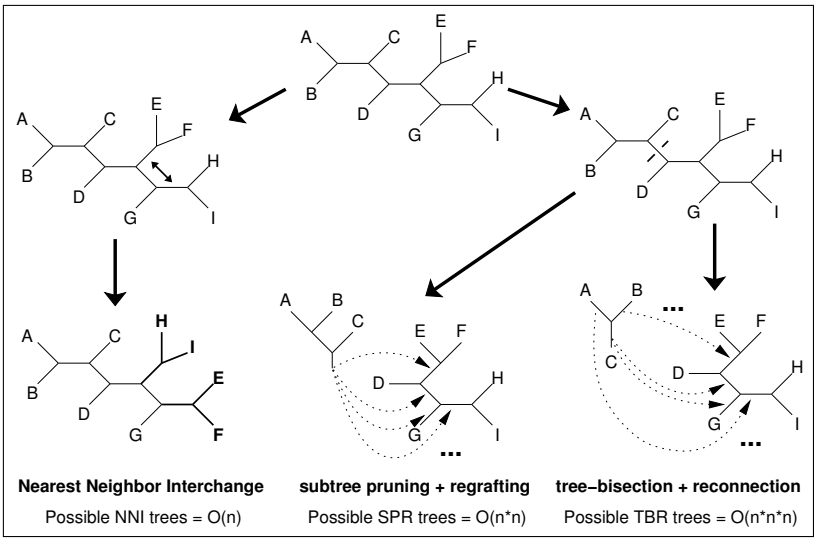


Figure 1.6: Three methods to accomplish branch swapping

binary tree. Next, a fourth taxon is chosen for insertion into the intial tree. Since the this can be attached on any of the three branches of the initial tree, we have three possible topologies for the four-species tree. All of these will be evaluated and the best tree will be stored for insertion of the fifth taxon. The iteration continues until the tree includes all taxa in the data set.

It is straightforward to see why both star decomposition methods as well as the stepwise insertion procedure are prone to obtain only locally optimal trees. Once a decision has been made about the position of a taxon in the tree, this decision is fix for the remaining part of the reconstruction procedure. However, to increase the flexibility, and by that to escape from local optima, tree-rearrangement methods exist to secondary override the strict dependency on previous decisions. In brief, the initial 'optimal' tree is modified such that a part of the tree is excised and re-inserted at a different position. The trees resulting from such 'branch swapps' are evaluated and subjected to one or more acceptance criteria. While a better tree will be always accepted, trees inferior to the one already obtained can be accepted under certain conditions. This deviation from the strict hill climbing approach facilitates the transition to better trees that are more than one rearrangement apart from the currently best tree. Currently, three alternative ways of branch swapping are in use (Figure 1.6). Nearest Neighbor Interchange, the computationally least demanding approach ($O(n)$), takes any internal branch of the tree and swaps two of the four connected subtrees. Note that only swapping of two subtrees located on the opposite sides of the internal branch leads to the formation of a new tree! Subtree pruning and regrafting ($O(n^2)$) excises a subtree and regrafts it with the cut surface at any branch on the tree. Tree-bisection and reconnection is the most complex way to swap branches ($O(n^3)$). The tree is bisected along an internal branch and the resulting subtrees are rejoined at any pair of branches.

As noted, any of the branch swapping methods is capable to guide the tree reconstruction pro-

cedure out of a local optimum. However, no guarantee is granted that it does not simply lead into the next local optimum, in which the benefit would is only limited. Apparently, if the branch swapping is performed only sufficiently long it becomes likely that sooner or later the global optimum will be found. However, two problems are associated with this: First, how shall we recognize the globally optimal tree, and second how long do we have to continue the tree search?

When (better) trees go extinct It is inherent in the heuristic approach that no matter how long we search, we can never be sure that we have found the globally optimal tree. Thus, we need an idea whether the tree we are looking at is just good and we can expect to find a better one soon, or whether it is already 'pretty good' and it will take a fortune to find a better one (although it might exist...). Generally, it is entirely up to the endurance of the researcher how long he allows his program to search for the best tree. Either a predefined number of optimization steps or a lower limit by which new trees improve led to a truncation of the search. Both criteria are arbitrary and a more stalwart basis would be desireable to decide whether to continue or end the tree search.
Recently, a method implemented into the software IQPNNI [64] was suggested that is based on the time of occurrence (i.e. number of iterations) of better trees during the search. Let $L_1, L_2, \ldots, L_j$ denote the log-likelihoods for the first j iterations, then the sequence $\tau(k)$ of record times (i.e. iteration number, when a better tree is found) is defined by

Replace log-likelihoods with a more general word

$$\tau(1) = 1,\ \tau(k+1) = \min\{j | L_j > L_{\tau(k)}\}.$$

This sequence is used to estimate the point in time, τ_{stop}, at which to stop the search, i.e., when it appears unlikely that a further search will lead to a better tree. Using the theory detailed in [14] and [47], one can estimate during the run of the tree search an upper 95% confidence limit $\tau_{95\%}$ of τ_{stop} . Once $\tau_{95\%}$ iterations have been carried out and a better tree was not detected the program will stop and output the best tree found. It can then be concluded that with a probability of 95% no better tree will be found during this search. On the other hand, if a better tree is found before $\tau_{95\%}$ is hit, the $\tau_{95\%}$ is recomputed on the basis of the new record time added to the sequence $\tau(k)$.

