Phylogenetic Reconstruction Methods: An Overview

Article *in* Methods in molecular biology (Clifton, N.J.) · January 2014

DOI: 10.1007/978-1-62703-767-9_13 · Source: PubMed

3 authors, including:

Alexandre De Bruyn
Sopra Steria Group
25 PUBLICATIONS **126** CITATIONS
SEE PROFILE

Darren P Martin
University of Cape Town
729 PUBLICATIONS **12,907** CITATIONS
SEE PROFILE

Chapter 13

Phylogenetic Reconstruction Methods: An Overview

Alexandre De Bruyn, Darren P. Martin, and Pierre Lefeuvre

Abstract

Initially designed to infer evolutionary relationships based on morphological and physiological characters, phylogenetic reconstruction methods have greatly benefited from recent developments in molecular biology and sequencing technologies with a number of powerful methods having been developed specifically to infer phylogenies from macromolecular data. This chapter, while presenting an overview of basic concepts and methods used in phylogenetic reconstruction, is primarily intended as a simplified step-by-step guide to the construction of phylogenetic trees from nucleotide sequences using fairly up-to-date maximum likelihood methods implemented in freely available computer programs. While the analysis of chloroplast sequences from various *Vanilla* species is used as an illustrative example, the techniques covered here are relevant to the comparative analysis of homologous sequences datasets sampled from any group of organisms.

Key words Phylogeny, DNA sequence, Alignment, Phylogenetic tree, Maximum likelihood

1 Introduction

Invented by Haeckel in 1866, and derived from the concept of genealogy, the term "phylogeny" describes the relationships between entities (such as species, genes, or genomes) [1] in such a way as to reflect their evolutionary histories. Given the assumption that measurable similarities between organisms are suggestive of their common evolutionary history, by comparing analogous features of contemporary or fossil organisms (such as beak sizes or the amino acid sequences of some specific gene), phylogeneticists try to infer the pathways of evolutionary change that yielded these features.

The patterns of relationship and evolutionary pathways that are revealed by phylogenetic analyses are most commonly depicted in the form of phylogenetic trees, the branching patterns and branch lengths of which graphically describe the relative relatedness of the different species, individuals, genes, or other entities of interest (often called operational taxonomic units or OTUs) that were used to construct the trees (Fig. 1).

Pascale Besse (ed.), *Molecular Plant Taxonomy: Methods and Protocols*, Methods in Molecular Biology, vol. 1115,
DOI 10.1007/978-1-62703-767-9_13,

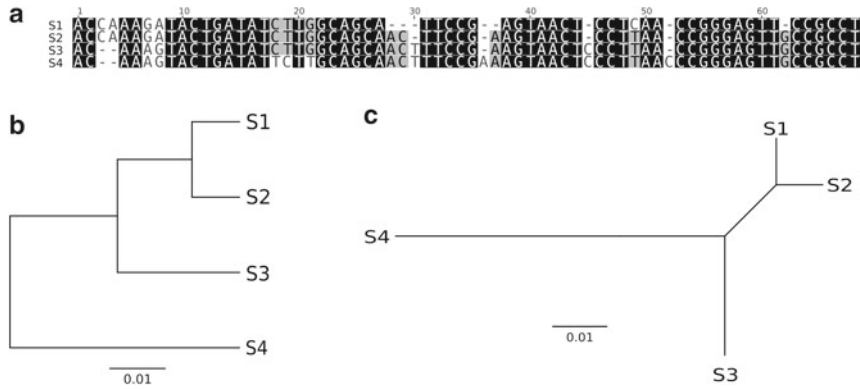

Fig. 1 (**a**) Alignment of four sequences S1, S2, S3 and S4, different shades of *grey* representing polymorphic sites. (**b**) and (**c**), respectively, rooted and unrooted tree of the four sequences presented in the alignment, scale bar corresponding to number of substitutions per site

1.1 Use of Macromolecular Data in Phylogenetic Reconstruction

For many years, phylogenetic trees were constructed based on comparisons among individuals of morphological and physiological characters such as skull morphologies in primates [2] or self-incompatibility in plants [3]. Recent advances in molecular biology and sequencing technologies have created an exponentially increasing volume of DNA and protein sequence data. Since such macromolecular data can be interpreted as a linear string of multistate characters (with four possible states for DNA sequences and 20 possible states for amino acids sequences), their use in phylogeny reconstruction is extremely straightforward. Almost since the moment such data first became available, it was clearly evident that there existed a "molecular clock" such that differences in homologous nucleotide or protein sequences should be good indicators of their relatedness [4, 5]. Furthermore, it was soon realized that the almost universality of the genetic code enabled the use of nucleotide and amino acid sequences to infer phylogenetic relationships even among organisms that were so distantly related that they shared no discernible common morphological or physiological characters. Finally, the mutational processes underlying nucleotide and amino acid sequence evolution are far more amenable to detailed fully probabilistic mathematical modelling than are the gross changes that occur during the evolution of morphological or physiological characters [6].

1.2 Main Methods Used in Phylogenetic Reconstruction

The starting material for constructing any phylogenetic tree from nucleotide or amino acid sequence data is the gathering of sets of evolutionarily related, or homologous, sequences. However, before using these sequences to construct a phylogenetic tree, it is important to ensure that each nucleotide or amino acid in each sequence is compared only with the corresponding homologous nucleotides or amino acids in the other sequences. This preliminary task is performed by aligning the sequences to one another, to obtain a

discrete character matrix called an alignment (see an example in Fig. 1a), within which each row represents one of the sequences and each column a set of homologous nucleotides or amino acids. This step will be described in more detail later, but it should be borne in mind that it is one of the trickiest parts of the whole phylogenetic reconstruction process.

Numerous methods have been developed to infer phylogenetic trees from multiple sequence alignments. Whereas some, called character-based methods, directly use the individual columns of aligned nucleotides or amino acids, others, called distance-based methods, use measures of the overall differences between all pairs of sequences in the alignment (represented as a matrix of pairwise genetic distances). Also, whereas some methods are limited to accommodate only the simplest models of evolution, others offer the capacity to explicitly model various different aspects of the evolutionary process.

Crucially, each method has its own good and bad points, and the choice of one method over another for any particular analysis tends to depend primarily on a compromise between the desired complexity or accuracy of the analysis and the amount of time it will take to run. Nevertheless, with the numerous computational optimizations that have been made to modern phylogenetic reconstruction algorithms and the speed of today's computers, even the various "slow-but-accurate" methods are applicable to most datasets. Applying one such tree construction method, called maximum likelihood (ML), will be the primary focus of the practical component of this chapter.

1.2.1 Distance-Based Methods

Distance-based methods rely on the calculation of genetic distances between each pair of sequences in a dataset with phylogenetic trees being constructed from the resulting distance matrix using a clustering algorithm [7]. The simplest distance measure, the "p-distance" (also called the Hamming distance), corresponds to the proportion of different sites between each pair of taxa. This value is an underestimation of the true evolutionary distance, since the possibility that multiple unseen mutations may have occurred at individual sites is not taken into consideration. Since phylogenetic analyses are concerned primarily with the inference of evolutionary relationships, it is desirable to, in most cases, calculate evolutionary distances rather than simply p-distances. This can be achieved using an explicit model of evolution that corrects the p-distance.

