# Phylogeny for the faint of heart: a tutorial

**Sandra L. Baldauf**

Department of Biology, University of York, Box 373, York, UK YO10 5YW

**Phylogenetic trees seem to be finding ever broader applications, and researchers from very different backgrounds are becoming interested in what they might have to say. This tutorial aims to introduce the basics of building and interpreting phylogenetic trees. It is intended for those wanting to understand better what they are looking at when they look at someone else's trees or to begin learning how to build their own. Topics covered include: how to read a tree, assembling a dataset, multiple sequence alignment (how it works and when it does not), phylogenetic methods, bootstrap analysis and long-branch artefacts, and software and resources.**

Phylogenetics is the science of estimating the evolutionary past, in the case of molecular phylogeny, based on the comparison of DNA or protein sequences. The idea of representing these hypotheses as trees probably dates back to Darwin, but the numerical calculation of trees using quantitative methods is relatively recent [1], and their application to molecular data even more so [2]. In the age of rapid and rampant gene sequencing, molecular phylogeny has truly come into its own, emerging as a major tool for making sense of a sometimes overwhelming amount information.

This tutorial aims to introduce the basic principles behind and programs for constructing evolutionary trees (phylogenetic analysis). It is intended primarily for those who want to read other people's trees, but also as a general introduction for those who might wish to begin to try building their own. In the latter case the reader is warned – phylogenetic analysis and evolutionary theory are not trivial pursuits; as with any new methodology, it is advisable to seek expert help before getting in too deep.

## Some basics

### *Terminology*

A phylogenetic tree is composed of branches (edges) and nodes. Branches connect nodes; a node is the point at which two (or more) branches diverge. Branches and nodes can be internal or external (terminal). An internal node corresponds to the hypothetical last common ancestor (LCA) of everything arising from it. Terminal nodes correspond to the sequences from which the tree was derived (also referred to as operational taxonomic units or 'OTUs'). Trees can be made up of multigene families (gene trees) or a single gene from many taxa (species trees, at least theoretically) or a combination of the two. In the first case, the internal nodes correspond to gene duplication events, in the second to speciation events.

### *Groups*

Trees are about groupings (Fig. 1). A node and everything arising from it is a 'clade' or a 'monophyletic group'. A monophyletic group is a natural group; all members are derived from a unique common ancestor (with respect to the rest of the tree) and have inherited a set of unique common traits (characters) from it. A group excluding some of its descendents is a paraphyletic group (e.g. animals excluding humans). A hodge-podge of distantly related OTUs, perhaps superficially resembling one another or retaining similar primitive characteristics, is polyphyletic; that is, not a group at all.

### *Trees*

Intuitively we draw trees from the ground up like real trees (Fig. 2a). However, as these trees get larger and more complex, they can become cluttered and difficult to read. As an alternative we can expand the nodes (Fig. 2b) and turn the tree on its side (Fig. 2c). Now the tree grows left to right, and all the labels are horizontal. This makes the tree

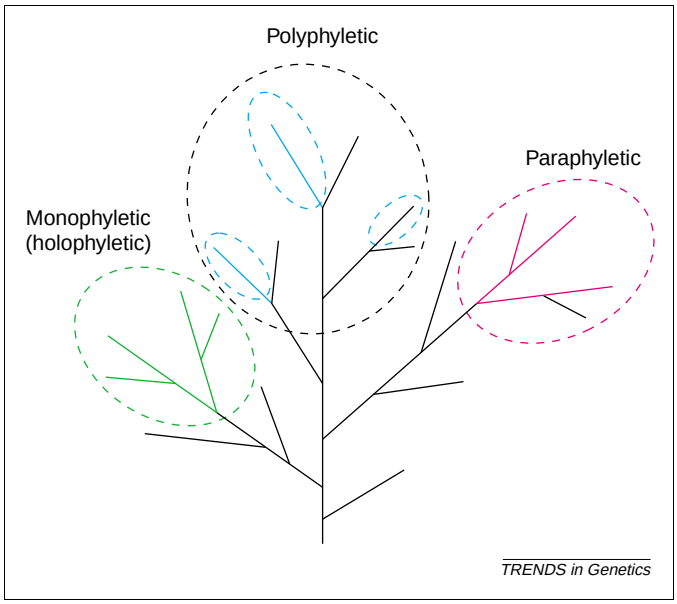


**Fig. 1.** Trees are about groups: monophyletic (holophyletic), paraphyletic and 'polyphyletic'.

*Corresponding author:* Sandra L. Baldauf (slb14@york.ac.uk).

http://tigs.trends.com 0168-9525/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved. doi:10.1016/S0168-9525(03)00112-4

