Phylogenetic Analysis of Protein Sequence Data Using the Randomized Axelerated Maximum Likelihood (RAXML) Program

Antonis Rokas[1]

[1]Department of Biological Sciences, Vanderbilt University, Nashville, Tennessee

ABSTRACT

Phylogenetic analysis is the study of evolutionary relationships among molecules, phenotypes, and organisms. In the context of protein sequence data, phylogenetic analysis is one of the cornerstones of comparative sequence analysis and has many applications in the study of protein evolution and function. This unit provides a brief review of the principles of phylogenetic analysis and describes several different standard phylogenetic analyses of protein sequence data using the RAXML (Randomized Axelerated Maximum Likelihood) Program. *Curr. Protoc. Mol. Biol.* 96:19.11.1-19.11.14. © 2011 by John Wiley & Sons, Inc.

Keywords: molecular evolution • bootstrap • multiple sequence alignment • amino acid substitution matrix • evolutionary relationship • systematics

INTRODUCTION

Phylogenetic analysis is a standard and essential tool in any molecular biologist's bioinformatics toolkit that, in the context of protein sequence analysis, enables us to study the evolutionary history and change of proteins and their function. Such analysis is essential to understanding major evolutionary questions, such as the origins and history of macromolecules, developmental mechanisms, phenotypes, and life itself. On a more practical level, phylogenetic analysis of protein sequence data is integral to gene annotation, prediction of gene function, the identification and construction of gene families, and gene discovery.

Phylogenetic trees are mathematical structures that depict the evolutionary history of a group of organisms or genes. The aim of phylogenetic trees is to depict historical (i.e., evolutionary) relationships, and not degree of similarity. For example, although lizards and crocodiles look more similar to each other than to humans, crocodiles are evolutionarily closer to humans because the last common crocodile-human ancestor was more recent than the last common crocodile–lizard ancestor. Similarly, the lysozyme protein of the colobus monkey exhibits 14 amino acid differences when compared to either the human or the baboon lysozyme protein (Stewart et al., 1987), even though humans diverged from the baboon-colobus monkey lineage almost 25 million years ago, whereas baboons and colobus monkeys diverged less than 15 million years ago (Sterner et al., 2006). Clearly, degree of sequence similarity does not equate with degree of evolutionary relationship.

A typical phylogenetic analysis of protein sequence data involves five distinct steps: (a) data collection, (b) inference of homology, (c) sequence alignment, (d) alignment trimming, and (e) phylogenetic analysis. Although this unit concentrates only on the last step, the first four steps are critical to accurate inference and are thus also worthy of brief discussion.

Data collection

The sources of protein sequence data used for phylogenetic analysis are very diverse. For example, data may be generated via PCR, cloning, and DNA sequencing of a particular locus (or loci) across several species (Murphy et al., 2001; Rokas et al., 2002; James et al., 2006) or a particular gene family across a genome (Garcia-Fernandez and Holland, 1994) and then translated into amino acid sequences. Alternatively, protein sequence data may be collected from high-throughput DNA sequencing experiments, such as EST- and RNA-sequencing (Dunn et al., 2008; Hittinger et al., 2010), or from whole-genome data (Rokas et al., 2003; Ciccarelli et al., 2006; Fitzpatrick et al., 2006).

Current Protocols in Molecular Biology 19.11.1-19.11.14, October 2011
Published online October 2011 in Wiley Online Library (wileyonlinelibrary.com).
DOI: 10.1002/0471142727.mb1911s96

Homology inference

Virtually every phylogenetic analysis of molecular data assumes that the proteins under study are homologous; that is, every analysis assumes that all proteins studied are related by descent to the same ancestral protein. It is only after homologs have been inferred (or assumed) that phylogenetic analysis can be performed. Depending on the question asked, homolog inference can take many forms. For example, in studies aimed at understanding the evolution of a certain gene family, homolog inference is typically performed by conducting similarity search analyses with local alignment search algorithms, such as BLAST (*UNIT 19.3*). Alternatively, in studies focused on reconstructing species histories from gene histories, homology inference requires inference of orthologs (homologs that have originated via speciation) using more complex search strategies (Remm et al., 2001; Li et al., 2003; Wall et al., 2003; Alexeyenko et al., 2006; Kuzniar et al., 2008; Salichos and Rokas, 2011).

Sequence alignment

Once the set of homologous proteins has been identified, sequences are typically aligned globally, that is, across their entire length, to construct a multiple sequence alignment (MSA). In contrast to local alignment search algorithms like BLAST (*UNIT 19.3*), where the objective is the accurate identification of homologs, MSA algorithms focus on accurately aligning all the individual amino acids across all the sequences. The industry classic for MSA is the CLUSTAL family of programs (Larkin et al., 2007). However, in recent years, a new generation of much faster and much more accurate programs, such as MAFFT (Katoh et al., 2002; Katoh and Toh, 2008), T-COFFEE (Notredame et al., 2000), and PRANK (Loytynoja and Goldman, 2008, 2010) have been developed.

Alignment trimming

MSAs constructed for many proteins contain regions that are aligned poorly. In several cases, removal of such poorly aligned regions has been shown to improve phylogenetic inference (Talavera and Castresana, 2007), which has resulted in the common practice of "trimming" such poorly aligned regions from protein MSAs prior to phylogenetic inference. Popular programs for MSA trimming include G-BLOCKS (Castresana, 2000) and TRIMAL (Capella-Gutierrez et al., 2009).

BRIEF INTRODUCTION TO PHYLOGENETIC ANALYSIS

Once a set of protein sequences has been aligned, the resulting MSA can be entered directly into a phylogenetic analysis. There are several different methods and protocols for molecular phylogenetic analysis (Swofford et al., 1996; Li, 1997; Kitching et al., 1998; Page and Holmes, 1998; Nei and Kumar, 2000; Huelsenbeck et al., 2001; Felsenstein, 2003). This abundance of methods means that a novice user will have to make numerous decisions and choices at several different steps and levels during analysis, which may vary from one data set to another.

The aim of any phylogenetic analysis is to identify which tree, out of all possible trees, best estimates the true evolutionary history of the protein sequence data analyzed. At the most fundamental level, this estimation of phylogenetic relationships involves two decisions. The first decision is which *optimality criterion* should be used. Given a set of alternative phylogenetic trees, the optimality criterion allows the user to decide which tree explains or fits the data better. There are several different optimality criteria including, but not limited to, maximum likelihood, Bayesian inference, and parsimony (for detailed descriptions of these and other optimality criteria see Swofford et al., 1996; Huelsenbeck et al., 2001). For example, under the parsimony optimality criterion, the best phylogenetic tree is the one that requires the smallest number of evolutionary changes.

The second decision is the choice of *search strategy* for exploration of tree space (for a detailed, but remarkably lucid, description of the different search strategies see Swofford et al., 1996). It so happens that one cannot typically estimate the best tree among all possible trees for a set of protein sequences, for two reasons. First, because the number of possible trees grows exponentially with the number of sequences, the numbers of alternative trees for even small numbers of sequences are extremely large. For example, the number of different phylogenetic trees that can depict the evolutionary relationships of 50 sequences is nearly as large as the number of atoms in the known universe (Stamatakis et al., 2007). Second, it has been proven that efficient solutions to the computational problem of finding the best phylogenetic tree do not exist (Day et al., 1986; Chor and Tuller, 2005), and search of the near-entire tree space is required for accurate identification of the best tree. Because *exhaustive* evaluation of such large numbers of trees is unfeasible for data sets that contain

a dozen sequences or more, phylogeneticists have devised a number of different *heuristic* search strategies for identifying the best tree. Although these heuristic search strategies are very accurate and much faster than an exhaustive search, they are not guaranteed to find the best tree.