Given a matrix of evolutionary distances, a tree can then be obtained using the unweighted pair grouping with arithmetic mean (UPGMA; [8]), least squares (LS, also called minimum evolution [9]), neighbor-joining (NJ [10]), or BIONJ [11] methods. The primary advantage of such distance-based methods is their computational speed. These methods are therefore ideally suited to the initial exploration of evolutionary relationships between sequences in a dataset. Many of these methods also directly help speed up

slower but more accurate character-based tree construction methods by directing these methods to preferably search for highly likely phylogenetic trees that resemble distance-based trees.

Despite their obvious utility, distance-based methods tend to disregard much of the potentially evolutionarily informative information within an alignment by compressing it into sets of pairwise evolutionary distances [1, 12]. Indeed, all information provided by the distribution of the character states, or relationships between particular characters and the tree, are lost in the process of pairwise-distance calculations [13].

1.2.2 Character-Based Methods

While distance-based methods compress phylogenetic information within a set of sequences into a pairwise-distance matrix, character-based methods take advantage of all the information available in sequences at each homologous site. The most widely used methods are the maximum parsimony (MP; [14, 15]) and maximum likelihood (ML; [16, 17]) methods. Whereas the maximum parsimony methods generally implicitly assume a very simple model of evolution (usually one where all possible nucleotide substitutions are equally probable), the primary appeal of maximum likelihood methods is that they are probabilistic and enable the application of a wide variety of explicit evolutionary models.

The Maximum Parsimony Method

The MP method was, until recently, one of the most widely used for phylogenetic inference. Initially developed for the analysis of morphological traits, its main underlying idea can be best summed up by the principle of Ockham's razor, which states that when several hypotheses with different degrees of complexity are proposed to explain the same phenomenon, one should choose the simplest hypothesis. Framed in a phylogenetic context, this principle proposes that the most believable or parsimonious phylogenetic tree will be the tree that invokes the smallest number of evolutionary changes during the divergence of the sequences it represents [18–20].

A major issue encountered when inferring the most parsimonious tree for a given set of nucleotide sequences is that there will frequently be multiple equally parsimonious trees. In such cases, it is often desirable to produce a strict consensus tree which includes only the topological features that are found in every tree. In a strict consensus tree, it is assumed that unresolved features of the tree topology represent relationships that cannot be resolved due to the studied sequences containing too few phylogenetically informative sites.

Another option when multiple equally parsimonious trees exist is to produce a majority-rule consensus tree which includes only the topological features that are present in at least half of the trees. In many cases, however, such majority-rule consensus trees can have topologies that contradict some of the equally parsimonious trees which should, in fact, be considered just as plausible as the consensus tree [21].

The Maximum Likelihood Method

Maximum likelihood (ML) is a statistical concept popularized by Fischer in the early 1900s [22] that has been widely used in many areas of biology such as population genetics and ecological modelling, and which was first applied to the field of phylogenetics by Joseph Felsenstein in 1973 [23]. The basic concept of likelihood is relatively simple to comprehend: given some data D (in our case, nucleotide or amino acid sequences), under a model of evolution, M (which is explicitly defined and describes the mutation process from one base to another), the likelihood of a set of parameters, θ (tree topology, tree branch lengths, substitution model parameters), corresponds to the probability of obtaining D under the model M with parameters θ. The maximum likelihood estimates of the parameter values included in θ correspond to the set of values that maximize this probability. If the principle is easily understandable, the calculation of the likelihood function can be mathematically complex, and this method can be very computationally expensive.

Although the actual calculation of the likelihood of a particular tree topology and a defined set of parameter values can be quite rapidly computed, finding the set of parameters, including the tree branching orders and branch lengths, for which the likelihood is maximized, is much more computationally daunting due to the incredibly large numbers of trees that must be evaluated even for datasets containing only modest numbers of sequences (e.g., 15 or more).

For this reason, various computational tricks have been devised to cut down on the numbers of trees that need to be evaluated to find the one with the maximum likelihood [24]. Unlike exhaustive search methods, which evaluate all possible phylogenetic trees, so-called exact tree searching methods such as the branch-and-bound method [25] reduce the number of trees that must be evaluated while still guaranteeing that the best tree will be found. However, even exact methods like branch-and-bound are too slow to evaluate datasets with more than a modest number of sequences.

For large datasets, it is usually necessary to use so-called approximate tree searching methods which, while guaranteeing to find trees with high likelihoods, will not always find the tree with the maximum likelihood [26]. The most commonly used approximate tree searching methods involve tree rearrangement approaches such as nearest-neighbor interchange (NNI) and subtree pruning and regrafting (SPR) [27]. These tree rearrangement approaches involve modifications of local branching patterns within small subsections of the tree while leaving the rest of the tree untouched and then testing the new tree to determine whether the rearrangements yield a tree with an improved likelihood. If the new tree has a better likelihood than the original, it is selected for a new round of rearrangements. The process continues until further rearrangements fail to improve the likelihood. The main failing of approximate tree searching methods like NNI and SPR is that just because simple branch rearrangements fail to yield trees

with improved likelihoods, it does not mean that more complex tree rearrangements will not. In many cases, the only pathway to the maximum likelihood tree will involve multiple simultaneous tree rearrangements. Nevertheless, despite the tendency of approximate tree searching methods to become entrapped on local likelihood peaks [26], they will generally very rapidly yield trees with likelihoods close to the maximum.

1.3 Step-by-Step Construction of Phylogenetic Trees

Sequence-based phylogenetic reconstructions can be divided into five main steps: (1) choosing a genome region to study, (2) identifying and retrieving sets of homologous sequences from the same genome region of related individuals, (3) aligning homologous nucleotide/amino acid sites within these sequences, (4) constructing a phylogenetic tree, and (5) visualizing the tree. Although one might assume that the fourth step is the most complex, it is very important to realize that to produce a meaningful tree, none of these steps should be neglected.

Here we describe a robust up-to-date protocol for constructing phylogenetic trees using what is today the most popular available maximum likelihood tree construction software. This chapter is aimed at phylogenetics newbies, and readers interested in a more detailed dissection of the phylogenetic tree construction process are encouraged to consult some of the many excellent reviews and books on advanced tree construction methodologies [13, 24, 28, 29]. We will illustrate the step-by-step construction of a phylogenetic tree using chloroplast *rbc*L sequences sampled from various species of vanilla plants (refer to Chapter 5 on sampling and sequencing protocols).

2 Materials

1. Windows, Mac OS X 10 3+, or Linux Operating System.
2. MEGA5 (http://www.megasoftware.net/) [30].
3. jModelTest2 (http://code.google.com/p/jmodeltest2/) [31].
4. PhyML3 (http://www.atgc-montpellier.fr/phyml/binaries.php) [32].
5. FigTree (http://tree.bio.ed.ac.uk/software/figtree/) [33].
6. Internet connection and an internet browser.
7. Example files located at http://phylorec.blogspot.com/.