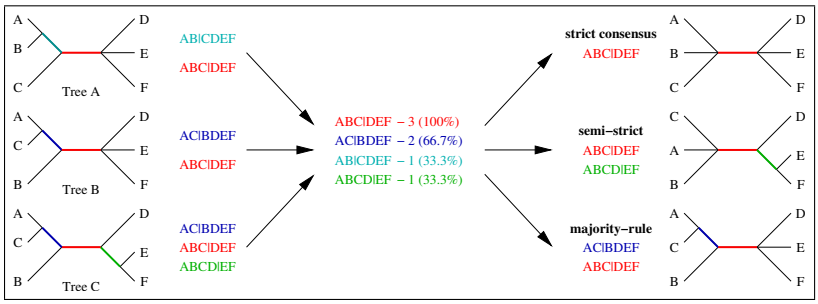
- Heuristics for large data sets (Vinh)
  - RAxML [57]
  - IQPNNI [64]
  - PhyNav [63]
  - PHYML [23]
  - MetaPIGA [32]
- Parallel computing (Heiko)
  - message passing (MPI [22, 56]), shared memory (OpenMP [16])
  - fastDNAml [58, 42]
  - RAxML [57]

- TREE-PUZZLE [53, 52]
- DPRml [31]
- parallel GAML [9, 69]
- TrExML [68, 69]
- maybe parallel IQPNNI

1.5 The advent of Phylogenomics (Heiko)

1.5.1 Multi-locus data sets

- Consensus tree vs. concatenation



- Incomplete sampling

1.5.2 Supertree methods

- Direct supertrees
- Indirect supertrees
- Medium level combinations

1.5.3 Introduction

The large amounts of molecular sequence data currently available serve two needs in the phylogenetic analysis of the relationship of species. On the one hand, the number of interesting species available for analysis grows (vertical growth). On the other hand, more different sequences per species get available (horizontal growth). Unfortunately, the number of sequences is not distributed evenly among species of interest. This often makes collecting a dataset for phylogenetic analysis a painful decision on the trade-off between the amount of taxa and the number of sequences.

While methods abound to infer phylogenies from a set of aligned sequences [61], only a small number of methods exists to combine data of different genes, proteins, or genomic areas for joint analysis.

There has been a large ongoing debate how to combine different datasets to reconstruct trees (cf. [17], [8], and [44], chap. 8 for review). Two paradigms have been discussed, which we will classify by the 'distance' of the combination event from the underlying data into *low level* and *high level* methods (cf. Fig. 1.7). Note that *low* and *high* does not imply a quality rating.

The first paradigm is *total evidence* (also often called *combined* or *simultaneous analysis*) where the data is combined directly by concatenating the alignments. Hence, *total evidence* methods will be referred to as *low level methods* (cf. Fig. 1.7).

The other paradigm is the so-called *separate analysis* (also called *taxonomic congruence* and *consensus* or *supertree approach*). Such methods combine trees reconstructed separately from the single datasets. Since the combination takes place far from the underlying data, such methods are referred to as *high level methods* (cf. Fig. 1.7).

Both paradigms have their advantages and drawbacks (for review see [17], [8], and references therein). Major criticisms are, e.g., the problems to jointly model the evolutionary process for the concatenated dataset in *low level methods* and the loss of information in the *high level methods* since the underlying data is in general not considered when the trees are combined. These arguments gave rise to the design of an alternative method which will be proposed in this chapter.

The method presented here follows a *medium level* paradigm, according to the level of combination (cf. Fig. 1.7). Before particularizing the procedure (section 1.5.5), commonly used *high* and *low level methods* are described (section 1.5.4). Then the new method is applied to a biological dataset in section **??**. The results are discussed in the context of *total evidence* and *consensus/supertree methods*. Finally, problems will be discussed and an outlook on further improvement and extensions will be given.

1.5.4 Methods to Combine Datasets

Low Level Combination: Total Evidence

As mentioned above, *low level methods* are commonly referred to as *total evidence*, *combined* or *simultaneous analysis* methods. Claiming that all information (evidence) available should be used for phylogenetic analysis, all single datasets are combined into one single *'supermatix'* [50] as shown in Fig. 1.7. This is achieved by concatenating all source alignments filling missing data with gap characters. The overall alignment is then used as input for phylogenetic analysis. This method seems unproblematic for complete datasets, where for each species one sequence in each source dataset exists. How *total evidence* approaches handle missing data crucially depends on the method used to reconstruct a phylogeny from the concatenated alignment. Since some programs exclude alignment columns with gaps, such programs will end up only with data from genes which are available for the whole set of species. Datasets with missing sequences are discarded.

High Level Methods

While *low level methods* combine the source datasets directly, *high level methods* construct a tree for each source dataset. The trees from the source datasets are then combined to an overall