The standard practice in molecular phylogenetics is to analyze each data set using several different optimality criteria (maximum likelihood, Bayesian inference, and parsimony are the three most popular). Selection of a particular search strategy is typically determined by computational feasibility considerations. Exhaustive searches on data sets with more than a dozen sequences are still prohibitively time-consuming irrespective of which optimality criterion is used; however, it is now customary to use the most rigorous heuristic search strategies available on data sets containing hundreds or thousands of sequences. Once the user has chosen which *optimality criterion* and *search strategy* to employ on a given data set, a series of trees is generated and evaluated, always keeping track of the 'best' tree(s) examined in the course of the search of tree space. Once the search reaches the point where a better tree cannot be found, the search ends, and the 'best' tree becomes the best estimate of the evolutionary history of the data set analyzed.

As with any other type of statistical analysis, phylogenetic analysis allows for many different options and many different ways to analyze protein sequence data. This unit describes how to perform a set of standard phylogenetic analyses on protein sequence data using the RAXML program (Stamatakis, 2006; Stamatakis et al., 2005, 2008), and how to interpret the results. All analyses described here use the maximum likelihood optimality criterion and a specific search strategy (see below), both of which are state-of-the-art and highly accurate. Nevertheless, depending on the data analyzed and the question(s) asked, the reader should be aware that publication of phylogenetic trees in most journals typically requires several different analyses using several different programs. It is important to demonstrate agreement in results obtained from application of different optimality criteria and from several measures of robustness of inference.

Phylogenetic analysis using the maximum likelihood optimality criterion

The concept of likelihood has a long tradition in the field of statistical inference and has many applications in biological research (Edwards, 1992). Briefly, in the context of phylogenetic analysis, the maximum likelihood optimality criterion states that the phylogenetic tree that makes a given sequence data set most likely constitutes the maximum likelihood estimate of the phylogeny and is the preferred explanation (Page and Holmes, 1998). Formally, the likelihood score L_D of a sequence data set D for phylogenetic hypothesis H can be estimated by calculating the probability of D given H, or $L_D = \Pr(D|H)$. It should be noted that H does not only correspond to the phylogenetic tree but also to the probabilistic model of sequence evolution used in phylogenetic reconstruction. Importantly, the likelihood criterion not only enables us to directly estimate the parameters in the model of sequence evolution, but also to identify their optimal values for the data set analyzed (the optimal values are the ones that maximize the likelihood).

The model of sequence evolution involves several parameters that describe how the sequences in a given data set evolve, such as the rates of substitution between amino acids, the frequencies of amino acids, and the heterogeneity in rate of evolution across sites of the MSA. The overwhelming majority of protein sequence phylogenetic analyses use empirically derived amino acid substitution matrices, whose rates are fixed to specific values estimated from large numbers of real protein MSAs (Whelan et al., 2001). For example, the RTREV substitution matrix is derived from virally-encoded amino acid data (Dimmic et al., 2002), whereas CPREV is derived from chloroplast-encoded amino acid data (Adachi et al., 2000). In addition to the chosen substitution matrix, the user can typically decide whether to use in phylogenetic estimation the empirical frequencies of the amino acids in the data set (or the ones calculated during matrix construction), as well as whether to allow for any variation in the rate of evolution across sites of the protein MSA. Most proteins show substantial heterogeneity in the rate of evolution across their sequence (sites critical to function tend to be highly conserved, whereas others tend to be much more variable), so explicitly accounting for this rate heterogeneity among sites in the specification of the model of sequence evolution is generally a very good idea. One standard and very popular approach for incorporating rate heterogeneity among sites into the phylogenetic analysis uses the gamma distribution to approximate the distribution of rates in a protein MSA (Yang, 1996). This distribution is very suitable for this task

because, depending on the value of its shape parameter α, it can be either L-shaped (appropriate for MSAs that exhibit extreme rate heterogeneity) or bell-shaped (appropriate for MSAs that exhibit minor rate heterogeneity).

The RAXML program

The RAXML (Randomized Axelerated Maximum Likelihood) program has been developed to perform both sequential (on a single processor) and parallel (on multiple processors) phylogenetic analysis using the maximum likelihood optimality criterion. Historically, RAXML stems from the FASTDNAML program (Olsen et al., 1994), which in turn stems from the DNAML program (Felsenstein, 1993). Although RAXML's design emphasis is on computationally efficient and biologically accurate analysis of very large data sets, it is also appropriate for and amenable to the analysis of data sets of any size. RAXML can use a variety of different character sets, including nucleotide, amino acid, binary, and multi-state character state data.

Versions of the RAXML program are available for the Unix/Linux, Mac, and Windows operating systems (from *http://wwwkramer.in.tum.de/exelixis/software.html*; Stamatakis, 2006), as well as from two Web servers (from the Swiss Institute of Bioinformatics at *http://phylobench.vital-it.ch/raxml-bb/*, and from the CIPRES Science Gateway at *http://www.phylo.org/portal2/*; Stamatakis et al., 2008). The stand-alone version is command-line based, but Graphical User Interface fronts are also available (from *http://sourceforge.net/projects/raxmlgui/* and *http://sourceforge.net/projects/wxraxml/*).

The RAXML search strategy. The first step of the search strategy employed by RAXML is the generation of a starting tree. This starting tree is constructed by adding the sequences one by one in random order, and identifying their optimal location on the tree under the parsimony optimality criterion (Stamatakis et al., 2005). The random order in which sequences are added is likely to generate several different starting trees every time a new analysis is