3 Methods

3.1 Obtaining a Sequence Dataset

The aim of phylogenetic tree construction is to determine as accurately as possible the evolutionary relationships between groups of organisms. As different sequence datasets can be built to answer

different kinds of question (e.g., do humans and chimpanzees have a more recent common ancestor than either have with gorillas?) or challenge different hypotheses (e.g., chimpanzees are the nearest extant primate relative of humans), before beginning a phylogeny reconstruction, it is important to properly think about the specific biological question (if any) that is being addressed. At this point, it is also crucial to realize that assembling an appropriate sequence dataset and producing an alignment are the most difficult and important part of almost all phylogenetic analyses. If an inappropriate set of sequences is selected (e.g., a set of human and primate sequences where chimpanzee and gorilla sequences are left out) or the alignment produced is of low quality (e.g., if a misalignment error makes it seem that humans are a divergent outlier among the primates), further analysis with the resulting phylogenetic tree could be extremely misleading [34]. Assuming a new DNA sequence was obtained from a newly sampled individual belonging to a particular species, it could be useful to gather sequences belonging to the same or related species from a database. The National Center for Biotechnology Information (NCBI) database and GenBank [35], along with the EMBL Nucleotide Sequence Database, EMBL-Bank [36], and the DNA Database of Japan (DDBJ) [37], host billions of DNA and protein sequence records. Each sequence record is associated with information such as the organism from which the sequence was derived, the author of the sequence, the scientific publications analyzing the sequence, its sampling date and place, and many other pieces of information. Importantly, each sequence is also associated with a set of unique identification numbers, the most important of which is called its accession number. With an accession number in hand, it is very straightforward to retrieve any previously determined and deposited nucleotide or amino acid sequence.

For the purpose of reconstructing the vanilla phylogeny, we generated a fake partial *rbc*L sequence that we will imagine has been obtained using two pairs of PCR primers that amplify two overlapping fragments (*see* ref. 38 for a detailed protocol), and we will use the NCBI database to retrieve homologues of this sequence. Two complementary approaches can be used to query this database and gather the sequence dataset. The first relies on the use of the NCBI basic local alignment search tool (BLAST) (http://blast.ncbi.nlm.nih.gov/Blast.cgi; [39]). BLAST allows one to find a set of sequences with some degree of similarity with a query sequence. This approach is really valuable when no knowledge of the studied sequence is available. BLAST matches are sorted by a "similarity score" (*S* value) and approximate probabilities (*E* values) of the identified similar sequences not being related by evolutionary descent (i.e., that they are similar by chance alone). The selection of homologous sequences is often based upon an *S* value threshold. This common approach consists of taking two sequences to be homologous only if their level of similarity is high enough

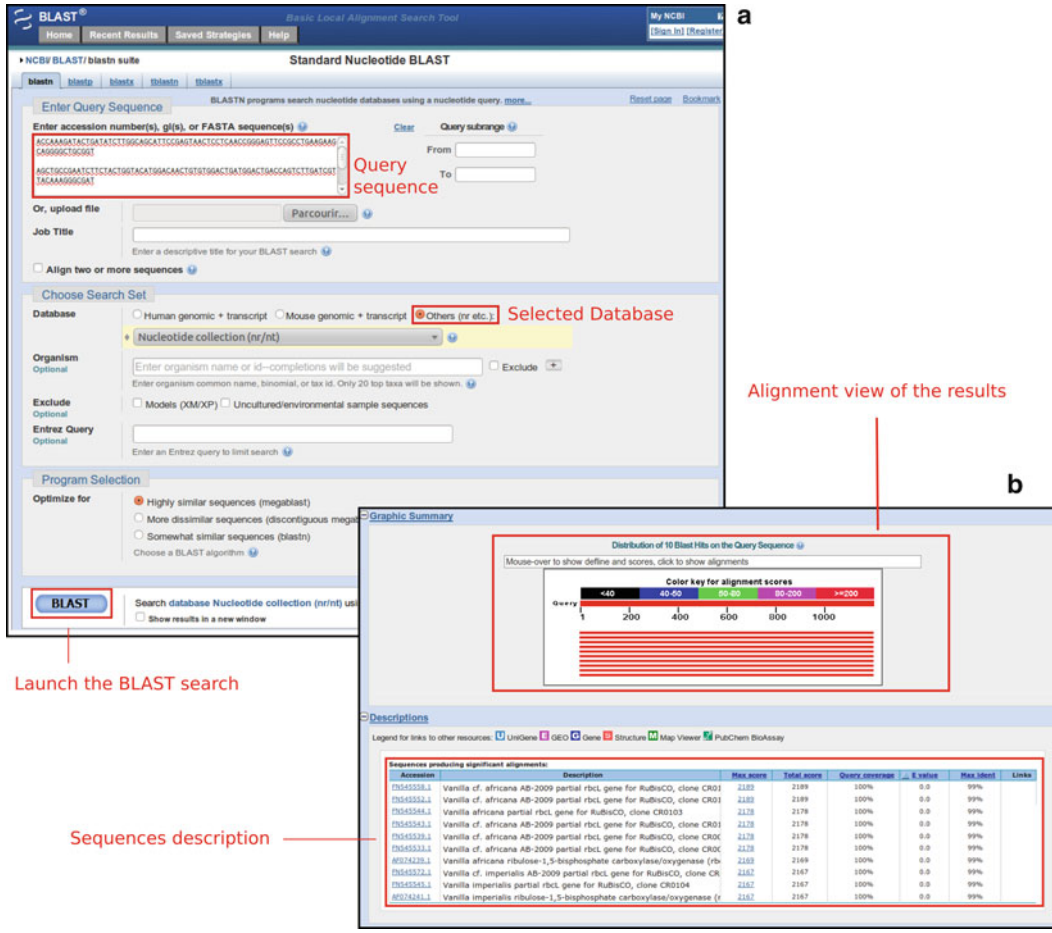

Fig. 2 Screenshots of BLAST with (**a**) search settings and (**b**) results pages

over a large enough proportion of the total query sequence length. Fixing a high level of similarity will also limit our search to sequences from more closely related organisms.

To perform the BLAST with our query sequence, open the NCBI BLAST page on a web browser and select the appropriate BLAST search, here "nucleotide BLAST" in the "basic BLAST" section. Then, copy or upload the query sequence in the "Enter Query Sequence" frame, select "others" in the database "choose search set" section, and then click on the BLAST button (Fig. 2a). The BLAST results will automatically appear (Fig. 2b), presenting the best sequence hits. An inspection of the hits informs us that, as expected, our query is highly related with *Vanilla rbc*L sequences with more than 99 % identity over 100 % of the query length, the first hit corresponding to a partial *rbc*L sequence of *Vanilla africana*. Once the BLAST search is performed, it is possible to download the hit sequences by ticking the appropriate box and clicking the "get selected sequence" button.