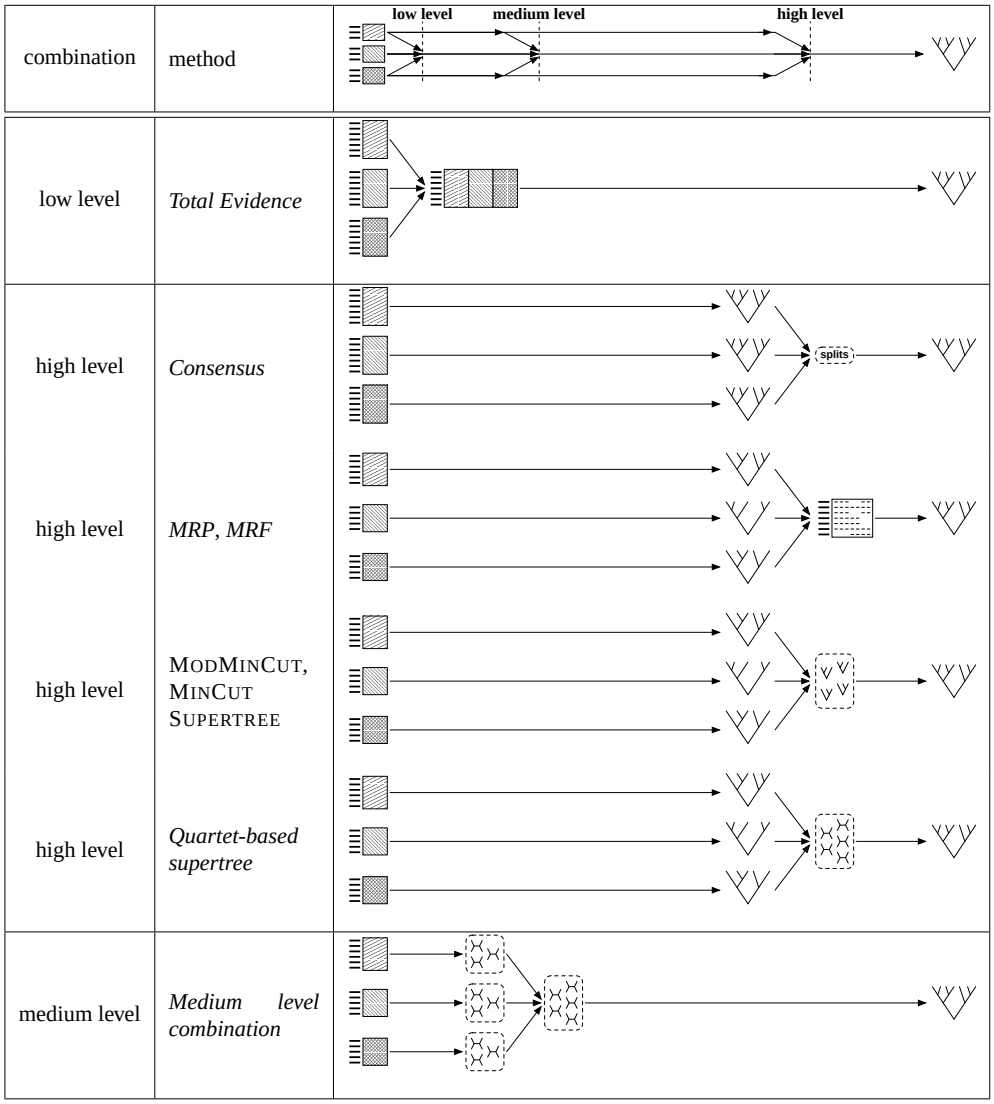


Figure 1.7: Level of combination with regard to distance from the underlying datasets

tree (Fig. 1.7). If all source datasets contain the full set of species, consensus techniques can be applied combining all the equally sized trees into one consensus tree. To combine sets of trees with unequal, but overlapping sets of species, so-called *supertree* methods have been developed which amalgamate the input trees into one overall *supertree*.

Combining Equal-Sized Datasets: Consensus Methods Consensus methods aim to construct a consensus tree from a set of trees preserving common topological details of the source

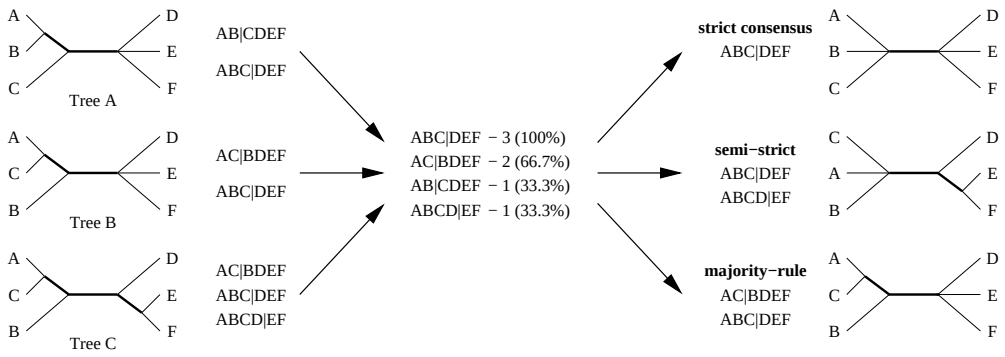


Figure 1.8: Consensus methods

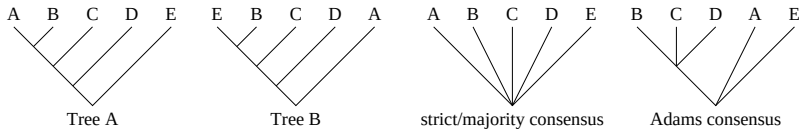


Figure 1.9: Adams Consensus

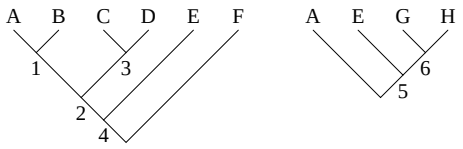
trees. Commonly applied consensus methods are strict consensus, majority-rule consensus [36], semi-strict consensus [10], and Adams consensus [1].

Most consensus methods are based on splits or bipartitions (cf. **??**) found in a majority of the source trees (M_l consensus *sensu* [38], where l represents the percent occurrence of included splits). Sets of non-contradicting bipartitions are chosen to reconstruct the consensus tree.

The *majority-rule consensus* [36] uses all splits occurring in the majority of input trees. Usually a *majority-rule consensus* is assumed (M_l consensus with $l > 0.5$, [38]). This means all splits occurring in more than 50% of the source trees (see Fig. 1.8). Considering $l > 0.5$ ensures that all splits can be incorporated in a tree topology, because no two contradicting splits are able to occur in more than 50% of the input trees.

For the *strict consensus* tree only bipartitions are chosen that are found in all input trees ($M_{1.0}$, see Fig. 1.8). The *semi-strict consensus* [10] contains all splits that are not contradicted in any of the input trees, e.g. the split $ABCD|EF$ is uncontradicted by the ABC-trifurcation in tree A and B in Fig. 1.8. Note that such splits can occur in less than half of the source trees. If all trees are binary trees *strict* and *semi-strict consensus* will always produce the same trees.