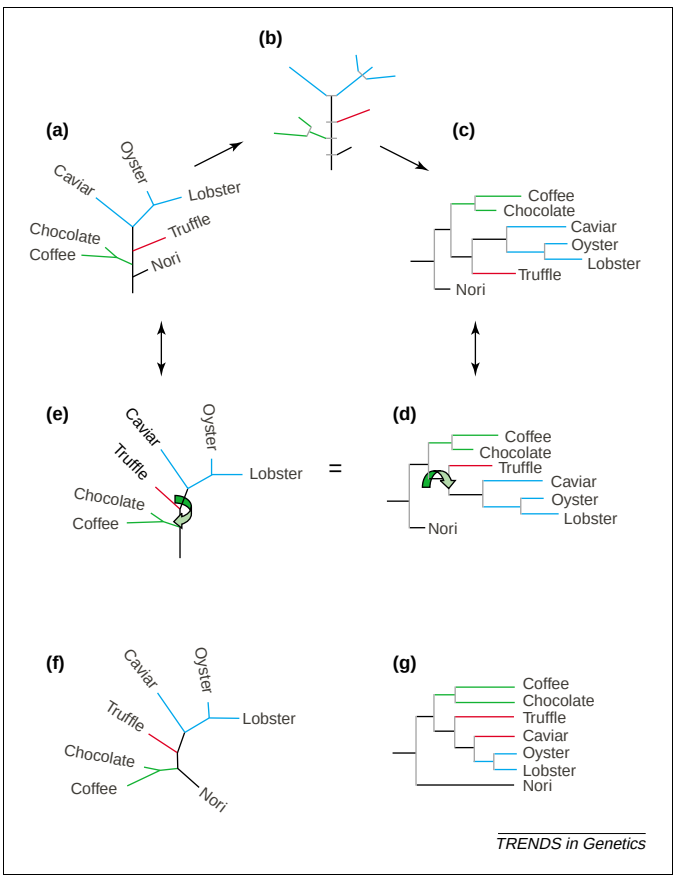


**Fig. 2.** Phylogenetic tree styles. All these trees have identical branching patterns. The only differences are (f), which is unrooted. (g) is a cladogram, so the branch lengths are right justified and not drawn to scale (i.e. they are not proportional to estimated evolutionary difference).

easier to read and to annotate. Thus, the widths of the nodes have no meaning; they are simply adjusted to give even spacing to the branches. To make things slightly more complicated, all branches can rotate freely about the plane of their nodes, so all trees in Fig. 2 are identical (except that tree F is 'unrooted', see below).

Molecular phylogenetic trees are usually drawn with proportional branch lengths; that is, the lengths of the branches correspond to the amount of evolution (roughly, percent sequence difference) between the two nodes they connect (Fig. 2a–f). Thus, the longer the branches the more relatively divergent (highly evolved) are the sequences attached to them. Alternatively, trees can be drawn to display branching patterns only ('cladograms'), in which case the lengths of the branches have no meaning (Fig. 2g), but this is rare done with molecular sequence trees.

### *Roots*

At the base of a phylogenetic tree is its 'root'. This is the oldest point in the tree, and it, in turn, implies the order of branching in the rest of the tree; that is, who shares a more recent common ancestor with whom. The only way to root a tree is with an 'outgroup', an external point of reference. An outgroup is anything that is not a natural member of the group of interest (i.e. the 'ingroup'). This might not seem like a difficult concept, but do not be misled. The excluded member of a monophyletic group (i.e. the exclusion that makes it paraphyletic, Fig. 1) is not an outgroup (just an outcast); for example, humans are not an outgroup to animals. In the absence of an outgroup, the best guess is to place the root in the middle of the tree (at its midpoint), or, better yet, not root it at all (Fig. 2f). Alternatively you can use extrinsic, more traditional taxonomic information, such as the fossil record in the case of species trees. This is obviously more difficult with gene trees.

### *Homology*

Evolution is about homology; that is, the similarity due to common ancestry. Homologues can be orthologues or paralogues (Fig. 3). Orthologues only duplicate when their host divides; *i.e.* along with the rest of the genome (Fig. 3a). They are strictly vertically transmitted (parent to offspring), so their phylogeny traces that of their host lineage (Fig. 3b). Paralogues are members of multigene families; they arise by gene duplication (Fig. 3a). If you try to infer species relationships with paralogues you can run into trouble; if some of the copies are missing, you can be very convincingly misled (Fig. 3c). However, if you have all copies of two paralogues in your tree, then you are fine. Better still, you have two mirror phylogenies (Fig. 3b). In this case, paralogues can serve as each other's natural