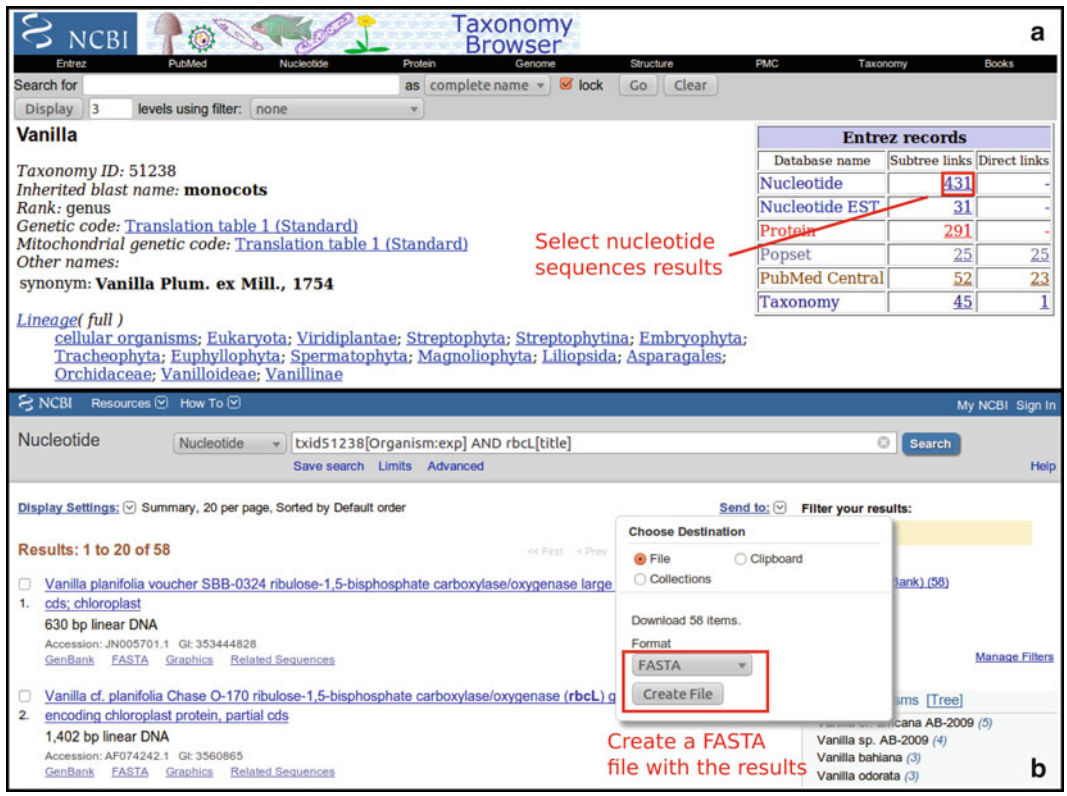

Fig. 3 (**a**) The *Vanilla* genus web page from the Taxonomy Browser Database (NCBI) and (**b**) corresponding nucleotide sequence search results

An alternative approach to obtaining sequences that could eventually be used to construct a *Vanilla* rbcL phylogenetic tree is to use the Taxonomy Browser database (http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html). Typing "vanilla" in the search box will present a classification scheme of the results. A further click on the appropriate link (here the "Vanilla" link at the root of the tree diagram) will load a new page presenting all the records for this taxon (Fig. 3a). On the right side of the results page, we select "nucleotide." Note that doing so just restricts the search to the database with the additional query term "txid51238[Organism:exp]," the taxonomy identification "txid51238" being a specific identification code for the genus *Vanilla*. As we are only interested in *rbc*L sequences, we can simply add to the current query the term "AND rbcL[title]" to restrict our search to this specific gene. Beware that without providing the "[title]" designator in conjunction with the "rbcL" term, the search will return any sequence entry with the consecutive letters r-b-c and -l present in any of the sequence record fields (many of which will not be *rbc*L genes records). Note that [Organism] and [title] are two of the many "Entrez query" terms that can be used

a FASTA format

```
>S1
ACCAAAGATACTGATATCTTGGCAGCA---TTCCG--AGTAACT-CCTCAA-CCGGGAGTT-CCGCCT-
>S2
ACCAAAGATACTGATATCTTGGCAGCAAC-TTCCG-AAGTAACT-CCTTAA-CCGGGAGTTGCCGCCT-
>S3
AC--AAAGTACTGATATCTTGGCAGCAACTTTCCG-AAGTAACTCCCTTAA-CCGGGAGTTGCCGCCTT
>S4
AC--AAAGTACTGATATTCTTGCAGCAACTTTCCGAAAGTAACTCCCTTAACCCGGGAGTTGCCGCCTT
```

b Phylip format

```
4 69
S1        ACCAAAGATACTGATATCTTGGCAGCA---TTCCG--AGTAACT-CCTCAA-CCGGGAGTT-CCGCCT-
S2        ACCAAAGATACTGATATCTTGGCAGCAAC-TTCCG-AAGTAACT-CCTTAA-CCGGGAGTTGCCGCCT-
S3        AC--AAAGTACTGATATCTTGGCAGCAACTTTCCG-AAGTAACTCCCTTAA-CCGGGAGTTGCCGCCTT
S4        AC--AAAGTACTGATATTCTTGCAGCAACTTTCCGAAAGTAACTCCCTTAACCCGGGAGTTGCCGCCTT
```

Fig. 4 Examples of (**a**) FASTA and (**b**) PHYLIP formats of multiple sequence alignments containing four sequences

to filter results according to sequence record file fields (such as deposition date and country of origin) or sequence characteristics (such as length). A total of 58 appropriate *rbc*L sequences are available for download simply by clicking the "Send to" button and choosing "file" (Fig. 3b). It is highly recommended that the sequences be downloaded in FASTA format. Among the large variety of alignment formats, the FASTA format is among the simplest and most widely usable by sequence analysis software. Adding or removing a sequence entry from a FASTA file can be performed in a simple text editor by pasting in, or deleting, the name and lines of sequence for that entry (see an example of the FASTA file format with example files and in Fig. 4a).

A further consideration when assembling a dataset is whether a rooted phylogenetic tree is desired. Specifically, a phylogenetic tree can be either rooted or unrooted. Whereas in a rooted tree it is clear which direction sequences are evolving in (usually represented in a left-to-right orientation with the left-most node representing the most recent common ancestor of all the sequences in the tree; Fig. 1b), in an unrooted tree, the direction of evolution is unspecified (Fig. 1c) [21]. It is usually desirable to root phylogenetic trees, and for this reason, different rooting methods have been devised. The most common and generally reliable of these is the outlier rooting method. With this method, an "outlier sequence" is selected that (1) must be homologous to the sequences in the dataset of interest and (2) must be from an organism that is less closely related to the sequences in the dataset than any of these sequences are to one another, and (3) of all the sequences not included in the dataset, the outlier should be as closely related to the sequences in the dataset as possible [40]. For the purposes of our example analysis, we have chosen the *Pseudovanilla ponapensis* partial *rbc*L sequence (accession number AY381131) as an outlier because it meets all of these criteria.

All the sequences (the *Vanilla rbc*L sequences, the *Pseudovanilla rbc*L outlier, and the sequence we are attempting to place phylogenetically) have to be pasted into the same FASTA file. Note that the sequences in the FASTA file downloaded from NCBI each have a very long name that includes a description of the sequence. For convenience, we have used a standard text editor to crop the names to include only the accession number (the "gb field" in the sequence name; final FASTA file available with example files).

3.2 Aligning the Sequences

Multiple sequence alignment involves lining up the homologous sites of homologous nucleotide or amino acid sequences. Specifically, a sequence alignment is essentially a table, or matrix, of data within which each sequence is assigned a separate row in the matrix, with homologous nucleotide or amino acid positions in different sequences lined up into columns. When the alignment is used to construct a phylogenetic tree, the residues in each particular column are assumed to be different states of a homologous trait derived by mutation from common residue in an ancestral sequence (Fig. 1a).

Whereas modern multiple sequence alignment algorithms are fast enough to align a fairly large set of sequences (relative to the kind of analysis we describe in this chapter), it can be an extremely difficult and a time-consuming task to refine the alignments that these algorithms yield. Whereas alignment is trivial when the degree of diversity between the sequences being analyzed is low (as is the case with our *Vanilla rbc*L dataset), it gets considerably more complex as sequences get more divergent. It is very important to realize that none of the frequently used alignment programs is capable of consistently producing perfect alignments even for moderately divergent sequences. Therefore, for both easy and more complex alignments, it is always important to check by eye for obviously misaligned nucleotides and shift these back into alignment by adding and removing alignment gaps.