Contrary to most consensus methods which are based on bipartitions and their percent occurrence in the source trees, the *Adams consensus* is based on common nestings in trees, i.e., taxa frequently occurring together in the same subtree. This method tries to find groups of sequences that are commonly occurring together in the source trees [1]. *Adams consensus trees* can contain groups that cannot be found in any of the source trees. This makes it difficult to interpret. Yet it can be informative when there are sequences that are difficult to place. Such sequences are moved to the root of the *Adams consensus tree* (Fig. 1.9).



Ragan/Baum

	1	2	3	4	5	6
A	1	1	0	1	0	0
B	1	1	0	1	?	?
C	0	1	1	1	?	?
D	0	1	1	1	?	?
E	0	0	0	1	1	0
F	0	0	0	0	?	?
G	?	?	?	?	1	1
H	?	?	?	?	1	1

Purvis

	1	2	3	4	5	6
A	1	1	0	1	0	**?**
B	1	1	0	1	?	?
C	0	1	1	1	?	?
D	0	1	1	1	?	?
E	**?**	0	**?**	1	1	0
F	**?**	**?**	**?**	0	?	?
G	?	?	?	?	1	1
H	?	?	?	?	1	1

Figure 1.10: Coding schemes to encode tree topologies for MRP methods: Baum/Ragan and Purvis' scheme

Combining Overlapping Datasets: Supertree Methods Supertree methods have been developed to combine sets of overlapping trees into *'supertrees'* containing all leaves found in the source trees. Some of the methods are restricted to rooted trees, based on the argument that the broad majority of trees published are rooted trees. Input trees cannot be produced from different datasets only, but can also be obtained from the literature.

Matrix Representation Methods

A commonly used method to construct supertrees is *Matrix Representation using Parsimony* (MRP, [6, 46]), which uses binary coding to represent the input trees. The splits induced by all source trees are coded into a matrix with binary characters ('0', '1') with missing data ('?'). The binary matrix is then used to construct an overall tree (cf. Fig. 1.7) using *Maximum Parsimony* methods [61]. The most abundant coding schemes are those independently suggested by [6] and [46] and the modified scheme by [45]. Baum/Ragan code the bipartitions of an input tree assigning '1' to all leaves in the one part of the bipartition and '0' to the other that contains the root. All missing taxa are assigned the missing data character '?' (Fig. 1.10). [45] differs in that only the sistergroup of a clade is assigned '0', while all other leaves of that part outside the clade are assigned '?' (Fig. 1.10).

To give input trees different weights in the analysis, different weighting schemes have been suggested (see [49], [50], and references herein for more details).

Recently, another method has been proposed called *Matrix Representation using Flipping* (MRF, [13]), which also uses a binary matrix coding of the trees as described above. The MRF supertree is constructed by 'flipping' conflicting character states from '1' to '0', or vice versa to produce a matrix that does not contain contradictions. Their optimal solution is the tree that needs the least 'flips'.

Direct Supertree Methods

Another family of supertree methods use a more graph theoretical approach. Members

of that family are the ONETREE/BUILD algorithm [2, 12], MINCUT SUPERTREE algorithm [54], and the modified MINCUT SUPERTREE algorithm (MODMINCUT SUPERTREE, [43]). These algorithms take as input sets of rooted trees. The BUILD algorithm and the ONETREE algorithm produce a supertree only if the input trees are compatible, i.e., the trees are not contradictory (cf. strict consensus in 1.5.4). Since this is almost never the case for biological data, these algorithms are mainly useful to test for compatibility of trees.

MINCUT SUPERTREE tries to yield a supertree applying a minimum cut algorithm to the graph $S_{\mathcal{T}}$ constructed from the set of input trees $\mathcal{T}$, following Alg. **??** (see also Fig. **??** and Fig. 1.7). The notations follow [43] – for further details see there.

[43] suggested a modification (here called MODMINCUT) to choose the minimum cuts, that takes into account preserving uncontradicted information from the source trees. The MODMINCUT SUPERTREE algorithm marks all edges in the graph $S_{\mathcal{T}}$ (step **??**) not contradicted by any tree in $\mathcal{T}$. If $S_{\mathcal{T}}$ has to be cut in step **??** minimum cuts are preferred which do not cut uncontradicted edges. Hence, the amount of uncontradicted information from the source trees is preserved in the supertree (cf. Fig. **??**).

Quartet-Based Supertree Recently a quartet-based supertree method has been proposed by [48]. Instead of using a matrix representation, all source trees are decomposed into sets of quartets contained in the topologies (Fig. 1.7). Each quartet is weighted by the highest support value found on the path of the quartet's middle edge in any of the input trees (the authors used the TREE-PUZZLE package to compute the trees and support values). From this set of weighted quartets an overall tree is constructed with the quartet method implemented in the AllTree program [7]. In a first step, this method computes all subsets of taxa and keeps those satisfying the most quartets. In a second step, it performs an exact search on each subset constructing an overall tree satisfying most quartets.

1.5.5 *Medium Level* Combined Phylogenetic Analysis

Notation

In the present chapter we assume a collection $\mathcal{S} = \{s_1, \ldots, s_n\}$ of n species and k different genes to be used for combined analysis.

With $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ we denote subsets of $\mathcal{S}$, such that

$$|\mathcal{S}_g| \leq 4 \qquad \text{for each} \quad g \qquad \text{and} \qquad \bigcup_{g=1}^{k} \mathcal{S}_g = \mathcal{S}. \tag{1.11}$$

Each $\mathcal{S}_g$ represents the subset of species for which a multiple alignment based on gene g is available. These subsets will be called genesets. Finally, $\mathcal{Q}_g, g = 1, \ldots, k$ represent the corresponding sets of possible quartets.