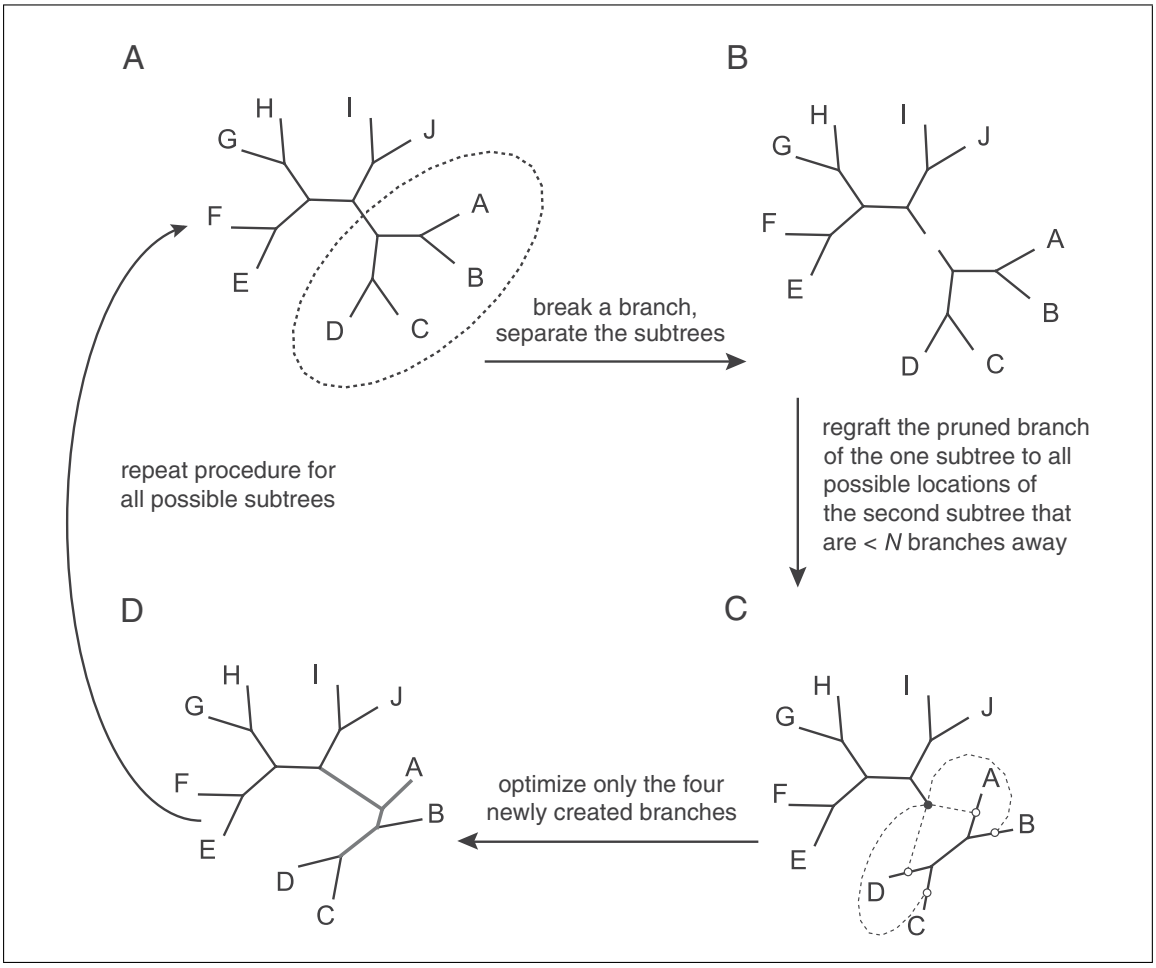

Figure 19.11.1 The lazy subtree rearrangement (LSR) tree search strategy.

run (especially for data sets with more than a few sequences), which allows better exploration of the tree space. If multiple analyses using different starting trees all converge on the same best tree, then confidence that this is the true best tree increases. The second step of the search strategy involves a method known as *lazy subtree rearrangement* or *LSR* (Stamatakis et al., 2005; Schmidt and von Haeseler 2009), which is summarized in Figure 19.11.1. Briefly, under LSR, all possible subtrees of a tree are clipped and reinserted at all possible locations as long as the number of branches separating the clipped and insertion points is smaller than *N* branches. RAXML estimates the appropriate *N* value for a given data set automatically, but one can also run the program with any fixed value. The LSR method is first applied on the starting tree, and subsequently multiple times on the currently best tree as the search continues, until no better tree is found.

PHYLOGENETIC ANALYSIS USING THE RAXML PROGRAM

The RAXML command line interface

Irrespective of operating system used, the typical way to perform phylogenetic analysis with RAXML is using the command line. Invoking the RAXML command by typing

```
RAXMLHPC -h
```

and hitting Enter at the terminal cursor (typically indicated by the `>`, `$`, or `%` signs) prints the program version, command line options, and author and contact information.

Like most command line programs, RAXML can be executed by typing the command invoking the program (i.e., `RAXMLHPC`) followed by a series of options. In the example above, the `-h` option displays a long help message that describes the multitude of options available via the program. The user can use these options to specify the data to be analyzed as well as to set and control the different parameters of the analysis.

Input format for protein alignments

The RAXML program accepts protein sequence alignments in the PHYLIP format. An example alignment of part of the mitochondrial cytochrome oxidase subunit II alignment in the PHYLIP format is shown in Figure 19.11.2 (it can be downloaded from *http://wwwkramer.in.tum.de/exelixis/hands-on/protein.phy* and is also available as a supplementary file of this unit). Generation of PHYLIP formatted alignments is a standard feature of many different alignment programs and sequence editors (e.g., *http://www-bimas.cit.nih.gov/molbio/readseq/*).

Constructing a maximum likelihood tree with RAXML

We can set a simple maximum likelihood analysis in RAXML by typing:

```
RAXMLHPC -s protein.phy -n A1
-m PROTGAMMAWAG
```

The option `-s protein.phy` specifies the sequence data file to be analyzed. The option `-n A1` specifies the file name appendix that will be added to all the output files produced by RAXML in this run, which will be in the format `RAXML_filename.A1`. Although RAXML will not overwrite previous results (the second time you use the same file name appendix the program will not run any analysis but will ask you to provide a different appendix), it is best practice to use a different file name appendix for every run. The option `-m PROTGAMMAWAG` specifies to RAXML three parameters associated with the model of sequence evolution employed: first, that we are using protein data (the `PROT` part); second, that we are accounting for rate heterogeneity among sites in our alignment by using the gamma distribution (the `GAMMA` part); and third, that we are employing the Whelan and Goldman (Whelan and Goldman, 2001) amino acid substitution matrix (the `WAG` part).

If we wanted to choose a different amino acid substitution matrix (e.g., the `RTREV` matrix), it would be necessary to simply replace the `WAG` part of the `-m` option with `RTREV`. To use empirical base frequencies drawn from the alignment (rather than use the pre-defined base frequencies that come with the matrix), all that is needed is to add the letter `F` to the `-m` option so that the RAXML command now looks like:

```
RAXMLHPC -s protein.phy -n A2
-m PROTGAMMARTREVF
```

There are a few different ways of deciding what amino acid substitution matrix should be used. As discussed above, these empirical amino acid substitution matrices are derived from several different sets of protein MSAs. One reasonable choice is to use a model that derives from data that are most similar to the data at hand. For example, for our mitochondrial sequence

```
10 50
Cow          MAYPMQLGFQ DATSPIMEEL LHFHDHTLMI VFLISSLVLY IISLMLTTKL
Carp         MAHPTQLGFK DAAMPVMEEL LHFHDHALMI VLLISTLVLY IITAMVSTKL
Chicken      MANHSQLGFQ DASSPIMEEL VEFHDHALMV ALAICSLVLY LLTLMLMEKL
Human        MAHAAQVGLQ DATSPIMEEL ITFHDHALMI IFLICFLVLY ALFLTLTTKL
Loach        MAHPTQLGFQ DAASPVMEEL LHFHDHALMI VFLISALVLY VIITTVSTKL
Mouse        MAYPFQLGLQ DATSPIMEEL MNFHDHTLMI VFLISSLVLY IISLMLTTKL
Rat          MAYPFQLGLQ DATSPIMEEL TNFHDHTLMI VFLISSLVLY IISLMLTTKL
Seal         MAYPLQMGLQ DATSPIMEEL LHFHDHTLMI VFLISSLVLY IISLMLTTKL
Whale        MAYPFQLGFQ DAASPIMEEL LHFHDHTLMI VFLISSLVLY IITLMLTTKL
Frog         MAHPSQLGFQ DAASPIMEEL LHFHDHTLMA VFLISTLVLY IITIMMTTKL
```

Figure 19.11.2 An example alignment of part of the mitochondrial cytochrome oxidase subunit II alignment in the PHYLIP format.

alignment, we could choose `MTREV` (Adachi and Hasegawa, 1996), an empirical substitution matrix estimated from the complete mitochondrial sequence data of 20 vertebrate species. A much more thorough, but computationally much more demanding approach is to use the program PROTTEST (Abascal et al., 2005), which calculates several different statistics to identify which model best fits the data. PROTTEST can be run online on a protein MSA from *http://darwin.uvigo.es/software/prottest_server.html*, although a stand-alone version of the program is also available (for a detailed theoretical and practical guide on the program see Posada, 2009). Finally, instead of using one of the standard models available, the user can actually estimate the amino acid model based on the amino acid data at hand, by replacing the `WAG` part with `GTR`, and typing:

```
RAXMLHPC -s protein.phy -n A3
-m PROTGAMMAGTR
```

Note, however, that one should employ this option only on protein alignments that contain thousands of amino acid columns, because only those contain sufficient data to estimate all possible amino acid substitution parameters.