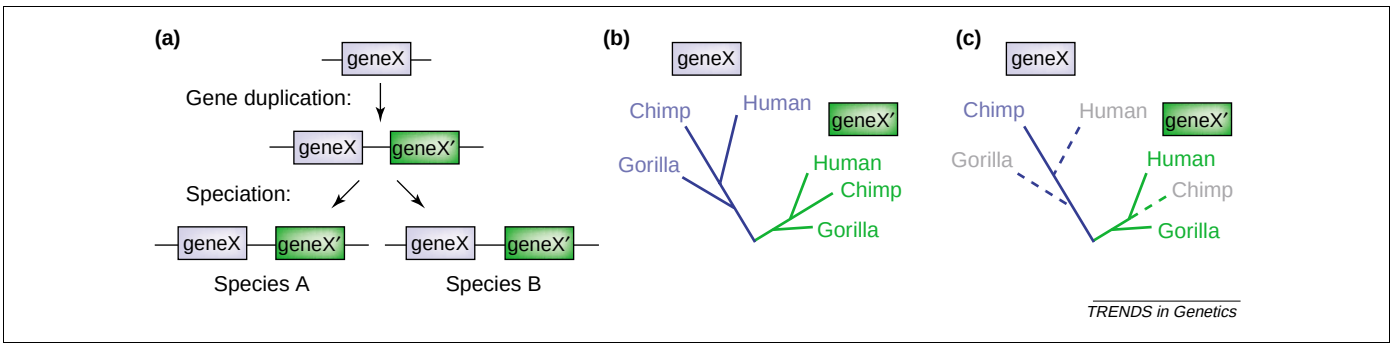


**Fig. 3.** The problem with paralogues. (a) Paralogous genes are created by gene duplication events. Gene X is duplicated in a common ancestor to species A and B resulting in two paralogous genes, X and X′. All subsequent species inherit both copies of the gene (unless one or the other is lost somewhere along the way). (b) Phylogenetic analysis of the X/X′ gene family gives two parallel phylogenies. All sequences of gene X are orthologues of each other, and all the sequences of gene X′ are orthologues of each other. However, X and X′ are paralogues. Both the X and X′ subtrees show the true relationships among the three species. The subtrees are also each other's natural outgroup, and as a result each subtree is rooted with the other (reciprocally rooting). (c) A tree of the X/X′ gene family can be misleading if not all the sequences are included (because of incomplete sampling or gene loss). If the broken branches are missing, then the true species relationships are misrepresented.

outgroup. This was the method used to infer the root of the universal tree of life [3–5].

### Step 1. Assembling a dataset

The first step in constructing a tree is building the dataset. For most of us, this means finding and retrieving sequences from the public domain. The main repository for these data is the public nucleotide database (Box 1), stored independent in the USA (GenBank), EU (EMBL) and Japan (DDBJ). Primary entries are redundant among them, and they are updated against each other nightly. Some of the most exciting molecular evolutionary data are coming from genome sequencing projects (Box 1). Much of this data, both in-progress and completed, is deposited in the public database, with some in-progress data partitioned off separately. Other genome project data are available only from their own websites; for example, The Institute for Genomic Research (TIGR, Box 1) and the Joint Genome Research Institute (DOE, Box 1). Comprehensive lists and progress reports of on-going genome sequencing projects are available from several sources (Box 1).

There are two basic kinds of search strategy for finding a set of related sequences – Keywords and similarity. A Keywords search identifies sequences by looking through their written descriptions (i.e. the annotation section of a database file); a similarity search looks at the sequences themselves (e.g. using 'BLAST' software, Box 1). Keywords searching is easier and seems more intuitive, but it is far from exhaustive. This is mostly because a lot of data entries are very scantily annotated or even mis-annotated (sometimes quite entertainingly so). This is particularly true for genomic data where high throughput is the priority. The best-annotated data are the painstakingly annotated protein data found in the SwissProt database [6]. This is accessible directly or through the main database sites (Box 1), but this is only a subset of all that is available.

The main search engines for Keywords searching are Entrez (NCBI) and SRS (everywhere else); both have excellent online tutorials (Box 1). Beginners might find SRS easier, with its simple forms and obvious blanks to fill in. The main search engine for similarity searching is the 'BLAST' software [7], available at all databanks and most genome websites (Box 1). The NCBI BLAST server is the most sophisticated with numerous 'flavours' and options such as honing a BLAST search using keywords, searching with alignment profiles to find distant homologues (PSI-BLAST), and much more.

A word on database 'etiquette'. A large body of unpublished genomic data is now freely available over the Internet. It is generally (although not universally) felt that these data should be treated as privileged communications, with any significant or large-scale analyses cleared with the submitters before publication and, obviously, gratefully acknowledged. This is basically a courtesy to the authors, most of whom are as publication-dependent as the rest of us [8].

### Step 2. Multiple sequence alignment – the heart of the matter

Molecular trees are based on multiple sequence alignments. Until 1989 these were all assembled by hand (e.g. [9]) because the exhaustive alignment of more than six or eight sequences was, and more or less still is, computationally unfeasible. Now, most multiple sequence alignments are constructed by the method known as 'progressive sequence alignment' [10,11]. This method builds an alignment up stepwise, starting with the most similar sequences and progressively adding the more dissimilar ('divergent') ones (Fig. 4a). The process begins with the construction of a crude 'guide tree' (Fig. 4a). This tree then determines the order in which the sequences are progressively added to build the alignment (Fig. 4b). Note that the guide tree is included as part of the alignment output, but only to show the user how the alignment was assembled.