Instead of reconstructing the trees $T(\mathcal{S}_g)$ for each set $\mathcal{S}_g$, we will compute an overall tree $T(\mathcal{S})$ combining the information provided from the evaluation of the quartet sets $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_k$ from the k different alignments. Note that a taxon is not represented by one sequence any more.

The Combined Quartet Method to Combine Genesets

Combining the ML Quartets In the method proposed, the genesets will be combined on the level of the quartets. As a guideline to combine the quartets we use the log-likelihood $\ell^{(g)}_{ab|cd}$, $\ell^{(g)}_{ac|bd}$, and $\ell^{(g)}_{ad|bc}$ for quartet $\{a,b,c,d\} \in \mathcal{Q}_g$ on the geneset $\mathcal{S}_g$.

To combine the datasets for each geneset $\mathcal{S}_1$, $\mathcal{S}_2$, $\ldots \mathcal{S}_k$, all three log-likelihoods for each quartet in $\mathcal{Q}_1$, $\mathcal{Q}_2$, $\ldots \mathcal{Q}_k$ are evaluated first (cf. *ML step* in **??**).

Then, we compute

$$\ell_{ab|cd} = \sum_{g=1}^{k} \ell^{(g)}_{ab|cd} \tag{1.12}$$

$$\ell_{ac|bd} = \sum_{g=1}^{k} \ell^{(g)}_{ac|bd} \tag{1.13}$$

$$\ell_{ad|bc} = \sum_{g=1}^{k} \ell^{(g)}_{ad|bc} \tag{1.14}$$

for each quartet. For the sake of simplicity, the log-likelihoods $\ell^{(g)}_{\tau} = 0$ if the quartet with the sequences $\mathcal{L}(\tau)$ is not represented by alignment g.

Alternatively, one can also compute

$$\overline{\ell_{ab|cd}} = \frac{\ell_{ab|cd}}{|\{g : \{a,b,c,d\} \in \mathcal{Q}_g\}|} \tag{1.15}$$

$$\overline{\ell_{ac|bd}} = \frac{\ell_{ac|bd}}{|\{g : \{a,b,c,d\} \in \mathcal{Q}_g\}|} \tag{1.16}$$

$$\overline{\ell_{ad|bc}} = \frac{\ell_{ad|bc}}{|\{g : \{a,b,c,d\} \in \mathcal{Q}_g\}|}. \tag{1.17}$$

$\overline{\ell_{\tau}}$ can be viewed as the average support a quartet tree τ receives from the set of sequence alignments. In case that a quartet is not represented by any alignment the averages are set equal to zero.

Computing the Overall Tree To reconstruct a phylogeny $T(\mathcal{S})$ based on the collection of log-likelihoods from Equations 1.12 to 1.14 or 1.15 to 1.17 we apply the PUZZLE idea from **??**. The straightforward application of the *QP* algorithm is only possible if information exist for each quartet in $\mathcal{Q}$. This, for instance, is the case if at least one alignment $\mathcal{S}_g$ comprises all the species in $\mathcal{S}$.

If some quartets are missing, which usually happens if each $\mathcal{S}_g$ is a proper subset of $\mathcal{S}$, then these quartets are treated as unresolved. A quartet tree is selected randomly among the three possible topologies. In this case it is necessary to examine whether the overlapping genesets can be combined. This will be explained in the following.

Assessing Whether Genesets Can Be Combined To combine two genesets $\mathcal{S}_i$ and $\mathcal{S}_j$ a minimum pairwise overlap of 3 sequence among the two subsets is required,

$$|\mathcal{S}_i \cap \mathcal{S}_j| \geq 3. \tag{1.18}$$

We call Eq. 1.18 the *pair-overlap condition.* The *pair-overlap condition* is different from the *overlap condition* (Eq. **??**) in section **??** where the subsets were constructed from one complete set of sequences to ensure combinability of the subsets. Here the subsets are genesets. Their sizes and overlap are defined by the availability of sequence data for the species in $\mathcal{S}$. Note, that not every pair of genesets might share a sufficient overlap.

To assess whether these gene datasets can be combined, we construct an overlap graph $G_{ovl} = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V}$ (genesets) and a set of undirected edges $\mathcal{E}$ (overlaps) with

$$\mathcal{V} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_k\} \tag{1.19}$$

$$\mathcal{E} = \{e_{(\mathcal{S}_i, \mathcal{S}_j)} \text{ for } |\{\mathcal{S}_i \cap \mathcal{S}_j\}| \geq 3\} \quad 1 \leq i < j \leq k. \tag{1.20}$$

The edge weights consist of the amount of overlap fulfilling the *pairwise-overlap condition* (see for an example Fig. **??**). This representation of the datasets provides insight into several properties of the entire dataset.

If the graph consists of unconnected subgraphs, not all data is suited for a combined analysis. According to the *pairwise-overlap condition* gene datasets from different connected components cannot be used simultaneously in the same analysis because no quartet information is are available to reasonably guide the insertion of the sequences.

Genesets from one connected component, however, can be used for combined analysis. Although genesets might exist within a connected component which do not share sufficient overlap with each other to be combined, they can be linked by means of other genesets which first have to be added satisfying the *pairwise overlap condition*, to connect the two.

Note, that by applying the *overlap condition* from Eq. **??**, sets from one connected component can gain an overlap ≥ 3 to the leafset $\mathcal{L}(T_i)$ of a tree T_i so far reconstructed from genesets of another connected component. Nevertheless, combining those sets will produce a very doubtful tree, because too little information is available to guide the insertion of sequences.