As mentioned above, RAXML generates a starting tree by adding the sequences one by one in random order and inferring the best starting tree using the parsimony optimality criterion. Thus, each time RAXML is run, a different starting tree is generated. Because, like all heuristic search strategies, the LSR search strategy employed by RAXML is not guaranteed to find the best tree, it is customary to conduct multiple searches for the best tree. If all searches that begin from different starting trees converge on the same best tree, then the researcher's confidence that the inferred tree is the best increases. To conduct multiple searches for the best tree, it is necessary to add the option `-# n`, where `n` is the desired number of multiple searches to perform. Thus, if the goal is to perform 10 searches, the RAXML command should look like:

```
RAXMLHPC -s protein.phy -n A4
-m PROTGAMMAWAGF -# 10
```

Visualizing the maximum likelihood tree

Examination of the contents of the directory where the different analyses were run shows that RAXML generated several different files (the number and type of files generated will vary depending on the option settings specified) from the several different analyses. These files provide detailed information and results about the analysis (e.g., the `RAXML_info.A1` file), the maximum likelihood tree (e.g., the `RAXML_bestTree.A1` file), the starting parsimony tree (e.g., the `RAXML_parsimonyTree.A1` file), etc. A full description of the contents of each file can be found in the program's manual (available from *http://wwwkramer.in.tum.de/exelixis/oldPage/RAxML-Manual.7.0.4.pdf*). Typically, the most useful output files are the tree files, which are written in the NEWICK format (a full description of the format can be found at *http://evolution.genetics.washington.edu/phylip/newicktree.html*), and can be opened for viewing in any of several different tree visualization programs. For example, Figure 19.11.3 shows a screenshot of `RAXML_bestTree.A1` file when opened with the tree visualization program FIGTREE (*http://tree.bio.ed.ac.uk/software/figtree/*).

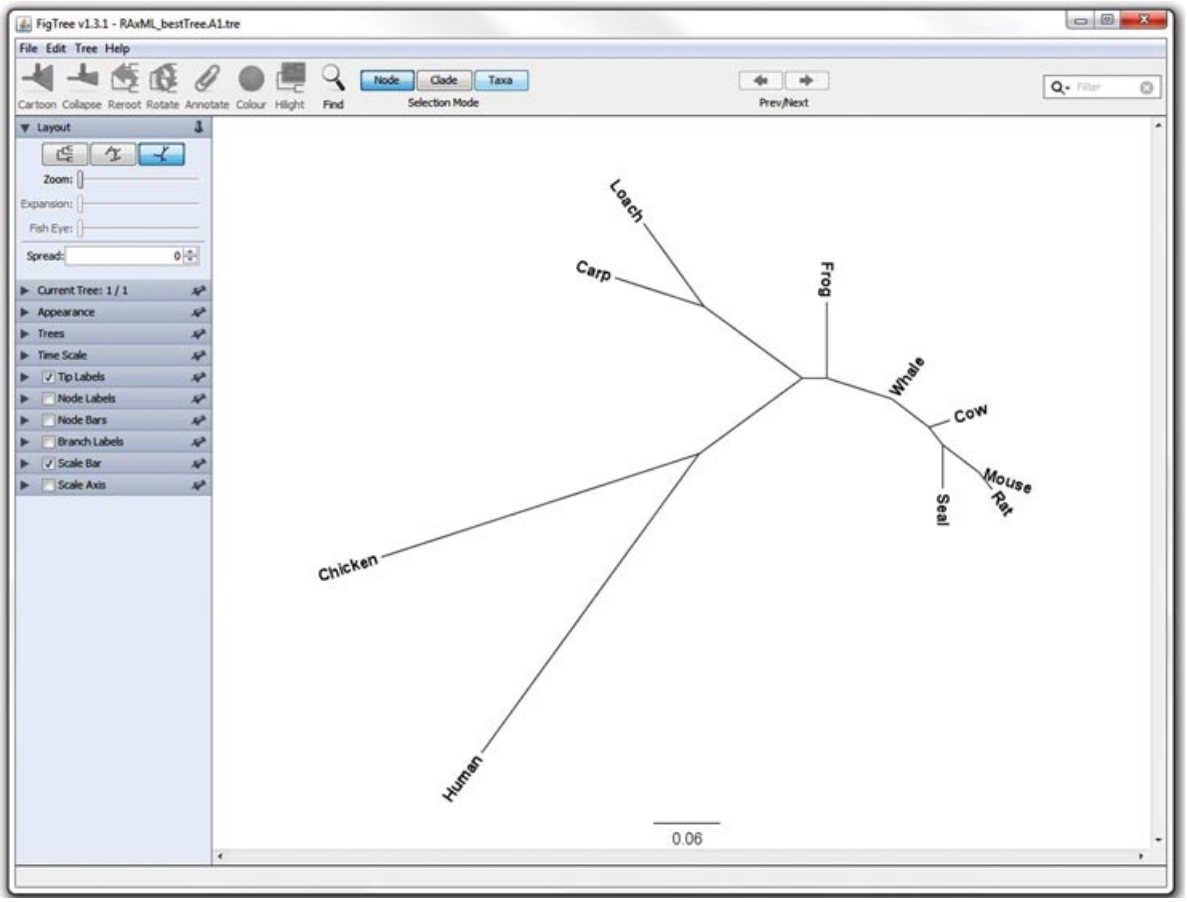

Figure 19.11.3 Screenshot of the (unrooted) maximum likelihood tree in the FIGTREE program. Branch lengths are in substitutions per site.

Rooting the phylogenetic tree

Phylogenetic trees can be either rooted or unrooted. Rooted phylogenetic trees have direction, since all lineages depicted on the tree originate from the same common ancestor, which is also known as the root of the tree. In contrast, unrooted trees lack a root and, consequently, do not inform us about the direction of evolution. Examples of rooted and unrooted phylogenetic trees are shown in Figure 19.11.4.

Although the majority of biologists want to obtain and work with rooted phylogenetic trees, the overwhelming majority of molecular phylogenetic programs produce unrooted phylogenetic trees. To produce a rooted phylogenetic tree, the user must include in the set of sequences to be analyzed a sequence from a species that is known, based on independent evidence (e.g., from paleontological data), to have diverged prior to the origin of our set of sequences. Such a sequence is known as an *outgroup*. In RAXML, one can specify an outgroup by adding the `-o sequence` option, where `sequence` is the name of one (or more) of the sequences in the multiple alignment that we want to use as the outgroup. Thus, if we want to use the `Carp` sequence as the outgroup, the RAXML command should look like:

```
RAXMLHPC -s protein.phy -n A5
-m PROTGAMMAWAGF -o Carp
```

The outgroup can also consist of more than one sequence. For example, in our data set, one may want to root the phylogenetic tree using the two fish sequences (`Carp` and `Loach`) as the outgroup (as shown in panel B of Fig. 19.11.4), in which case the RAXML command should look like:

```
RAXMLHPC -s protein.phy -n A6
-m PROTGAMMAWAGF -o Carp,Loach
```

Note that if the sequences specified as the outgroup do not form a monophyletic group (i.e., a group of sequences descended from the same common ancestral sequence not shared with any other group of sequences), RAXML