The MEGA5 [30] computer program implements one of the best free multiple sequence alignment editors, and it is the one that we will use here. It allows the use of two distinct automatic alignment methods, ClustalW [41] and MUSCLE [42]. Sequences can be aligned as a whole or a subset of sites can be selected. This last functionality is very useful when different parts of the sequences have different degrees of diversity (as is commonly the case when the aligned sequences include both coding and noncoding sequences).

It is worth noting that MEGA allows one to alter various alignment parameters that can in some cases improve the quality of the alignments produced by ClustalW and MUSCLE. The most noteworthy of these parameters are the gap open penalty (GOP) and gap extension penalty (GEP) and, in the case of the MUSCLE method, the "maximum iterations" setting. Alignment methods such as ClustalW and MUSCLE focus on maximizing an alignment

score where matching characters in columns are given a positive score and mismatches are given a negative score. To obtain the alignment, gaps (the "-" character) corresponding with hypothetical ancestral insertion/deletion events are introduced into the sequences at sites that maximize the alignment score. Wherever a gap is added in isolation, the alignment score is penalized by the amount specified by the GOP parameter, and when subsequent gaps are added adjacent to an existing gap, the alignment score is penalized by the amount specified by the GEP parameter. Whereas this procedure is performed in a single round with the ClustalW algorithm, multiple iterations of the same process can be performed with the MUSCLE algorithm (controlled by the maximum iterations parameter). Increasing the iteration number will theoretically improve the alignment but will have a cost in terms of speed.

However, even when multiple iterations are performed, the MUSCLE method is considerably faster than the ClustalW method. As a first alignment step, one might align a set of sequences with MUSCLE with default parameters with the maximum iterations setting between 2 and 6. Dependent on the alignment quality obtained in this first step, it is advisable that a second alignment step should be to realign particularly badly aligned regions of the alignment either with the ClustalW method or with a mixture of ClustalW and MUSCLE with GOP and GEP settings that are lower than the default values (i.e., so as to penalize alignment gaps less severely). The third step should then be to edit the alignment by eye focusing particularly on the exact placement of gap characters so as to minimize the number of mismatches in individual alignment columns.

Following this alignment procedure using MEGA, it is recommended that the final alignment be saved in FASTA format by pressing the "Data" menu option, then the "Export Alignment" menu option, and then the "FASTA format" menu option.

Open MEGA5 and click on "Align" and then "Edit/Build alignment." Choose "Retrieve sequences from a file" and select the FASTA file that you obtained in the previous exercise. You will be presented with a matrix of characters with unaligned sequences running in rows. In our example, due to a low level of variability between *rbc*L sequences, one iteration of the MUSCLE algorithm is sufficient to obtain a good alignment. Select all sequences (CTRL+A) and align them with MUSCLE by clicking on the MUSCLE icon (you can also do this via the "Alignment" menu and select the "Align by Muscle" option). Keep gap penalties with their default values, modify "Max Iterations" to one, and then click on the "Compute" button. As emphasized before, any automatically produced alignment must be checked by eye for (1) misaligned sections of sequence (i.e., incorrectly placed gap characters), (2) truncated/partial sequences, and (3) sequences that are either nonhomologous or in the incorrect orientation. Once the overall alignment is completed,

the ends of the alignment should in most cases be trimmed to the size of the sequences of interest (e.g., from the beginning of the *rbc*L start codon to the end of the *rbc*L stop codon).

Once the alignment is performed and has been properly checked, simply export the file in FASTA format. Whereas jModelTest2 [31], the program we next use, will work with many different alignment formats, PhyML3 [32], the program we use after jModelTest2, requires an alignment in PHYLIP format (see example file and Fig. 4b). To obtain a PHYLIP file, the easiest is to use the same converter used with jModelTest2. Simply run the "alter.jar" application available in the jModelTest2/lib repertory, open your last saved FASTA file by clicking on "Load…" under the input box, press the convert button, and choose the "GENERAL" option and then the "PHYLIP" option to save the alignment as a PHYLIP file.

3.3 Determining the Best-Fit Nucleotide Substitution Model

The major advantage of using ML in the context of phylogenetic inference is that it allows the construction of phylogenetic trees within a very well-defined model-based statistical framework. This framework allows one to rationally determine whether the evolution of the observed sequences is more consistent with certain evolutionary models (detailing nucleotide substitution and tree topology parameters) than it is with others. The likelihood score of a particular tree topology and nucleotide substitution model with particular nucleotide substitution parameters is calculated by multiplying the estimated probabilities of each alignment column given the model under consideration. This likelihood score is usually very small even when alignments are very short such that for convenience the likelihood is usually presented in the form of the negative of its natural logarithm (usually denoted –lnL). The smaller the –lnL value is, the more likely the model is [24].

As in every statistical analysis, the model describing the nucleotide substitution process can be either simple (all possible types of substitution are equally probable) or complex (all types of substitution occur at different rates), and the degree of complexity of the model will depend on how informative the data is. Fortunately, programs like jModelTest2 are available to compare different possible models of evolution and identify which models are best supported by the observed sequences. For each of up to 88 separate models jModelTest2, maximum likelihood scores are computed and compared. Four different model comparison procedures are implemented including likelihood ratio tests (hLRT [43]), the Akaike information criterion tests (AIC [44]), the Bayesian information criterion tests (BIC [45]), and the decision theory method (DT [46]). Each of those model selection procedures compares the likelihood of the models, taking into account the fact that in many cases different models have different numbers of free parameters. It has been determined that, in the

phylogenetic context at least, the BIC and DT tests tend to support simpler models that perform as well as more complex models selected using hLRT and AIC tests and that the BIC and DT tests should therefore be preferentially used when deciding on the best-fit model [47, 48].

The models that jModelTest2 evaluates range from the simple Jukes-Cantor 1969 model, [49] where all mutations occur at the same basal rate, to more complex general time reversible (GTR) models where every nucleotide substitution is free to occur at a distinct rate [50]. Three other parameters can increase the complexity of models including accounting for unequal base frequencies (+F), the varying proportions of invariant sites (+I), and variations in the substitution rate across sites (+G). For this last parameter, a given number of discrete rate categories are used to model variations in substitution rates among sites. The default number of substitution rate categories in PhyML3 is four, but this number can be freely changed (although it must be realized that the analysis time will increase linearly with this number).

To begin the model selection procedure, simply start the jModelTest.jar application. Load the data (the alignment in either FASTA or PHYLIP format) with file >open>select your data. The model selection procedure could be time-consuming, and depending of the amount of data, a restricted number of models can be tested. A number of substitution schemes (NSS) of three is straightforward to set up in PhyML3 for later tree reconstruction and will be chosen for the analysis we perform here. Select "+F," "+I," and "+G" before running the analysis by pressing the "Compute Likelihoods" button (Fig. 5a).