The combinability of the genesets from a connected component is characterized by two features of the *overlap graph*. A high connectivity within a component (each geneset shows overlap to many other geneset) results in network-like quartet information among the genesets. This reduces the need for mediating sets to connect non-overlapping genesets. Even more important is the size of the overlap. The higher the overlap, the more quartets are shared among two neighboring genesets. Consequently, a higher amount of information is available to guide the insertion of sequences from a geneset into a tree already reconstructed.

Overlap-Guided Puzzling Step As described above, sufficient overlapping information is needed to reasonably insert a new leaf s_{i+1} into a tree T_i. Hence, instead of using a random permutation of leaves, the permutation has to follow certain restrictions. To satisfy the *pairwise overlap condition*, we apply a graph based procedure similar to Prim's minimum spanning tree (MST) algorithm (see, e.g., [15]) assuming equal edge weights. We start with the leaves from a randomly picked geneset $\mathcal{S}' \in \{\mathcal{S}_1 \ldots \mathcal{S}_k\}$. We define a front set $\mathcal{F}$ containing all unused nodes (genesets) from the overlap graph that are connected by an edge to any geneset already used in the tree. From the sequences in $\mathcal{S}'$ we construct the tree $T(\mathcal{S}')$ applying the usual puzzling step algorithm (see section **??** and chapter **??**). Then we randomly

draw one geneset $\mathcal{S}''$ from $\mathcal{F}$. We remove $\mathcal{S}''$ from $\mathcal{F}$ and add the sets connected to $\mathcal{S}''$, but not yet used to $\mathcal{F}$. The sequences of $\mathcal{S}''$ are then added to the tree. Guided by the overlap graph we thus construct an intermediate tree by sequentially adding the genesets. As in the usual *puzzling step* many intermediate trees are constructed using different orders of leaves and genesets.

Relative Majority Consensus Since missing data adds more ambiguity, this naturally hampers the construction of a resolved majority-rule consensus tree. We therefore decided to apply a majority consensus which is slightly different from the majority-rule consensus used in section **??**. Instead of using only splits occurring in more than 50% (M_l with $l > 0.5$) of the intermediate trees, also splits below 50% are considered. The aim is to use as many splits as are supported by relative majority to extract a maximum uncontradicted information from the intermediate trees.

The relative majority consensus M_{rel} adds all congruent splits from the set of intermediate trees in descending order of occurrence. No splits will be accepted for the consensus that have the same or lower percentage occurrence as any incongruent split. This consensus procedure does not imply a fixed threshold but uses a variable percentage down to the first incongruence guided by the relative majority.

1.6 Networks

1.6.1 Incongruent phylogenies

Reasons

- Methodological reasons
  - randomized taxon order
  - sampling sites (bootstrapping)
- Biological reasons
  - Lineage sorting and recombination
  - Horizontal gene transfer
  - noise in the data
  - ring of life [37, 70] + Martin (1999) BioAssays (cf. doolittle1999a.science.pdf)

1.6.2 Constructing networks

- early models [65], ML on networks [59, 60]
- Median Networks [4, 3]
- Neighbor-Net [11]
- Split decomposition/splits trees [18, 19, 27, 28]

- Spectronet [26]
- Consensus Networks [25]
- Splits graphs/supernetworks [19, 29]
- Moulton and Huber (2005) for review
- ausserdem [5, 35, 39, 40, 66],
- Maximum Agreement of Two Nested Phylogenetic Networks [30]

1.7 Other data sources for tree reconstruction

- linear order / presence-absence of genes
- Hybridisation data, cot
- restriction data
- Expression data (Michael R.)

1.8 Outlook

1.8.1 Model selection

Bibliography

[1] Edward N. Adams III. N-trees as nestings: Complexity, similarity, and consensus. *J. Classif.*, 3:299–317, 1986.

[2] Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, and Jeffrey D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J. Comput.*, 10:405–421, 1981.

[3] Hans-J. Bandelt, Peter Forster, and Arne Röhl. Median-joining networks for inferring intraspecific phylogenies. *Mol. Biol. Evol.*, 16:37–48, 1999.

[4] Hans-J. Bandelt, Peter Forster, Bryan C. Sykes, and Martin B. Richards. Mitochondrial portraits of human populations using median networks. *Genetics*, 141:743–753, 1995.

[5] Mihaela Baroni, Charles Semple, and Mike Steel. A framework for representing reticulate evolution. *Ann. Combin.*, 8:391–408, 2004.

[6] Bernard R. Baum. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, 41:3–10, 1992.

[7] Amir Ben-dor, Benny Chor, Dan Graur, Ophir Ron, and Dan Pelleg. Constructing phylogenies from quartets: Elucidation of eutherian superordinal relationships. *J. Comput. Biol.*, 5:377–390, 1998.

[8] Olaf R. P. Bininda-Emonds. MRP supertree construction in the consensus setting. In M. F. Janowitz, F.-J. Lapointe, F. R. McMorris, B. Mirkin, and F. S. Roberts, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 61, pages 231–242. American Mathematical Society, Providence, Rhode Island, 2003.

[9] Matthew J. Brauer, Mark T. Holder, Laurie A. Dries, Derrick J. Zwickl, Paul O. Lewis, and David M. Hillis. Genetic algorithms and parallel processing in maximum-likelihood phylogeny inference. *Mol. Biol. Evol.*, 19:1717–1726, 2002.

[10] K. Bremer. Combinable component consensus. *Cladistics*, 6:369–372, 1990.

[11] David Bryant and Vincent Moulton. Neighbor-net: An agglomerative method for the construction of phylogenetic networks. *Mol. Biol. Evol.*, 21:255–265, 2004.