The cardinal rule of progressive sequence alignment is 'once a gap always a gap'; gaps can only be added or enlarged, never moved or removed [10]. This is based on the assumption that the best information on gap placement will be found among the most similar

**Box 1. Bioinformatic resources**

**Databases**

*General*

**DDBJ (Japan)**: http://www.ddbj.nig.ac.jp/
**EMBL (EU)**: http://www.ebi.ac.uk/Databases/
**GenBank (USA)**: http://www.ncbi.nlm.nih.gov/

*Genomes*

**TIGR**: http://www.tigr.org/tdb/mdb/mdbcomplete.html
**JGI**: http://www.jgi.doe.gov/JGI_microbial/html/index.html
**Sanger**: http://www.sanger.ac.uk/Projects/Microbes/
**NCBI**: http://ncbi.nlm.nih.gov/Genomes/index.html

*Lists of genomes in progress*

http://wit.integratedgenomics.com/GOLD/
http://www.tigr.org/tdb/mdb/mdbinprogress.html

**Data acquisition (search engines)**

*Keyword*

**SRS**: http://srs.ebi.ac.uk/ (tutorials can be found at http://www.icgeb.trieste.it/~netsrv/courses/RH/srs/ or http://www.no.embnet.org/Programs/DB/srs_tut.php3)
**Entrez**: http://www.ncbi.nlm.nih.gov/Entrez/ (tutorial can be fond at http://www.ncbi.nlm.nih.gov:80/Database/tut1.html)

*Similarity*

**BLAST**: http://www.ncbi.nlm.nih.gov/BLAST/

**Multiple sequence alignment**

**ClustalX**: ftp://ftp.ebi.ac.uk/pub/software/ (tutorial can be found at http://www.bioinf.org/molsys/ClustalX_tutorial.html)
**BioEdit**: http://www.mbio.ncsu.edu/BioEdit/bioedit.html
**BCM search launcher**: http://searchlauncher.bcm.tmc.edu/
**GCG**: http://www.accelrys.com/products/gcg_wisconsin_package
**Lists**: http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/welcome.html/

**Phylogenetic analysis**

**PAUP***: http://paup.csit.fsu.edu/index.html (tutorial can be found at http://paup.csit.fsu.edu/Quick_start_v1.pdf)
**Mega2**: http://www.megasoftware.net/
**PHYLIP**: http://evolution.genetics.washington.edu/phylip.html
**Treeview**: http://taxonomy.zoology.gla.ac.uk/rod/treeview.html
**List**: http://evolution.genetics.washington.edu/phylip/software.html

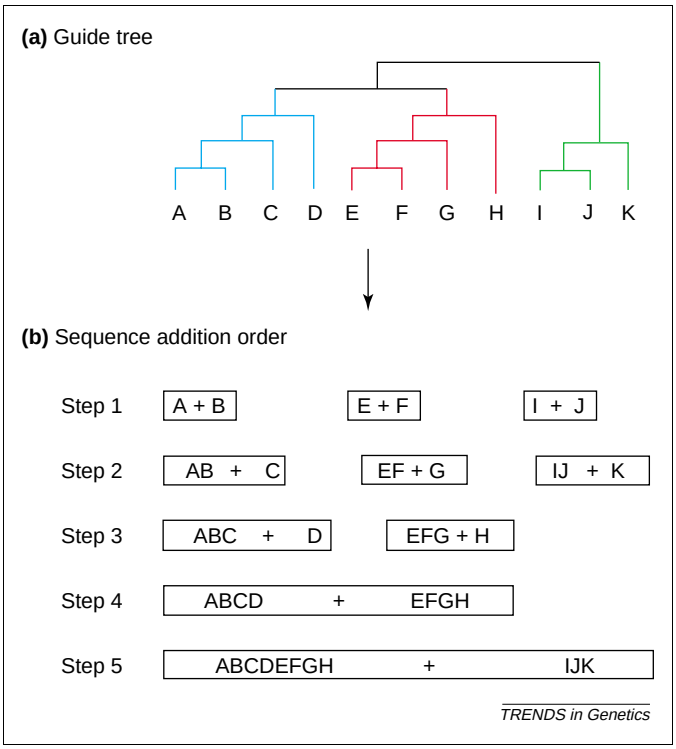


**Fig. 4**. Steps in progressive sequence alignment. (a) The first step is to calculate the guide tree. (b) This determines the order in which sequences are added to the growing alignment.