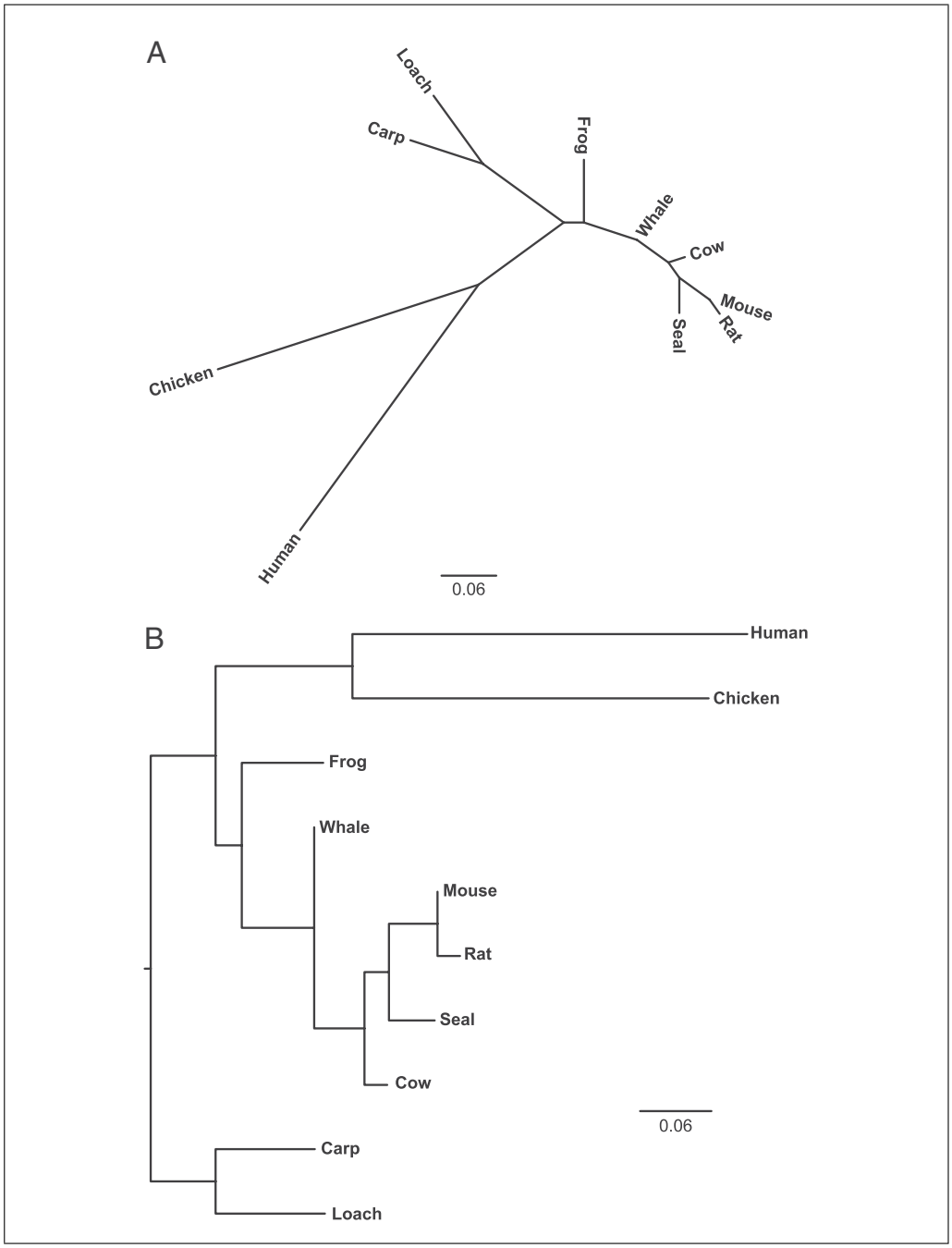

Figure 19.11.4 Examples of an unrooted (**A**) and a rooted (**B**) phylogenetic tree for the example data set. The unrooted tree on the top panel, if rooted on the branch leading to the `Carp` and `Loach` sequences, corresponds to the rooted tree on the bottom. Note that in the unrooted tree neighboring sequences are not necessarily closely related.

will be unable to place all of them as the outgroup. In this event, RAXML will print a warning in the RAXML information file (in this example, this is the `RAXML_info.A6` file) and proceed to root the phylogeny using the first of the sequences specified in the `-o` option as the outgroup.

Assessing robustness of inference

The standard statistical approach for assessing robustness in the inference of phylogenetic relationships uses a technique known as bootstrapping (Felsenstein, 1985). In bootstrapping, one generates multiple data sets that have the same number of alignment columns as the original data set by randomly sampling the original alignment columns with replacement. Each bootstrap replicate data set is then analyzed in exactly the same way as the original data set, which results in the production of a maximum likelihood tree from each bootstrap replicate data set. The frequency of

occurrence of any given grouping on the set of bootstrap trees (which is known as the bootstrap support value) signifies the measure of support for that particular grouping in our data (Soltis and Soltis, 2003).

To conduct a bootstrap analysis in RAXML, it is necessary to specify two additional options. The first is the `-b n` option, where `n` can be any positive integer, and which specifies the random number seed required for the bootstrap analysis. Using the same random seed number in different runs of the same data set will result in the generation of identical bootstrap replicate data sets, so it may be desirable to pick a new random seed number every time an analysis is run. The second option is the `-# n` option, where `n` can be any positive integer, and which specifies the number of bootstrap replicates to be performed. After specifying these two options, the RAXML command should look like:

```
RAXMLHPC -s protein.phy -n A7
-m PROTGAMMAWAGF —b 0123 -#
100
```

The typical number of bootstrap replicates performed varies greatly between studies, and can range from a hundred to thousands of replicates. Because the number of bootstrap replicates required to obtain bootstrap support values of high quality varies with the type and size of data set analyzed (Pattengale et al., 2010), RAXML allows the user to automatically estimate when an appropriate number of bootstrap replicates has been performed through the use of several different *stopping* criteria. The logic underlying all these criteria is the same; after every 50 bootstrap replicates, the program performs 100 random splits of the bootstrap replicate set into two halves and computes statistics, which vary depending on the criterion implemented. For example, the frequency-based criterion, which is specified by setting the `-#` option to `autoFC`, determines whether enough replicates have been performed by calculating the Pearson and Sierk correlation coefficient (2005) in the two halves from the 100 splits. Bootstrapping stops if there are at least 99 splits whose halves show a correlation coefficient greater than 0.99. Thus, the RAXML command implementing the frequency-based stopping criterion should look like:

```
RAXMLHPC -s protein.phy -n A8
-m PROTGAMMAWAGF —b 0123 -#
autoFC
```

Running this command, RAXML calculates that 800 bootstrap replicates are sufficient for high-quality bootstrap values and saves the results in the `RAXML_bootstrap.A8` file (the final number of bootstrap replicates will vary between runs, even if the same random seed is used, because the statistics are calculated on random splits).

It is customary to visualize bootstrap support values on the maximum likelihood tree. Therefore, we can use the maximum likelihood tree generated from the `A4` analysis (see above), which was saved in the `RAXML_bestTree.A4` file, as the tree on which to display the bootstrap support values. We can instruct RAXML to draw bootstrap values on the maximum likelihood tree using the following command:

```
RAXMLHPC -n A9 -m
PROTGAMMAWAGF -f b -
t RAXML_bestTree.A4 -z
RAXML_bootstrap.A8
```

Here, the `-f b` option specifies the analysis to be performed (draw bootstrap support values on a given tree), the `-t RAXML_bestTree.A4` option specifies the tree that we want the values depicted on, whereas the `-z RAXML_bootstrap.A8` option specifies the file containing the trees generated via bootstrapping.