[12] David Bryant and Mike Steel. Extension operations on sets of leaf-labeled trees. *Adv. Appl. Math.*, 16:425–453, 1995.

[13] D. Chen, L. Diao, O. Eulenstein, D. Fernández-Baca, and M. J. Sanderson. Flipping: A supertree construction method. In M. F. Janowitz, F.-J. Lapointe, F. R. McMorris, B. Mirkin, and F. S. Roberts, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 61, pages 135–160. American Mathematical Society, Providence, Rhode Island, 2003.

[14] Peter Cooke. Optimal linear estimation of bounds of random variables. *Biometrika*, 67:257–258, 1980.

[15] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 2 edition, 2001.

[16] Leonardo Dagum and Ramesh Menon. OpenMP: An industry-standard API for shared-memory programming. *IEEE Comput. Sci. Eng.*, 5:46–55, 1998.

[17] Alan de Queiroz, Michael J. Donoghue, and Junhyong Kim. Separate versus combined analysis of phylogenetic evidence. *Annu. Rev. Ecol. Syst.*, 26:657–681, 1995.

[18] Andreas Dress, Daniel Huson, and Vincent Moulton. Analyzing and visualizing sequence and distance data using SPLITSTREE. *Discr. Appl. Math.*, 71:95–109, 1996.

[19] Andreas W. M. Dress and Daniel H. Huson. Constructing splits graphs. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 1:109–115, 2004.

[20] Joseph Felsenstein. The number of evolutionary trees. *Syst. Zool.*, 27:27–33, 1978.

[21] R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger in Mathematics*, 41:155–160, 1912.

[22] William Gropp, Steven Huss-Lederman, Andrew Lumsdaine, Ewing Lusk, Bill Nitzberg, William Saphir, and Marc Snir. *MPI: The Complete Reference - The MPI Extensions*, volume 2. The MIT Press, Cambridge, Massachusetts, 2 edition, 1998.

[23] Stéphane Guindon and Olivier Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.*, 52:696–704, 2003.

[24] Michael D. Hendy and David Penny. A framework for the quantitative study of evolutionary trees. *Syst. Zool.*, 38:297–309, 1989.

[25] Barbara R. Holland, Katharina T. Huber, Vincent Moulton, and Peter J. Lockhart. Using consensus networks to visualize contradictory evidence for species phylogeny. *Mol. Biol. Evol.*, 21:1459–1461, 2004.

[26] Katharina T. Huber, Michael Langton, David Penny, Vincent Moulton, and Michael Hendy. Spectronet: a package for computing spectra and median networks. *Appl. Bioinform.*, 20:159–161, 2002.

[27] Daniel H. Huson. SplitsTree: A program for analyzing and visualizing evolutionary data. Materialien/Preprints 110, Universität Bielefeld, Forschungsschwerpunkt Mathematisierung – Stukturbildungsprozesse, Bielefeld, 1997.

[28] Daniel H. Huson. SplitsTree: Analyzing and visualizing evolutionary data. *Bioinformatics*, 14:68–73, 1998.

[29] Daniel H. Huson, Tobias Dezulian, Tobias Klöpper, and Mike A. Steel. Phylogenetic super-networks from partial trees. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 1:151–158, 2004.

[30] Jesper Jansson and Wing-Kin Sung. The maximum agreement of two nested phylogenetic networks. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC 2004)*, volume 3341 of *Lecture Notes in Computer Science*, pages 581–593, New York, 2004. Springer.

[31] T. M. Keane, T. J. Naughton, S. A. A. Travers, J. O. McInerney, and G. P. McCormack. DPRml: distributed phylogeny reconstruction by maximum likelihood. *Bioinformatics*, 21:969–974, 2005.

[32] Alan R. Lemmon and Michel C. Milinkovitch. The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. *Proc. Natl. Acad. Sci. USA*, 99:10516–10521, 2002.

[33] David R. Maddison, Maryellen Ruvolo, and David L. Swofford. Geographic origins of human mitochondrial DNA: phylogenetic evidence from control region sequences. *Syst. Biol.*, 41:111–124, 1992.

[34] Wayne P. Maddison, Michael J. Donoghue, and David R. Maddison. Outgroup analysis and parsimony. *Syst. Zool.*, 33:83–103, 2005.

[35] Vladimir Makarenko and Pierre Legendre. From a phylogenetic tree to a reticulated network. *J. Comput. Biol.*, 11:195–212, 2004.

[36] Tim Margush and Fred R. McMorris. Consensus n-trees. *Bull. Math. Biol.*, 43:239–244, 1981.

[37] William Martin and T. Martin Embley. Evolutionary biology: Early evolution comes full circle. *Nature*, 431:134–137, 2004.

[38] Fred R. McMorris and Dean Neumann. Consensus functions defined on trees. *Math. Soc. Sci*, 4:131–136, 1983.

[39] Bernard M.E. Moret, Luay Nakhleh, Tandy Warnow, C. Randal Linder, Anna Tholse, Anneke Padolina, Jerry Sun, and Ruth Timme. Phylogenetic networks: Modeling, reconstructibility, and accuracy. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 1:13–23, 2004.

[40] Luay Nakhleh, Allen Clement, Tandy Warnow, C. Randal Linder, and Bernard M. E. Moret. Quality measures for phylogenetic networks. UNM Computer Science Tech-Reports TR-CS-2004-06, University New Mexico, Albuquerque, NM, USA, January 2004.

[41] Kevin C. Nixon and James M. Carpenter. On outgroups. *Cladistics*, 9:413–426, 1993.

[42] Gary J. Olsen, Hideo Matsuda, Ray Hagstrom, and Ross Overbeek. fastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Comput. Appl. Biosci.*, 10:41–48, 1994.