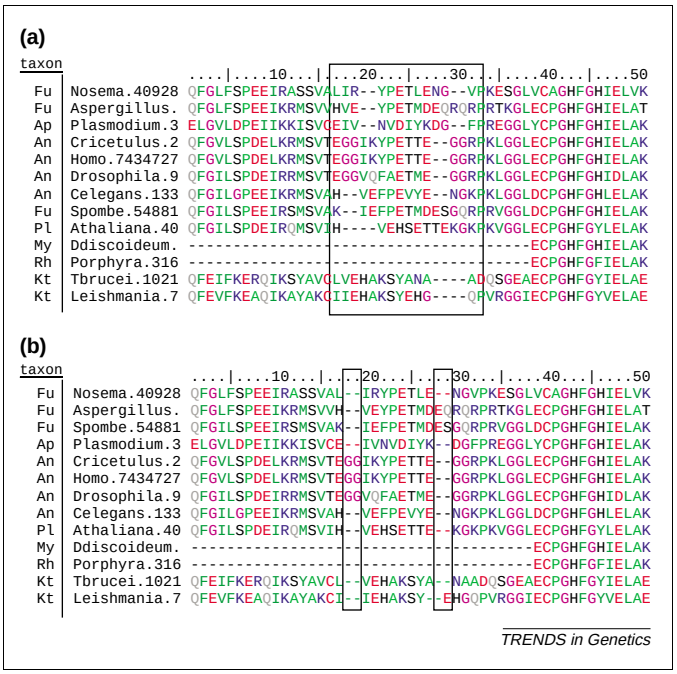


**Fig. 5**. Refining an alignment. (a) The raw output from a ClustalX alignment of rpb1 sequences, which predicts six insertion/deletion events (boxed), some of which are blatantly inconsistent with known taxonomy. (b) The refined alignment makes much better evolutionary sense, because it shows only two insertion events in well-defined taxonomic groups (animals and higher fungi). Taxon labels are Fu (fungi), An (animals), Pl (green plant), Ap (apicomplexan), Rh (rhodophyte), My (mycetozoan), Kt (kinetoplastids). In (b), the sequence from *Saccharomyces pombe* has been placed adjacent to the other fungi to make these relationships more obvious.

sequences, but also for practical purposes; if gaps were adjusted at every step the alignment process would be tremendously slower. However, the once-a-gap rule can also be the source of some obvious silliness, because there often is better information in the full alignment on where the gaps really belong. This can be particularly apparent for small deletions, which might clearly be shared by several sequences but nonetheless placed at slightly different positions in each (Fig. 5a). These types of error are among the reasons for the widespread and fairly well-accepted practice of 'adjusting' alignments 'by eye' to 'minimize insertion/deletion events' (Fig. 5b), using a program such as BioEdit (see below).

Alignments are about gaps – where to put them and how big to make them. These are two different issues. Genes do not generally take insertions and deletions lightly. One out of three changes the reading frame, not to mention adding new stop codons or unwieldy junk to a protein's structure. However, the size of a gap is much less important than the fact that it is there at all, so alignment programs have separate penalties for inserting a gap (which is costly) and for making it bigger (relatively cheap). Ideally gap penalties should differ for closely related versus distantly related sequences, for different kinds of sequence, and for different regions of the same sequence, but this is mostly impractical. Therefore, all gap penalties are compromises, and an alignment can look very different depending on the penalties that are used. In the end, the user might need to try a range of penalties, compare these by eye and pick the most logical combination [12].

As exhaustive multiple sequence alignment is essentially impossible, new improved methods are something of a cottage industry (Box 1). However, the oldest program is also one of the easiest, friendliest and most widely used, only partly because its also free. This is Clustal [11], now in its W and X incarnations (X being the X-window version of W; Box 1). Besides basic alignment, the program allows iterative alignment of selected regions, profile alignment (i.e. alignment of alignments), and basic phylogenetic analysis (see below [12]). Although it does not allow you to modify your alignment within the program, Clustal can be run as a subroutine of the BioEdit sequence editor (Box 1) or a Clustal alignment imported into BioEdit for subsequent editing (Box 1). Of course the GCG package will do all this as well – for a substantial fee (Box 1).

## Step 3. Trees – methods, models and madness

### *The infile*

The basic premise of a multiple sequence alignment is that, for each column in the alignment, every residue from every sequence is homologous; that is, has evolved from the same position in a common ancestral sequence without insertion or deletion. When this premise is met, a multiple sequence alignment can hold a wealth of information about protein structure and function, mode of evolution and, of course, phylogeny. However, a molecular phylogeny is only as good as the alignment it's based on. At best, misaligned sequence has no useful phylogenetic information; at worst, it might have convincing misinformation.

Therefore, the first step in tree building is to inspect your alignment carefully and to decide what should and should not be included in your analysis. The general rule is

to delete all positions with gaps plus any adjacent, ambiguously aligned positions (i.e. columns in the alignment). This is for two reasons. First, you cannot be confident that these regions are correctly aligned, and if they are misaligned then they will have no real phylogenetic information. Second, even if it is clear that a region containing a gap is correctly aligned, it can have an undue influence on your tree. This is because the larger the gap, the more characters uniting the groups that share it; for example, a 9-nucleotide insertion would be nine shared characters for the OTUs that have it. This is inappropriate because, in reality, a gap is a single evolutionary event, regardless of its size. The importance of a gap is not proportional to its size for the same reason that gap insertion penalties are so much larger than gap extension penalties.

Another important consideration in building molecular trees from protein-coding genes is whether to analyse your sequences at the DNA or the protein level. For closely related sequences, there will be more change (information) at the DNA level, so you'll want to use DNA sequences. For more distant relationships, amino acid sequences hold more information. However, you can still use DNA sequences for distant relationships, but first it is important to remove third codon positions as these will be pure noise or worse.

### *Methods*

The methods for calculating phylogenetic trees fall into two general categories [13]. These are distance-matrix methods, also known as clustering or algorithmic methods (e.g. UPGMA, neighbour-joining, Fitch–Margoliash), and discrete data methods, also known as tree searching methods (e.g. parsimony, maximum likelihood, Bayesian methods) [13–16]. Distance is relatively simple and straightforward – a single statistic, the distance (roughly, the percent sequence difference), is calculated for all pairwise combinations of OTUs, and then the distances are assembled into a tree. Discrete data methods examine each column of the alignment separately and look for the tree that best accommodates all of this information. Unsurprisingly, distance methods are much faster than discrete data methods. However, a distance analysis yields little information other than the tree. Discrete data analyses, however, are information rich; there is an hypothesis for every column in the alignment, so you can trace the evolution at specific sites in the molecule (e.g. catalytic sites or regulatory regions).