Once the command is executed, the tree containing the bootstrap support values drawn on the maximum likelihood tree will be saved in the `RAXML_bipartitions.A9` file (Fig. 19.11.5). One can also use the set of trees produced from the bootstrap replicates to construct various kinds of *consensus* trees that summarize their agreements. For example, *strict* consensus trees contain only those groupings present in all bootstrap replicate trees, whereas *majority rule* consensus trees contain only those groupings that are present in more than half of the bootstrap replicate trees. Consensus tree construction in RAXML uses the `-J` option. For example by setting `-J STRICT`, one can construct a strict consensus tree (Fig. 19.11.5):

```
RAXMLHPC -n A10 -m PROTGAM-
MAWAGF -J STRICT -z
RAXML_bootstrap.A8
```

whereas by setting `-J MR` one can construct a majority rule consensus tree (Fig. 19.11.5):

```
RAXMLHPC -n A11 -m
PROTGAMMAWAGF -J MR -z
RAXML_bootstrap.A8
```

Standard bootstrapping can be computationally very demanding, especially for larger

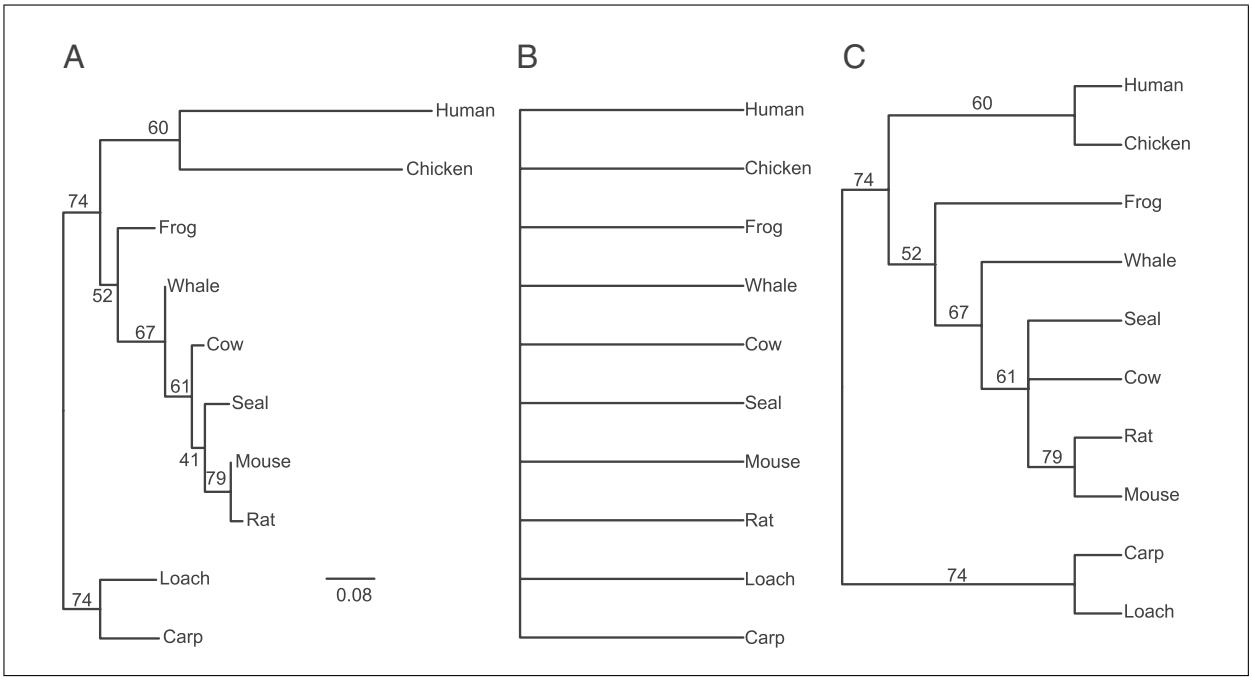

Figure 19.11.5 Different ways of visualizing bootstrap support values on phylogenetic trees. (**A**) Bootstrap support values depicted on the maximum likelihood tree. (**B**) The strict consensus tree, which is completely unresolved because none of the groupings was present in all bootstrap replicate trees. (**C**) Bootstrap support values depicted on the majority rule tree.

data sets. To facilitate faster analysis, the RAXML program also contains a rapid bootstrapping algorithm that is at least an order of magnitude faster than the standard one, while similarly accurate (Stamatakis et al., 2008). To run an analysis using this algorithm, the user need simply change the `-b n` option to the `-x n` option, where `n` can be any positive integer. After replacing the `-b n` with the `-x n` option, the RAXML command should look like:

```
RAXMLHPC -s protein.phy -n A12
-m PROTGAMMAWAGF —x 0123 -#
100
```

or, if the frequency-based stopping criterion is to be implemented, it should look like:

```
RAXMLHPC -s protein.phy -n A13
-m PROTGAMMAWAGF —x 0123 -#
autoFC
```

For comparison, in a standard 2.3-GHz processor, the 100 bootstrap replicates performed for the `A7` analysis took ~266 sec to run, whereas the 100 rapid bootstrap replicates performed for the `A12` analysis took ~74 sec.

One useful feature of the program is that it allows the user to simultaneously perform maximum likelihood and rapid bootstrapping analysis by adding the `-f a` option, so that the RAXML command looks like:

```
RAXMLHPC —f a -s protein.phy
-n A14 -m PROTGAMMAWAGF —x
0123 -# 100
```

Execution of the command will generate the maximum likelihood tree file (`RAXML_bestTree.A14`) and the bootstrap replicate tree file (`RAXML_bootstrap.A14`), as well as the tree file containing the bootstrap support values drawn on the maximum likelihood tree (`RAXML_bipartitions.A14`).

Comparing different phylogenetic trees

Researchers often want to test directly different phylogenetic hypotheses. For example, one frequent question is whether a phylogenetic tree obtained from a given protein alignment is significantly different from the traditional phylogeny. Such questions can be addressed by performing tests that evaluate whether the likelihood scores of different phylogenetic trees are significantly different. One such very useful and frequently used test is the *Shimodaira-Hasegawa test* (or SH test), which examines whether the maximum likelihood tree is significantly better than user-supplied phylogenetic trees (Shimodaira and Hasegawa, 1999).

For example, examination of the maximum likelihood tree estimated from our data set

(Fig. 19.11.3) shows that the human sequence groups with the chicken sequence and not with the other mammal sequences, as one would expect based on the vertebrate phylogeny. In this case, we can use the SH test to evaluate whether the maximum likelihood tree is significantly better than the traditional vertebrate tree using the following RAXML command:

```
RAXMLHPC -f h -s protein.phy
-n A15 -m PROTGAMMAWAGF -t
RAXML_bestTree.A4 -z verte-
brate.tree
```

In this command, the '-f h' option specifies that we want to perform an SH test. Similar to several previous commands, the maximum likelihood tree is specified by the -t RAXML_bestTree.A4 option, and the vertebrate tree by the -z vertebrate.tree option. This latter file can be created by writing the vertebrate phylogeny for the 10 sequences used in this data set in the NEWICK format:

```
(((((Human,(Mouse,Rat)),
((Cow,Whale),Seal)),Chicken),
Frog),Carp,Loach);
```

The analysis produces a single output file (RAXML_info.A15) that reports the results of the SH test in its last few lines:

```
Model optimization, best Tree:
−411.163389

Found 1 trees in File verte-
brate.tree

Tree: 0 Likelihood:
−423.777863 D(LH): −12.614474
SD: 6.714754

Significantly Worse: No (5%),
No (2%), No (1%)
```

The best Tree: −411.163389 text reposrts the likelihood score (after logarithmic transformation) of the best tree, whereas the Tree: 0 Likelihood: −423.777863 text reports the likelihood score of the vertebrate tree. The D(LH): −12.614474 SD: 6.714754 text reports the difference D(LH) in the likelihood scores between the two trees and its standard deviation SD. Finally, the last line reports whether this difference in likelihood between the best tree and the vertebrate tree is significant at the 5%, 2%, and 1% level. Given that the reported result is No for all three levels of significance, it can be concluded that the maximum likelihood phylogenetic tree estimated from this data set does not significantly differ from the standard vertebrate phylogeny.