[43] Roderic D. M. Page. Modified mincut supertrees. In *Proceedings of the 2nd Workshop on Algorithms in Bioinformatics (WABI 2002)*, volume 2452 of *Lecture Notes in Computer Science*, pages 537–551, New York, 2002. Springer.

[44] Roderic D. M. Page and Edward C. Holmes. *Molecular Evolution: A Phylogenetic Approach*. Blackwell Science, Oxford, 1998.

[45] Andy Purvis. A composite estimate of primate phylogeny. *Philos. Trans. R. Soc. Lond. Ser. B*, 348:405–421, 1995.

[46] M. A. Ragan. Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.*, 1:53–58, 1992.

[47] David L. Roberts and Andrew R. Solow. Flightless birds: When did the dodo become extinct? *Nature*, 426:245, 2003.

[48] Marc Robinson-Rechavi and Dan Graur. Usage optimization of unevenly sampled data through the combination of quartet trees: An eutherian draft phylogeny based on 640 nuclear and mitochondrial proteins. *Isr. J. Zool.*, 47:259–270, 2001.

[49] Nicolas Salamin, Trevor R. Hodkinson, and Vincent Savolainen. Building supertrees: An empirical assessment using the grass family (poaceae). *Syst. Biol.*, 51:136–150, 2002.

[50] Michael J. Sanderson, Andy Purvis, and Chris Henze. Phylogenetic supertrees: Assembling the trees of life. *TREE*, 13:105–109, 1998.

[51] Michael J. Sanderson and H. Bradley Shaffer. Troubleshooting molecular phylogenetic analyses. *Annu. Rev. Ecol. Syst.*, 33:49–72, 2002.

[52] Heiko A. Schmidt, Ekkehard Petzold, Martin Vingron, and Arndt von Haeseler. Molecular phylogenetics: Parallelized parameter estimation and quartet puzzling. *J. Parallel Distrib. Comput.*, 63:719–727, 2003.

[53] Heiko A. Schmidt, Korbinian Strimmer, Martin Vingron, and Arndt von Haeseler. TREE-PUZZLE: Maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*, 18:502–504, 2002.

[54] Charles Semple and Mike Steel. A supertree method for rooted trees. *Discr. Appl. Math.*, 105:147–158, 2000.

[55] Andrew B. Smith. Rooting molecular trees: problems and strategies. *Biol. J. Linn. Soc.*, 51:279–292, 1994.

[56] Marc Snir, Steve W. Otto, Steven Huss-Lederman, David W. Walker, and Jack Dongarra. *MPI: The Complete Reference - The MPI Core*, volume 1. The MIT Press, Cambridge, Massachusetts, 2 edition, 1998.

[57] Alexandros P. Stamatakis, Thomas Ludwig, and Harald Meier. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics*, 21:456–463, 2005.

[58] Craig A. Stewart, David Hart, Donald K. Berry, Gary J. Olsen, Eric A. Wernert, and William Fischer. Parallel implementation and performance of fastDNAml - a program for maximum likelihood phylogenetic inference. In *Proceedings of the International Conference on High Performance Computing and Communications - SC2001*, pages 191–201, November 2001.

[59] Korbinian Strimmer and Vincent Moulton. Likelihood analysis of phylogenetic networks using directed graphical models. *Mol. Biol. Evol.*, 17:875–881, 2000.

[60] Korbinian Strimmer, Carsten Wiuf, and Vincent Moulton. Recombination analysis using directed graphical models. *Mol. Biol. Evol.*, 18:97–99, 2001.

[61] David L. Swofford, Gary J. Olsen, Peter J. Waddell, and David M. Hillis. Phylogeny reconstruction. In David M. Hillis, Craig Moritz, and Barbara K. Mable, editors, *Molecular Systematics*, pages 407–514. Sinauer Associates, Sunderland, Massachusetts, 2 edition, 1996.

[62] Rosa Tarrío, Francisco Rodríguez-Trelles, and Francisco J. Ayala. Shared nucleotide composition biases among species and their impact on phylogenetic reconstructions of the drosophilidae. *Mol. Biol. Evol.*, 18:1464–1473, 2001.

[63] Le Sy Vinh, Heiko A. Schmidt, and Arndt von Haeseler. PhyNav: A novel approach to reconstruct large phylogenies. In Claus Weihs and Wolfgang Gaul, editors, *Classification - The Ubiquitous Challenge*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 386–393. Springer, Heidelberg, 2005.

[64] Le Sy Vinh and Arndt von Haeseler. IQPNNI: Moving fast through tree space and stopping in time. *Mol. Biol. Evol.*, 21:1565–1571, 2004.

[65] Arndt von Haeseler and Gary A. Churchill. Network models for sequence evolution. *J. Mol. Evol.*, 37:77–85, 1993.

[66] Lusheng Wang, Kaizhong Zhang, and Louxin Zhang. Perfect phylogenetic networks with recombination. *J. Comput. Biol.*, 8:69–78, 2001.

[67] Ward C. Wheeler. Nucleic acid sequence phylogeny and random outgroups. *Cladistics*, 6:363–368, 2005.

[68] Marty J. Wolf, Simon Easteal, Margaret Kahn, Brendan D. McKay, and Lars S. Jermiin. TrExML: a maximum-likelihood approach for extensive tree-space exploration. *Bioinformatics*, 16:383–394, 2000.

[69] B. B. Zhou, M. Till, A. Y. Zomaya, and L. S. Jermiin. Parallel implementation of maximum likelihood methods for phylogenetic analysis. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS2004)*, pages 237a 1–6, Los Alamitos, USA, 2004. IEEE.

[70] Wolfgang Zillig. Comparative biochemistry of Archaea and Bacteria. *Curr. Opin. Genet. Dev.*, 1:544–551, 1991.