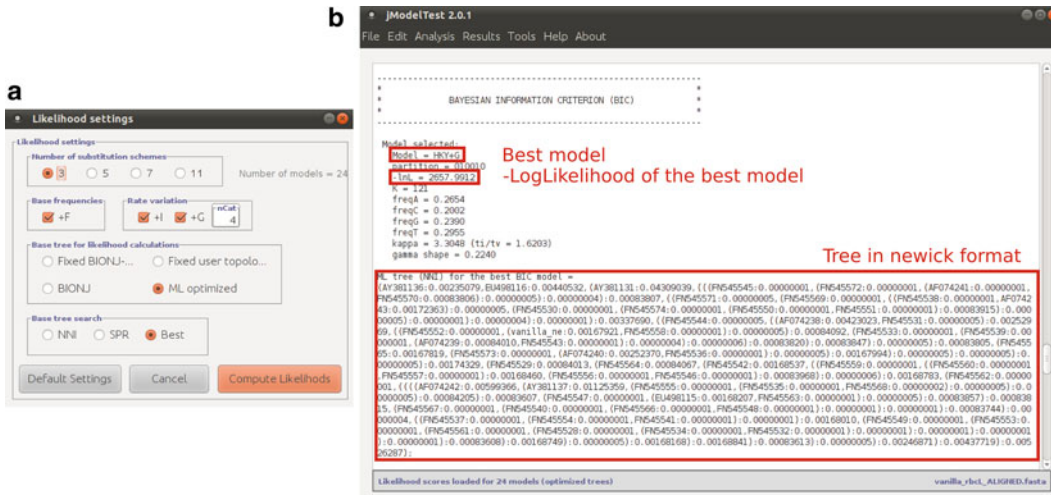

Fig. 5 (**a**) jModelTest2 settings window and (**b**) BIC result for the best-fit model with its associated tree in parenthetic format

Once the analysis is over, select either BIC or DT in the Analysis panel, leaving the confidence interval as is (this function permits one to choose a hierarchical selection of models in order to do model-averaging parameters estimation and will not be further mentioned in this chapter). After running the test, in the output the best supported model will be selected and appears first in the list. In our example, the best supported model according to the BIC test is the HKY + G model (Fig. 5b), a model for which transitions and transversions are free to occur at different rates and where basal substitution rates vary from site to site along the sequences and all bases are not assumed to occur at equal frequency [51].

3.4 Reconstructing a Phylogeny

The jModelTest2 software provides in its log file the phylogenetic tree obtained with the best-fit model (Fig. 5b). The tree is available in a parenthetic format (the Newick format) and can be directly copied from the log to any text editor before being loaded in an appropriate tree viewing program. While useful, this tree lacks any indication of how well supported individual branches within the tree are. To achieve this, it is usually desirable to perform bootstrap tests to identify the branches that are most and least supported by the available data.

To assess the robustness of the ML tree produced by jModelTest2, we will use PhyML3 to analyze the *rbc*L sequences with the HKY + G model indicated to be the best-fit model by jModelTest2. Two tests of branch support are available in PhyML3. The first is the well-established and widely used bootstrap test [52]. In this test, alignment sites (the columns) are sampled with replacement to obtain new "virtual alignments" derived from our real data. Different phylogenetic trees are then inferred for each of these resampled alignments (usually between 100 and 1,000 times alignment + tree combinations), and the percentage of times that particular groups of sequences that cluster in the tree constructed from the real sequences also cluster in the bootstrapped trees is used as a measure of robustness of the branches separating these sequences from the remainder of the tree. The statistical meaning of such bootstrap values is obscure, and it must be remembered that they are in no way "p-values." In fact it is generally accepted that branches that are supported in as few as 70 % of the bootstrap replicates should be considered to be robustly supported.

Considerably less statistically obscure is the alternative to the bootstrap test that is provided by PhyML3: the approximate likelihood ratio test (aLRT, a derivative of the LRT; [53]). This fast nonparametric test provides branch statistics that are essentially approximate *p*-values that indicate whether trees containing the branch have a significantly better likelihood than the best alternative tree topologies where the branch is absent. Besides being much faster to compute than branch supports determined by bootstrapping, branch supports determined by aLRT are also much easier to interpret.

To start PhyML3, simply double click on its icon or launch it from the command line. A screen appears asking for the alignment name. This alignment must be in PHYLIP format (obtained previously) and should be placed in the same directory as the PhyML3 program (otherwise, the full path of the alignment should be provided with the name). A menu will then appear with default parameters for the input data. Four pages of submenu, the input data, the model of substitution, the tree searching, and the branch support menu, must be checked and set to the appropriate values before running the analysis. To navigate between the sub-menus, the "+" and "-" commands are used. For each line of the menu, the value of a given entry is written on the right of the screen. To toggle the value, simply type the letter given between brackets on the left of the line until the correct parameter is set (see the squared "m" letter on the line of the model of nucleotide substitution, Fig. 6). For example, to toggle from DNA to protein, simply type "d." Another "d" will set the value back to DNA. All the default parameters are fine in this sub-menu. However, the "Run ID" option, permitting one to name the output files, can be useful to keep track of the analyses performed with different analysis settings.

In the next sub-menu, we have to set up the substitution model parameters. The best-fit evolutionary model selected with jModelTest2 was HKY + G. To set this model, select the "m" option. Next select the "r" option so as to allow rate variation across sites, and then select the "c" option and make sure that four rate categories are defined.

Once the model is set up, press "+" to reach the tree searching submenu. Here, the tree topology search operations should be set to "Best of NNI and SPR" using the "s" option. This setting is

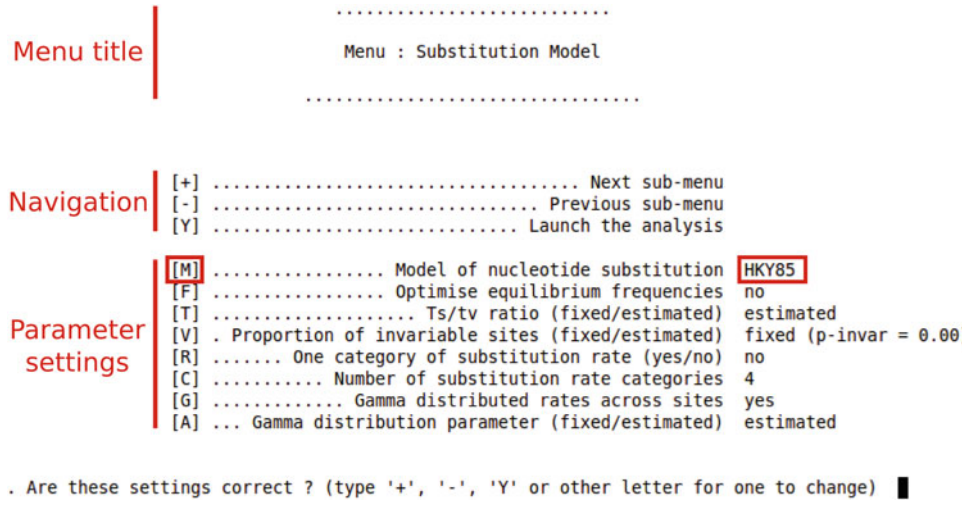

Fig. 6 PhyML substitution model submenu

more computationally intensive than the default setting ("NNI") but often ensures better results. Finally in the last submenu, the bootstrap analysis and aLRT can be turned on and off. Note that turning on the bootstrap analysis will turn off the aLRT and vice versa. Several statistics are available to infer branch confidence with the aLRT method. Documentation about these different statistics can be found in [53] and [54]. Here we will use the SH-like statistic, which have been proved to be more robust to violations of model assumptions [54]. Once every parameter is set appropriately, simply type "y" to launch the analysis.

Once the tree search begins, the program will continuously keep you updated on its progress and will eventually shut down automatically when it perceives it has found what is probably the maximum likelihood tree. This tree will be written to the file <align_name>_phyml_tree.txt<_RunID>.txt, in a parenthetic format, and the run summary statistics will be written to the file <align_name>_phyml_stats.txt<_RunID>.txt.