Analyzing multiple data sets

Phylogenies based on single proteins are often unreliable or lack the phylogenetic signal necessary for successfully inferring phylogenies (Rokas et al., 2003). Consequently, in recent years, researchers have been increasingly analyzing multiple data sets. If the user constructs a single data matrix that contains the alignments of both proteins and provides RAXML with information about the boundaries of the different data sets, such analyses of multiple data sets can be performed in RAXML. Importantly, RAXML allows the user to specify different models of sequence evolution for each data set and optimizes parameters separately for each data set.

For example, let us hypothesize that our example data set is actually a composite of two different protein alignments, with amino acid columns 1-30 corresponding to protein A and amino acid columns 31-50 corresponding to protein B. We can inform RAXML that our sequence file is a composite of two different data sets by creating a plain text file (let us name it partition.txt) that contains the following text:

```
WAGF, proteinA = 1-30
RTREVF, proteinB = 31-50
```

In this file, each line describes each protein in the data set. The first part of each line (WAGF in the first and RTREVF in the second) describes the amino acid substitution matrix we have chosen to use for each of the proteins. The second part (proteinA = 1-30 in the first and proteinB = 31-50 in the second) describes the names of the two proteins, which are arbitrary, as well as the multiple sequence alignment columns they occupy (the first 30 amino acid columns are from the alignment of protein A, whereas the last twenty for the alignment of protein B). Somewhat confusingly, executing a multiple data set analysis in RAXML requires that we also specify the model of sequence evolution, the −m option. Although the models of amino acid evolution specified in the partition.txt file take precedence over the model specified via the −m option, this option is still necessary and useful in that it allows us to specify if we want to account for rate heterogeneity among sites in the two proteins. Thus, by setting -m PROTGAMMAGTR, the RAXML command should look like:

```
RAXMLHPC -s protein.phy -n
A16 -m PROTGAMMAGTR -q parti-
tion.txt
```

Using this command, RAXML will use the WAG model for protein A and the RTREV model for protein B, as specified in the partition.txt file (and ignore the GTR model specified via the —m option). It will also use the empirical amino acid frequencies of each protein (because we specified so in the partition.txt file), and estimate the degree of rate heterogeneity independently for each protein (because we specified so in the -m option). Examination of the RAXML output indicates that the program analyzes each protein separately, but produces a *single* maximum likelihood tree (RAXML_bestTree.A16) that summarizes the results from the analysis of the two proteins.

CONCLUDING REMARKS

The theory and practice of phylogenetic analysis of sequence data has blossomed in the last three decades. As such, any protocol aimed at describing how to perform a set of analyses is bound to serve as an introduction to this rather complex field, rather than as a full description of the state-of-the-art methods of analysis. Readers interested in delving deeper into the theory and practice of molecular phylogenetics are advised to consult any of the several excellent and more in depth descriptions of the theory and practice of phylogenetic inference (Swofford et al., 1996; Page and Holmes, 1998; Nei and Kumar, 2000; Felsenstein, 2003; Salemi et al., 2009), as well as explore several different optimality criteria and programs for phylogenetic analysis (Swofford, 2002; Zwickl, 2006; Drummond and Rambaut, 2007; Guindon et al., 2010). A remarkably up to date list of phylogeny programs can be found at *http://evolution.genetics.washington.edu/phylip/software.html*.

LITERATURE CITED

Abascal, F., Zardoya, R., and Posada, D. 2005. Prottest: Selection of best-fit models of protein evolution. *Bioinformatics* 21:2104-2105.

Adachi, J. and Hasegawa, M. 1996. Model of amino acid substitution in proteins encoded by mitochondrial DNA. *J. Mol. Evol.* 42:459-468.

Adachi, J., Waddell, P.J., Martin, W., and Hasegawa, M. 2000. Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. *J. Mol. Evol.* 50:348-358.

Alexeyenko, A., Tamas, I., Liu, G., and Sonnhammer, E.L. 2006. Automatic clustering of orthologs and inparalogs shared by multiple proteomes. *Bioinformatics* 22:E9-E15.

Capella-Gutierrez, S., Silla-Martinez, J.M., and Gabaldon, T. 2009. trimAl: A tool for automated alignment trimming in large-scale phylogenetic analyses. *Bioinformatics* 25:1972-1973.

Castresana, J. 2000. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Mol. Biol. Evol.* 17:540-552.

Chor, B. and Tuller, T. 2005. Maximum likelihood of evolutionary trees: Hardness and approximation. *Bioinformatics* 21:97-106.

Ciccarelli, F.D., Doerks, T., von Mering, C., Creevey, C.J., Snel, B., and Bork, P. 2006. Toward automatic reconstruction of a highly resolved tree of life. *Science* 311:1283-1287.

Day, W.H.E., Johnson, D.S., and Sankoff, D. 1986. The computational complexity of inferring rooted phylogenies by parsimony. *Math. Biosci.* 81:33-42.

Dimmic, M.W., Rest, J.S., Mindell, D.P., and Goldstein, R.A. 2002. rtREV: An amino acid substitution matrix for inference of retrovirus and reverse transcriptase phylogeny. *J. Mol. Evol.* 55:65-73.

Drummond, A.J. and Rambaut, A. 2007. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol. Biol.* 7:214.

Dunn, C.W., Hejnol, A., Matus, D.Q., Pang, K., Browne, W.E., Smith, S.A., Seaver, E., Rouse, G.W., Obst, M., Edgecombe, G.D., Sorensen, M.V., Haddock, S.H., Schmidt-Rhaesa, A., Okusu, A., Kristensen, R.M., Wheeler, W.C., Martindale, M.Q., and Giribet, G. 2008. Broad phylogenomic sampling improves resolution of the animal tree of life. *Nature* 452:745-749.

Edwards, A.W.F. 1992. Likelihood (Expanded Edition). The John Hopkins University Press, Baltimore, Maryland.

Felsenstein, J. 1985. Confidence limits on phylogenies: An approach using the bootstrap. *Evolution* 39:783-791.

Felsenstein, J. 1993. PHYLIP (Phylogeny Inference Package). Distributed by the Author, Department of Genetics, University of Washington, Seattle.

Felsenstein, J. 2003. Inferring Phylogenies. Sinauer, Sunderland, Massachusetts.

Fitzpatrick, D.A., Logue, M.E., Stajich, J.E., and Butler, G. 2006. A fungal phylogeny based on 42 complete genomes derived from supertree and combined gene analysis. *BMC Evol. Biol.* 6:99.

Garcia-Fernandez, J. and Holland, P.W.H. 1994. Archetypal organization of the amphioxus Hox gene cluster. *Nature* 370:563-566.

Guindon, S., Dufayard, J.F., Lefort, V., Anisimova, M., Hordijk, W., and Gascuel, O. 2010. New algorithms and methods to estimate maximum likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.* 59:307-321.

Hittinger, C.T., Johnston, M., Tossberg, J.T., and Rokas, A. 2010. Leveraging skewed transcript

abundance by RNA-Seq to increase the genomic depth of the tree of life. *Proc. Natl. Acad. Sci. U.S.A.* 107:1476-1481.

Huelsenbeck, J.P., Ronquist, F., Nielsen, R., and Bollback, J.P. 2001. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science* 294:2310-2314.