One way to look at the two classes of methods is to imagine trying to come up with an evolutionary classification of the flowers in your garden. You would start by counting the number of petals, sepals and stamens, etc. for each – that's your dataset. If you used a distance approach, you could sort your flowers simply by the number of characters they share; the flowers with the most characters in common would be presumed to be the most closely related. To use a tree searching method, you would first calculate a set of all possible classification Scheme (i.e. all possible trees), and then measure how each of your characters would have to evolve on each of these trees (e.g. would asymmetrical flowers have to evolve twice on tree 1 versus tree 2, etc). The best possible classification scheme (tree) would be the one that required the simplest set of hypotheses.

In the end, in most cases, you would come up with the same groupings. However, if you had any very unusual or highly degenerate flowers, ones that bore little resemblance to any of the rest, then the problem gets harder. Under these conditions, one or other of your methods could fail, although for somewhat different reasons. This is why people generally test their trees with more than one phylogenetic method (see bootstrapping below).

### *Models*

This is all complicated by the fact that molecular evolution is ancient history, a kind of molecular archaeology where we are trying to recover the past by extrapolating backward from a small set of surviving clues. If little evolution has occurred, this is fairly straightforward. However, and quite rapidly, the true evolutionary difference between two sequences becomes obscured by multiple mutations (changes on top of changes), especially at the more rapidly evolving sites. In these cases, a simple count of the differences between two sequences will underestimate how much evolution has actually occurred. Various models (corrections) have been developed to try to estimate the true difference between sequences based on their present states, such as amino acid substitution matrices (e.g. Dayoff, Blossom, etc.) or gamma corrections (giving more weight to changes at slowly evolving sites), etc. However, it is beyond the scope of this tutorial to explain these, and the interested reader should consult one of several excellent texts on molecular evolution for further detail [13–17].

### *Programs*

PHYLIP, Mega and PAUP* (pronounced 'pop star') are the most comprehensive and widely used phylogeny packages (Box 1). All are inexpensive or free, and all allow a variety of models and methods. PHYLIP is the granddaddy of them all, but Mega2 is perhaps the easiest to use (at least on a PC), as it has straightforward pulldown menus. Both the PHYLIP and Mega2 manuals are also good all-round primers on phylogenetic theory and practice. PAUP*, a perpetual work in progress, is easily the most sophisticated and versatile of the lot, but also has the steepest learning curve (Box 1).

### *An example*

Clustal (see above) can also be used to calculate trees using evolutionary distances. Although even the authors do not recommend this for serious phylogeny, quick and easy has its uses. It is also easy to explain and a good way for the beginner to get their feet wet. To calculate a phylogenetic tree using ClustalX involves three or four very simple steps. All the necessary commands are located in the 'Trees' menu. First, remove the regions of the alignment with gaps (select 'delete positions with gaps'). Second, if your sequences are less than ~95% identical, you should choose to correct your distance measures for multiple substitutions ('correct for multiple substitutions'). Third, make sure you have the right format for the output (follow the 'output format' link', change 'bootstrap label options'

from 'branches' to 'nodes' and close). Finally, calculate the tree by selecting 'bootstrap NJ tree'.

This last analysis actually incorporates several steps. First, the program calculates a distance (neighbour-joining) tree, then it evaluates the tree by a statistical test called bootstrapping (see below). Finally, it puts all the information together into a single output file, consisting of the tree with the appropriate bootstrap values on it. To view these results you need a tree-viewing program, such as Treeview (Box 1), and you're done. This is about as basic as it gets, but it is often a good place to start; for example, to sort through mountains of sequences quickly to pick a representative set to work with or to decide whether your data look interesting enough to pursue in more detail.

## Step 4. Tests – telling the forest from the trees

### *Bootstrapping*

So how good was that tree? The simplest test of phylogenetic accuracy is the bootstrap [18]; it is rare now to see a tree without it. Bootstrapping essentially tests whether your whole dataset is supporting your tree, or if the tree is just a marginal winner among many nearly equal alternatives. This is done by taking random subsamples of the dataset, building trees from each of these and calculating the frequency with which the various parts of your tree are reproduced in each of these random subsamples. If group X is found in every subsample tree, then its bootstrap support is 100%, if its found in only two-thirds of the subsample trees, its bootstrap support is 67% (Fig. 6). Each of the subsamples is the same size as the original, which is accomplished by allowing repeat sampling of sites; that is, random sampling with replacement (Fig. 6a). It is a simple test, but bootstrap analyses of known phylogenies (viral populations evolved in the laboratory) show that it is a generally dependable measure of phylogenetic accuracy, and that values of 70% or higher are likely to indicate reliable groupings [19].