3.5 Visualization and Editing of Trees

The last step of the phylogeny reconstruction process is the visualization and editing of the trees that PhyML3 produces. Several programs have the ability to take a tree file in parenthetic format and plot a graphic of the tree(s) depicted in the file. The program we present here is called FigTree [33]. After starting the figtree.jar application, simply open the tree file (here, the <align_name>_phyml_tree.txt<_RunID>.txt file) with file>open. A pop-up question window will appear offering the option for you to rename the branch labels (i.e., type either "bootstrap" or "aLRT" depending on the test that was performed). Once the tree is opened, among the many options available are those allowing one to edit, color, and modify the appearance of the tree. To label branches with their bootstrap/aLRT support values, tick the "Node labels" box and select label/bootstrap/aLRT. The scale can be adjusted for convenience.

The *Vanilla* genus has been shown to segregate into three groups: α, β, and γ [38]. Our ML phylogenetic tree indicates that our new sequence sits within the γ group (colored in yellow on the tree) corresponding to Old World and Caribbean species and more precisely with *Vanilla africana* accessions (FN545552 and FN545558). This placement within the tree is consistent with the results obtained in our earlier BLAST search.

The tree presented in FigTree can be printed as is or exported in any one of several different image formats (Fig. 7). Note that the enhanced metafile (.emf) or windows metafile (.wmf) formats are good choices for a further editing of the image file in any graphics editing programs. The Scalable Vector Graphics (.svg) format is also a good choice if one would like to edit the graphics in open-source graphics editors such as Inkscape (www.inkscape.org).

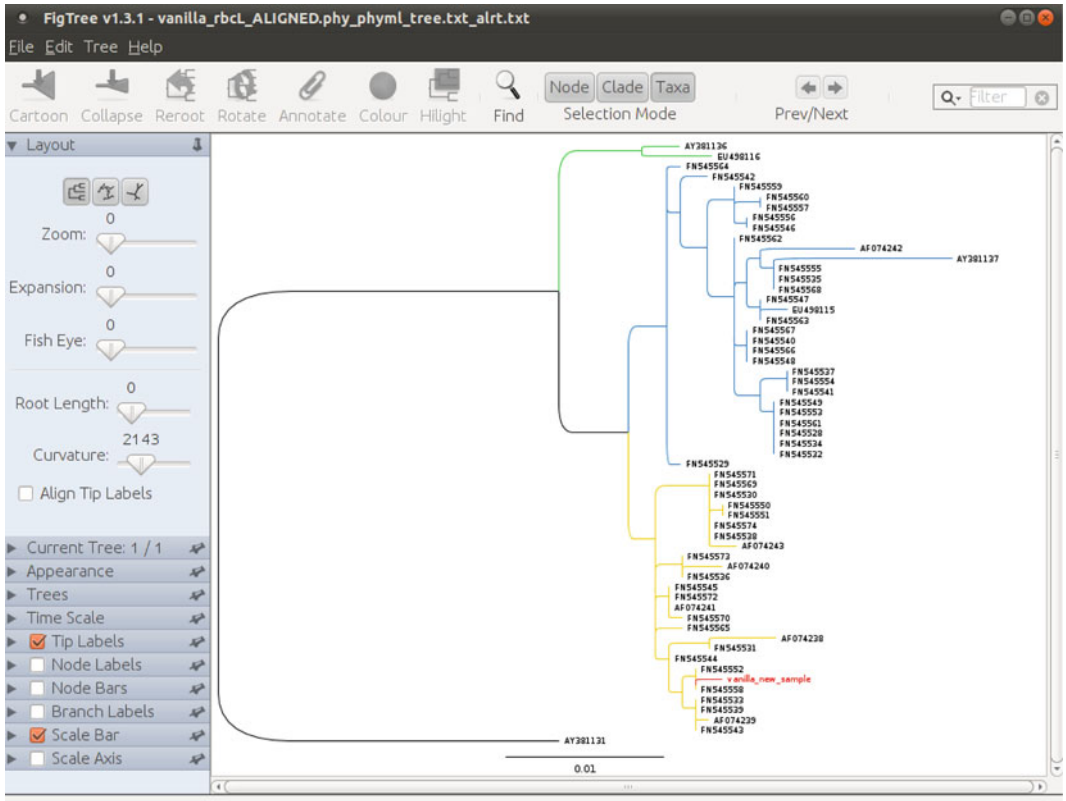

Fig. 7 FigTree representation of the phylogenetic reconstruction of *Vanilla* samples, with colors representing the three distinct clades of the genus (*green*, α; *blue*, β; *yellow*, γ [38]), and the query sequence used in our example shown in *red*. For the purpose of clarity, aLRT values are not displayed (Color figure online)

4 Notes

1. *PhyML special model*
   In the example given above, we select only one of several nucleotide substitution models available in PhyML3. Several other models exist and can be set up in PhyML3 using "custom" settings in the "Substitution model" menu. The model is then set using a binary representation of the substitution matrix. Information on how these models are implemented can be found in both the jModelTest and PhyML manuals.
2. *Other software with (nearly) the full procedure implemented*
   The phylogenetic reconstruction procedure described here is implemented in part or whole in various other computer programs that the user may want to use. Usually these programs still employ the programs we present here or use a similar analysis pipeline. It is important to keep in mind that the software landscape is moving rapidly and faster than the methods.

3. *Amino acid substitution model*
   While we present our phylogenetic reconstruction using nucleotide sequences, it is equally applicable to amino acid sequences. Obviously in the case of amino acid sequences, amino acid and not nucleotide substitution models have to be chosen. Amino acid substitution model selection can be performed using the ProtTest3 program (http://darwin.uvigo.es/software/prottest3/prottest3.html; [55]), designed by the same group that produces jModelTest2.
4. *Recombination*
   As different portions of an organism's genome can have different origins, recombination is a major issue for phylogenetic reconstruction [56]. The evolutionary history of a recombinant sequence cannot usually be depicted by a single phylogenetic tree (the regions of the recombinant genome derived from its two parents should be described by different phylogenetic trees), and attempting to explain the evolutionary history of such sequences with a single tree can be very misleading [56, 57]. This should be borne in mind before any phylogenetic analysis and especially when one chooses the type of data or genetic locus to analyze.
5. *Bayesian phylogenetic inference*
   Another excellent approach for phylogenetic reconstruction relies on Bayesian inference and is implemented in programs such as MrBayes3 (http://mrbayes.sourceforge.net/; [58]) and BEAST (http://beast.bio.ed.ac.uk/Main_Page; [59]). ML and Bayesian inference are known to perform similarly, with respect to determining the true phylogeny underlying the evolution of a set of sequences [60, 61]. Rather than focusing on the inference of a single "best" tree, Bayesian inference focuses on the identification of groups of similarly plausible phylogenetic trees. In this regard, Bayesian inference of phylogenies is favored in applications where one would like to account for phylogenetic uncertainty while inferring, for example, ancestral sequences or sites evolving under natural selection.

Acknowledgments

ADB is supported by the Conseil Général de La Réunion and CIRAD. DPM is supported by the Wellcome Trust. PL is supported by CIRAD and Conseil Régional de La Réunion and European Union (FEDER). The authors wish to thank Dr. Jean-Michel Lett for his helpful comments.