James, T.Y., Kauff, F., Schoch, C.L., Matheny, P.B., Hofstetter, V., Cox, C.J., Celio, G., Gueidan, C., Fraker, E., Miadlikowska, J., Lumbsch, H.T., Rauhut, A., Reeb, V., Arnold, A.E., Amtoft, A., Stajich, J.E., Hosaka, K., Sung, G.H., Johnson, D., O'Rourke, B., Crockett, M., Binder, M., Curtis, J.M., Slot, J.C., Wang, Z., Wilson, A.W., Schussler, A., Longcore, J.E., O'Donnell, K., Mozley-Standridge, S., Porter, D., Letcher, P.M., Powell, M.J., Taylor, J.W., White, M.M., Griffith, G.W., Davies, D.R., Humber, R.A., Morton, J.B., Sugiyama, J., Rossman, A.Y., Rogers, J.D., Pfister, D.H., Hewitt, D., Hansen, K., Hambleton, S., Shoemaker, R.A., Kohlmeyer, J., Volkmann-Kohlmeyer, B., Spotts, R.A., Serdani, M., Crous, P.W., Hughes, K.W., Matsuura, K., Langer, E., Langer, G., Untereiner, W.A., Lucking, R., Budel, B., Geiser, D.M., Aptroot, A., Diederich, P., Schmitt, I., Schultz, M., Yahr, R., Hibbett, D.S., Lutzoni, F., McLaughlin, D.J., Spatafora, J.W., and Vilgalys, R. 2006. Reconstructing the early evolution of Fungi using a six-gene phylogeny. *Nature* 443:818-822.

Katoh, K., and Toh, H. 2008. Recent developments in the MAFFT multiple sequence alignment program. *Brief. Bioinformatics* 9:286-298.

Katoh, K., Misawa, K., Kuma, K., and Miyata, T. 2002. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.* 30:3059-3066.

Kitching, I.J., Forey, P.L., Humphries, C.J., and Williams, D.M. 1998. Cladistics: The Theory and Practice of Parsimony Analysis, 2nd Ed. Oxford University Press, New York.

Kuzniar, A., van Ham, R.C.H.J., Pongor, S., and Leunissen, J.A.M. 2008. The quest for orthologs: Finding the corresponding gene across genomes. *Trends Genet.* 24:539-551.

Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., McGettigan, P.A., McWilliam, H., Valentin, F., Wallace, I.M., Wilm, A., Lopez, R., Thompson, J.D., Gibson, T.J., and Higgins, D.G. 2007. Clustal W and Clustal X version 2.0. *Bioinformatics* 23:2947-2948.

Li, L., Stoeckert, C.J. Jr., and Roos, D.S. 2003. OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Res.* 13:2178-2189.

Li, W-H. 1997. Molecular Evolution. Sinauer, Sunderland, Massachusetts.

Loytynoja, A. and Goldman, N. 2008. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science* 320:1632-1635.

Loytynoja, A. and Goldman, N. 2010. webPRANK: A phylogeny-aware multiple sequence aligner with interactive alignment browser. *BMC Bioinformatics* 11:579.

Murphy, W.J., Eizirik, E., Johnson, W.E., Zhang, Y.P., Ryder, O.A., and O'Brien, S.J. 2001. Molecular phylogenetics and the origins of placental mammals. *Nature* 409:614-618.

Nei, M. and Kumar, S. 2000. Molecular Evolution and Phylogenetics. Oxford University Press, New York.

Notredame, C., Higgins, D.G., and Heringa, J. 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.* 302:205-217.

Olsen, G.J., Matsuda, H., Hagstrom, R., and Overbeek, R. 1994. Fastdnaml: A tool for construction of phylogenetic trees of DNA-sequences using maximum-likelihood. *Comput. Appl. Biosci.* 10:41-48.

Page, R.D.M. and Holmes, E.C. 1998. Molecular Evolution: A Phylogenetic Approach. Blackwell Science, Malden, Massachusetts.

Pattengale, N.D., Alipour, M., Bininda-Emonds, O.R.P., Moret, B.M.E., and Stamatakis, A. 2010. How many bootstrap replicates are necessary? *J. Comput. Biol.* 17:337-354.

Pearson, W.R. and Sierk, M.L. 2005. The limits of protein sequence comparison? *Curr. Opin. Struct. Biol.* 15:254-260.

Posada, D. 2009. Selecting models of evolution. In: The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing (P. Lemey, M. Salemi, and A.M. Vandamme, eds.) pp. 345-361. Cambridge University Press, Cambridge.

Remm, M., Storm, C.E., and Sonnhammer, E.L. 2001. Automatic clustering of orthologs and inparalogs from pairwise species comparisons. *J. Mol. Biol.* 314:1041-1052.

Rokas, A., Nylander, J.A.A., Ronquist, F., and Stone, G.N. 2002. A maximum likelihood analysis of eight phylogenetic markers in gallwasps (Hymenoptera: Cynipidae): Implications for insect phylogenetic studies. *Mol. Phylogenet. Evol.* 22:206-219.

Rokas, A., Williams, B.L., King, N., and Carroll, S.B. 2003. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature* 425:798-804.

Salemi, M., Vandamme, A-M., and Lemey, P. 2009. The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing, 2nd Ed. Cambridge University Press, Cambridge.

Salichos, L. and Rokas, A. 2011. Evaluating ortholog prediction algorithms in a yeast model clade. *PLoS One* 6:e18755.

Schmidt, H.A. and von Haeseler, A. 2009. Phylogenetic inference using maximum likelihood methods. *In* The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing (P. Lemey, M. Salemi, and A.M. Vandamme, eds.) pp. 181-209. Cambridge University Press, Cambridge.

Shimodaira, H. and Hasegawa, M. 1999. Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Mol. Biol. Evol.* 16:1114-1116.

Soltis, P.S. and Soltis, D.E. 2003. Applying the bootstrap in phylogeny reconstruction. *Stat. Sci.* 18:256-267.

Stamatakis, A. 2006. RAXML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22:2688-2690.

Stamatakis, A., Ludwig, T., and Meier, H. 2005. RAXML-III: A fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics* 21:456-463.

Stamatakis, A., Blagojevic, F., Nikolopoulos, D.S., and Antonopoulos, C.D. 2007. Exploring new search algorithms and hardware for phylogenetics: RAXML meets the IBM cell. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* 48:271-286.

Stamatakis, A., Hoover, P., and Rougemont, J. 2008. A rapid bootstrap algorithm for the RAXML Web servers. *Syst. Biol.* 57:758-771.

Sterner, K.N., Raaum, R.L., Zhang, Y.P., Stewart, C.B., and Disotell, T.R. 2006. Mitochondrial data support an odd-nosed colobine clade. *Mol. Phylogenet. Evol.* 40:1-7.

Stewart, C.B., Schilling, J.W., and Wilson, A.C. 1987. Adaptive evolution in the stomach lysozymes of foregut fermenters. *Nature* 330:401-404.

Swofford, D.L. 2002. PAUP*: Phylogenetic Analysis Using Parsimony (*and Other Methods). Sinauer, Sunderland, Massachusetts.

Swofford, D.L., Olsen, G.J., Waddell, P.J., and Hillis, D.M. 1996. Phylogenetic inference. *In* Molecular Systematics (D.M. Hillis, C. Moritz, and B.K. Mable, eds.) pp. 407-514. Sinauer, Sunderland, Massachusetts.

Talavera, G. and Castresana, J. 2007. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Syst. Biol.* 56:564-577.

Wall, D.P., Fraser, H.B., and Hirsh, A.E. 2003. Detecting putative orthologs. *Bioinformatics* 19:1710-1711.

Whelan, S. and Goldman, N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol.* 18:691-699.

Whelan, S., Lio, P., and Goldman, N. 2001. Molecular phylogenetics: State-of-the-art methods for looking into the past. *Trends Genet.* 17:262-272.

Yang, Z. 1996. Among-site rate variation and its impact on phylogenetic analyses. *Trends Ecol. Evol.* 11:367-372.

Zwickl, D.J. 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. Doctoral thesis. The University of Texas at Austin.