### *Long branches*

The most problematic and pervasive problem in molecular phylogeny is the problem of 'long branch attraction' [20,21]. This is the tendency of highly divergent sequences (i.e. those with long terminal branches) to group together in a tree regardless of their true relationships. This is at least partly because rapidly evolving sequences, or sequences without any close relatives, will have numerous unique mutations (with respect to the rest of the tree). Because there are only a limited number of possible states (20 amino acids or 4 nucleotides) for rapidly evolving sites to change to, sequences with a lot of these changes will start to pick up spurious similarities to each other. If their branches are very long (i.e. if there are a lot of these changes), these spurious similarities can override the true phylogenetic signal, and the sequences will be 'attracted' to each other.

This causes all sorts of problems, one of which is that bootstrap values all over the tree tend to deteriorate. In fact, one way to test whether you have a problem with long branches is to remove these sequences from your dataset and see if the bootstrap values go up. However, there is no easy solution to the problem of long-branch attraction.

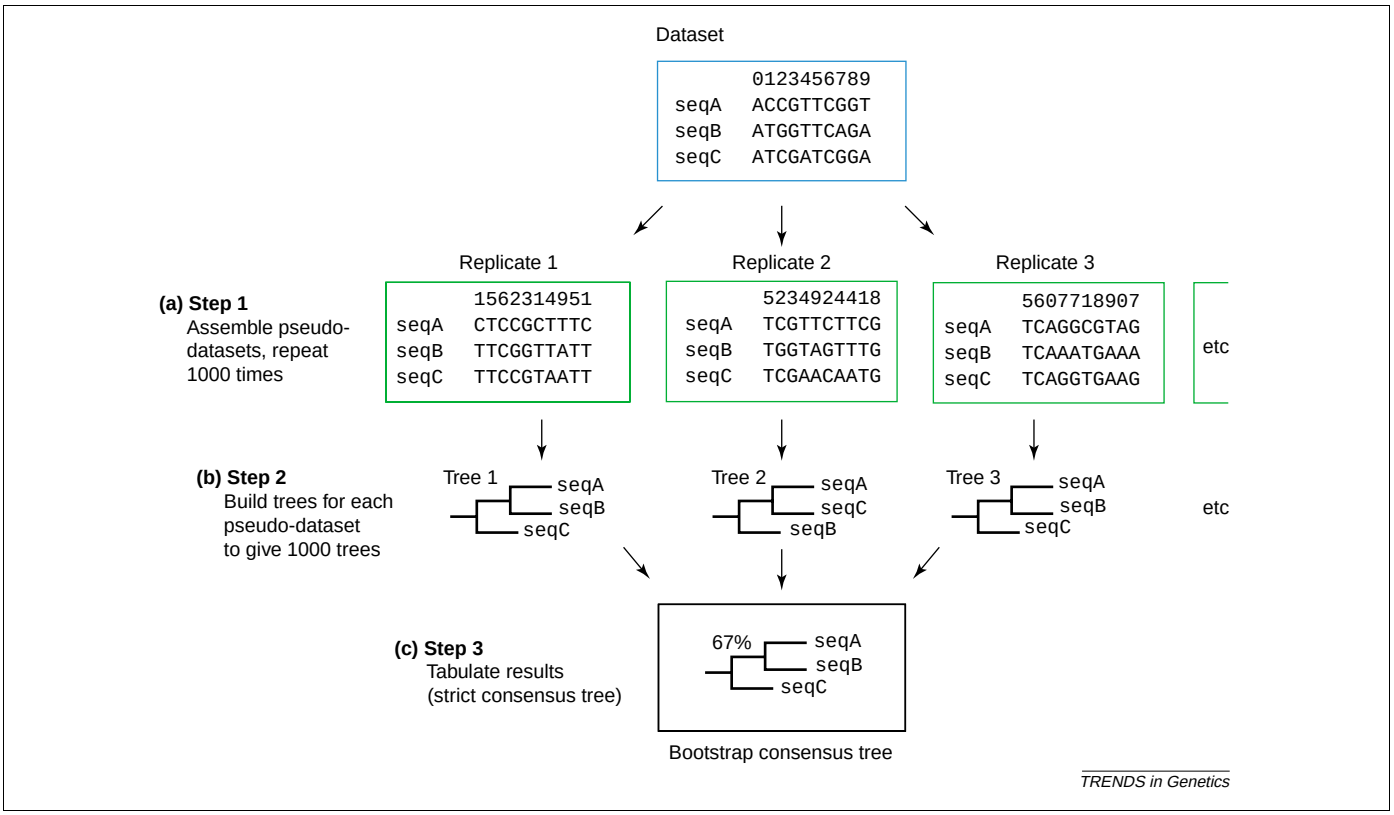


**Fig. 6.** Bootstrap analysis proceeds in three steps. The dataset is randomly sampled with replacement to create multiple pseudo-datasets of the same size as the original ((a), three are shown in this example). (b) Individual trees are constructed from each of the pseudo-datasets. (c) Each of the pseudo-dataset trees are scored for which nodes (groupings) appear and how often. In this case, a node uniting seqA plus seqB is found in two of the three replicate trees. This gives a bootstrap support for this grouping of 2/3 or 67%.