References

1. Darlu P, Tassy P (1993) La reconstruction phylogénétique. Concepts et Méthodes. Masson
2. Groves C (1986) Systematics of the great apes. In: Swindler DR, Erwin J (eds) Comparative primate biology: systematics, evolution and anatomy, vol 1. Liss AR, New York, pp 187–217
3. Hemsley AR, Poole I (2004) The evolution of plant physiology. From whole plants to ecosystems. Elsevier Academic Press, Amsterdam
4. Caputo P (1997) DNA and phylogeny in plants: history and new perspectives. Lagascalia 19:331–344
5. Zuckerkandl E, Pauling L (1965) Molecules as documents of evolutionary history. J Theor Biol 8:357–366
6. Nei M, Kumar S (2000) Molecular evolution and phylogenetics. Oxford University Press, New York
7. Van de Peer Y (2009) Phylogeny inference based on distance methods. In: Salemmi M, Vandamme AM (eds) The phylogenetic handbook, a practical approach to DNA and protein phylogeny. Cambridge University Press, New York, pp 101–135
8. Michener CD, Sokal RR (1956) A quantitative approach to a problem in classification. Evolution 11:130–162
9. Fitch WM, Margoliash E (1967) Construction of phylogenetic trees. Science 155:279–284
10. Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol 4:406–425
11. Gascuel O (1997) BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol Biol Evol 14: 685–695
12. Steel MA, Hendy MD, Penny D (1988) Loss of information in genetic distances. Nature 336:118
13. Felsenstein J (2004) Inferring phylogenies. Sinauer Associates, Sunderland
14. Sober E (1988) Reconstructing the past: parsimony, evolution, and inference. MIT Press, Cambridge
15. Edwards AWF, Cavalli-Sforza LL (1964) Reconstruction of evolutionary trees. In: Heywood VH, McNeill J (eds) Phenetic and phylogenetic classification: a symposium. Systematics Association, London, pp 67–76
16. Cavalli-Sforza LL, Edwards AWF (1967) Phylogenetic analysis: models and estimation procedures. Evolution 32:550–570
17. Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17:368–376
18. Farris JS (1970) Methods for computing Wagner trees. Syst Zool 19:83–92
19. Fitch WM (1971) Towards defining the course of evolution: minimum change for a specific tree topology. Syst Zool 20:406–416
20. Kluge AG, Farris JS (1969) Quantitative phyletics and the evolution of anurans. Syst Zool 18:1–32
21. Harrison CJ, Langdale JA (2006) A step by step guide to phylogeny reconstruction. Plant J 45:561–572
22. Aldrich J (1997) R. A. Fisher and the making of maximum likelihood 1912–1922. Statist Sci 12:162–176
23. Felsenstein J (1973) Maximum-likelihood estimation of evolutionary trees from continuous characters. Am J Hum Genet 25:471–492
24. Schmidt HA, von Haeseler A (2009) Phylogenetic inference using maximum likelihood methods. In: Salemmi M, Vandamme AM (eds) The phylogenetic handbook, a practical approach to DNA and protein phylogeny. Cambridge University Press, New York, pp 181–209
25. Hendy MD, Penny D (1982) Branch and bound algorithms to determine minimal evolutionary trees. Math Biosci 59:277–290
26. Swofford DL, Sullivan J (2003) Phylogeny inference based on parsimony and other methods using Paup*. In: Salemmi M, Vandamme AM (eds) The phylogenetic handbook, a practical approach to DNA and protein phylogeny. Cambridge University Press, New York, pp 267–312
27. Swofford DL, Olsen GJ (1990) Phylogeny reconstruction. In: Hillis DM, Moritz C, Mable BK (eds) Molecular systematics. Sinauer Associates, Sunderland, pp 411–501
28. Swofford DL et al (1996) Phylogenetic inference. In: Hillis DM, Moritz C, Mable BK (eds) Molecular systematics. Sinauer Associates, Sunderland, pp 407–514
29. Ronquist F, van der Mark P, Huelsenbeck JP (2009) Bayesian phylogenetic analysis using MrBayes. In: Salemmi M, Vandamme AM (eds) The phylogenetic handbook, a practical approach to DNA and protein phylogeny. Cambridge University Press, New York, pp 210–266
30. Tamura K et al (2011) MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. Mol Biol Evol 28:2731–2739
31. Posada D (2008) jModelTest: phylogenetic model averaging. Mol Biol Evol 25: 1253–1256
32. Guindon S et al (2010) New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst Biol 59:307–321

33. Morariu V et al (2008) Automatic online tuning for fast Gaussian summation. Advances in Neural Information Processing Systems (NIPS) 1–8
34. Hall BG (2007) Phylogenetic trees made easy: a how-to manual, 3rd edn. Sinauer Associates, Sunderland
35. Benson DA et al (1994) GenBank. Nucleic Acids Res 22:3441–3444
36. Cochrane G et al (2009) Petabyte-scale innovations at the European Nucleotide Archive. Nucleic Acids Res 37:D19–D25
37. Tateno Y et al (2002) DNA Data Bank of Japan (DDBJ) for genome scale research in life science. Nucleic Acids Res 30:27–30
38. Bouetard A et al (2010) Evidence of transoceanic dispersion of the genus *Vanilla* based on plastid DNA phylogenetic analysis. Mol Phyl Evol 55:621–630
39. Altschul SF et al (1990) Basic local alignment tool. J Mol Biol 215:403–410
40. Maddison WP, Donoghue MJ, Maddison DR (1984) Outgroup analysis and parsimony. Syst Zool 33:83–103
41. Thompson JD, Higgins DG, Gibson TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucleic Acids Res 22:4673–4680
42. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 32: 1792–1797
43. Posada D, Crandall KA (1998) Model test: testing the model of substitution. Bioinformatics 14:817–818
44. Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control 19:716–723
45. Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6:461–464
46. Minin V et al (2003) Performance-based selection of likelihood models for phylogeny estimation. Syst Biol 52:674–683
47. Luo A et al (2010) Performance of criteria for selecting evolutionary models in phylogenetics: a comprehensive study based on simulated datasets. BMC Evol Biol 10:242
48. Ripplinger J, Sullivan J (2008) Does choice in model selection affect maximum likelihood analysis? Syst Biol 57:76–85
49. Jukes TH, Cantor CR (1969) Evolution of protein molecules. In: Munro HN (ed) Mammalian protein metabolism. Academic, New York, pp 21–132
50. Tavaré S (1986) Some probabilistic and statistical problems in the analysis of DNA sequences. Lect Math Life Sci (Am Math Soc) 17:57–86
51. Hasegawa M, Kishino H, Yano T (1985) Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol 22:160–174
52. Felsenstein J (1985) Confidence limits on phylogenies: an approach using the bootstrap. Evolution 39:783–791
53. Anisimova M, Gascuel O (2006) Approximate likelihood-ratio test for branches: a fast, accurate, and powerful alternative. Syst Biol 55: 539–552
54. Anisimova M et al (2011) Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes. Syst Biol 60:6 85–699
55. Darriba D et al (2011) ProtTest3: fast selection of best-fit models of protein evolution. Bioinformatics 27:1164–1165
56. Ruths D, Nakhleh L (2005) Recombination and phylogeny: effects and detection. Int J Bioinform Res Appl 1:202–212
57. Posada D, Crandall KA (2002) The effect of recombination on the accuracy of phylogeny estimation. J Mol Evol 54:396–402
58. Ronquist F, Huelsenbeck JP (2003) MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574
59. Drummond AJ et al (2012) Bayesian phylogenetics with BEAUti and the BEAST 1.7. Mol Biol Evol 29:1969–1973. doi:10.1093/molbev/mss075
60. Rannala B, Yang Z (1996) Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. J Mol Evol 43:304–311
61. Mau B, Newton M, Larget B (1999) Bayesian phylogenetic inference via Markov chain Monte Carlo methods. Biometrics 55:1–12