Methods such as maximum likelihood tend to be less affected than others, but rarely strikingly so [22]. Generally, it is best to break up the long branches, by adding intermediate sequences to a tree [23]. To return to the flower analogy, with highly derived forms, if you can find some of the intermediate relatives it becomes much easier to see how they fit in with the rest of your taxa. If all else fails, and you do not really need the offending sequence, you can either omit it from your dataset, being completely transparent about why, or analyse your data with and without it and see what difference it really makes.

### Step 5. Data presentation

Finally, there are few set rules on how to present phylogenetic trees, but there are some widely accepted conventions. In molecular phylogenetic trees, branch lengths are almost always drawn to scale; that is, proportional to the amount of evolution estimated to have occurred along them (Fig. 2a–f). Although the relationship between branch lengths and real time is far from straightforward and probably unreliable for any single gene, lengths still give a good general impression of relative rates of change across a tree. Bootstrap values should be displayed as percentages, not raw values. This makes the tree easier to read and to compare with other trees. By convention, only bootstrap values of 50% or higher are reported; lower values mean that the node in question was found in less than half of the bootstrap replicates. And finally, please, please use meaningful names for OTUs or annotate your tree by indicating important groupings with brackets, color etc.; a tree full of three letter acronyms or bare database accession numbers can be excruciating to interpret.

Phylogenetic analysis is a powerful tool for sorting and interpreting molecular data. With even a very basic understanding of general principles and conventions it is possible to glean valuable information from a phylogenetic tree – on the origin, evolution and possible function of genes and the proteins they might encode. I hope that you will now feel more confident in reading trees and more intrigued than ever to discover what they might have to say.

### References


1 Sneath, P.H.A. and Sokal, R.R. (1973) *Numerical Taxonomy*, W.H. Freeman
2 Zuckerkandl, E. and Pauling, L. (1965) Evolutionary divergence and convergence in proteins. In *Evolving Genes and Proteins* (Bryson, V. and Vogel, H.J., eds) pp. 97–166, Academic Press
3 Iwabe, N. *et al.* (1989) Evolutionary relationship of archaebacteria, eubacteria, and eukaryotes inferred from phylogenetic trees of duplicated genes. *Proc. Natl. Acad. Sci. U. S. A.* 86, 9355–9359
4 Gogarten, J.P. *et al.* (1989) Evolution of the vacuolar $H^+$-ATPase: implications for the origin of eukaryotes. *Proc. Natl. Acad. Sci. U. S. A.* 86, 6661–6665
5 Baldauf, S.L. *et al.* (1996) The root of the universal tree and the origin of eukaryotes based on elongation factor phylogeny. *Proc. Natl. Acad. Sci. U. S. A.* 93, 7749–7754
6 Boeckmann, B. *et al.* (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.* 31, 365–370
7 Altschul, S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402
8 Roberts, L. (2003) A tussle over the rules for DNA data sharing. *Science* 298, 1312–1313
9 Schwartz, R.M. and Dayhoff, M.O. (1978) Matrices for detecting distant relationships. In *Atlas of Protein Sequence and Structure* (Dayhoff, M.O., ed.), pp. 353–358, National Biomedical Research Foundation
10 Feng, D.F. and Doolittle, R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* 25, 351–360
11 Higgins, D.G. and Sharp, P.M. (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* 73, 237–244
12 Hall, B.G. (2000) *Phylogenetic Trees Made Easy: a How-To Manual for Molecular Biologists*, Sinauer Associates
13 Page, R.D.M. and Holmes, E.C. (1998) *Molecular Evolution: a Phylogenetic Approach*, Blackwell Science
14 Graur, D. and Li, W-H. (1999) *Fundamentals of Molecular Evolution*, Sinauer Associates
15 Durbin, R. *et al.* (2000) *Biological Sequence Analysis*, Cambridge University Press
16 Nei, M. and Kumar, S. (2000) *Molecular Evolution and Phylogenetics*, Cambridge University Press
17 Swofford, D.L. *et al.* (1996) Phylogenetic inference. In *Molecular Systematics* (Hillis, D.M. *et al.*, eds), Sinauer Associates
18 Felsenstein, J. (1985) Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* 39, 783–791
19 Hillis, D.M. and Bull, J.J. (1993) An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analyses. *Syst. Biol.* 42, 182–192
20 Felsenstein, J. (1978) Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Zool.* 27, 401–410
21 Gribaldo, S. and Philippe, H. (2002) Ancient phylogenetic relationships. *Theor. Popul. Biol.* 61, 391–408
22 Hillis, D.M. *et al.* (1994) Application and accuracy of molecular phylogenies. *Science* 264, 671–677
23 Zwickl, D.J. and Hillis, D.M. (2002) Increased taxon sampling greatly reduces phylogenetic error. *Syst Biol.* 51, 588–598



**Do you want to reproduce material from a *Trends* journal?**

This publication and the individual contributions within it are protected by the copyright of Elsevier Science. Except as outlined in the terms and conditions (see p. ii), no part of any *Trends* journal can be reproduced, either in print or electronic form, without written permission from Elsevier Science. Please address any permission requests to:

Rights and Permissions,
Elsevier Science Ltd,
PO Box 800, Oxford, UK OX5 1DX.