INFERRING LARGE PHYLOGENIES: THE BIG TREE PROBLEM

by

Rutger Aldo Vos
Doctorandus, Universiteit van Amsterdam, 2000

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in the Department
of
Biological Sciences

SIMON FRASER UNIVERSITY
Summer, 2006

APPROVAL

Name: Rutger Aldo Vos

Degree: Doctor of Philosophy

Title of Thesis:

Inferring large phylogenies: The big tree problem

Examining Committee:

Chair: Dr. H. Hutter, Associate Professor

Dr. A. Mooers, Assistant Professor, Senior Supervisor
Department of Biological Sciences, S.F.U.

Dr. W. Maddison, Professor
Departments of Zoology and Botany, University of British Columbia

---

Dr. B. Crespi, Professor
Department of Biological Sciences, S.F.U.

Dr. S. Graham, Assistant Professor
Botanical Garden & Centre for Plant Research, and Department of Botany, University of British Columbia
Public Examiner

Dr. R. Page, Professor
Division of Environmental and Evolutionary Biology,
University of Glasgow
External Examiner

---

12 May 2006
Date Approved

ABSTRACT

Phylogenetic trees are graph-like structures whose topology describes the inferred pattern of relationships among a set of biological entities, such as species or DNA sequences. Inference of these phylogenies typically involves evaluating large numbers of possible solutions and choosing the optimal topology, or set of topologies, from among all evaluated solutions. Such analyses are computationally intensive, especially when the pattern of relationships among a large number of entities is being sought.

This thesis introduces two novel algorithms for the inference of large trees; one is applicable to the likelihood framework, the other to the Bayesian framework. Both approaches rely on the notion of a multi-modal tree 'landscape' through which inferential algorithms traverse. Using sampling techniques, the landscape can be perturbed sequentially, such that local optima can be evaded. The algorithms find good solutions in reasonable time, as demonstrated using real and simulated data sets.

An example of large phylogeny inference is presented in the form of a novel estimate of Primate phylogeny – the largest estimate for this Order to date. The phylogeny is based on previously published smaller phylogenies, and hence serves as a summary of the present state of Primate phylogeny. In addition to this 'supertree's' topology, composite estimates of divergence are provided also. These estimates are based on multiple, clock-like genes combined using a novel approach presented here.

Handling sets of trees and sequences poses practical problems in terms of conversion of data and the interoperation between computer programs. The thesis therefore concludes with a chapter discussing suitable data structures and programming

patterns for phylogenetics. The appendix discusses an implementation of some of these concepts in an object-oriented application programming interface.

Keywords: Phylogenetics; maximum likelihood; Bayesian systematics; MRP supertrees; divergence date estimation; tree data structures

Voor mijn grootouders en ouders.

I dedicate this dissertation to my parents and grandparents.

Tweedledum looked round him with a satisfied smile.
"I don't suppose," he said, "there'll be a tree left standing,
for ever so far round,
by the time we've finished!"

—Lewis Carroll, *Through the Looking-Glass*

ACKNOWLEDGEMENTS

I am very happy to introduce this thesis by expressing my gratitude to the many people who have helped me on the way. I am thankful to my family and friends, here and in the Netherlands, for their encouragement and support.

In particular, I would like to express my gratitude to Arne Mooers, my senior supervisor. Arne was the one to suggest I come to Vancouver, and he has been extremely supportive since. Thank you for guiding me through this stage of my academic career; you are a great supervisor, scientist and friend. I thank Bernie Crespi for his supervision, his generous offer to work in his lab and the great feedback he dispenses in the lab or over cinnamon chicken at his house. I am grateful to Wayne Maddison, for moving to Vancouver so that I could pick his brains. It is with great pleasure that I now move on to the next stage in my academic career where I get to work in the Maddison lab. I thank Rod Page, the external examiner, Sean Graham, the public examiner, and Harald Hutter, the chair at my defence, for their kind contributions of time and effort in evaluating my research.

I thank Marlene Nguyen, Barbara Sherman and Penny Simpson, who have been very helpful with the logistics of writing a thesis.

I am very grateful to all the wonderful people in the lab where I did much of the work in writing this thesis: Jeff, Christine, Patrik, Mick, Steve, Sampson, Martin and many others whose companionship and cleverness I have had the privilege to enjoy.

That lab, the Crespi lab, is part of a group of labs interested in evolution: the FAB* lab. Being part of this group has been a great academic experience; the discussions, during or outside of the lab meetings, have taught me many things and given

me many new ideas. The faculty members, Felix Breden, Arne Mooers, Bernie Crespi and Mike Hart are all very supportive, approachable and with an inspiring passion for science. I thank them all for creating this environment.

My research has been funded by Simon Fraser University, through Graduate Fellowships and the President's Research Stipend. In addition, I was funded by NSERC, the Society of Systematic Biology, and by CIPRES, the research project in which I will now continue my academic career. I am very grateful to these institutions for making my research possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I - GENERAL INTRODUCTION

Rutger A. Vos

HISTORY OF PHYLOGENETICS

Even before the notion was advanced that all life forms are related through evolutionary processes, humankind has strived to organize the diversity of life in various schemes. Initially, these schemes were based mostly on metaphysical beliefs about the meaning of biodiversity and the ordering of the universe. For many centuries, everything – including life forms, but also, among other things, rocks and angels – was thought to fit into a strict hierarchical system known as the "great chain of being". As both scientific and biological knowledge grew, observers began to infer a more tree-like organization.

Linnaeus introduced a binomial system of organization (Linnaeus, 1735), still in use today, from which this tree was readily apparent: Kingdoms split up into Classes, which split into Orders, which split into Families, which split into Genera, which are comprised of Species. However, this organization does not necessarily imply relatedness through shared inheritance of characteristics; rather, it is a system that is meant to bring order in the diversity of life forms – the way library books can be organized, or car models. Darwin made explicit that the observed tree-like pattern reflects the outcome of an underlying evolutionary process:

> *"The affinities of all the beings of the same class have sometimes been represented by a great tree. I believe this simile largely speaks the truth. The green and budding twigs may represent existing species; and those produced during former years may represent the long succession of extinct species. At each period of growth all the growing twigs have tried to branch out on all sides, and to overtop and kill the surrounding twigs and branches, in the same manner as species and groups of species have at all times overmastered other species in the*

great battle for life. The limbs divided into great branches, and these into lesser and lesser branches, were themselves once, when the tree was young, budding twigs; and this connexion of the former and present buds by ramifying branches may well represent the classification of all extinct and living species in groups subordinate to groups." (Darwin, 1872)

The only illustration in early editions of the *Origin of Species* shows a graph somewhat like an evolutionary tree, describing a thought experiment where different lineages diverge along a horizontal axis over discrete generations marked on the vertical axis.

By viewing the classification of life forms as reflecting the resulting pattern of an underlying physical process the accuracy of the proposed pattern becomes the subject not just of philosophical debates and esoteric metaphysics but also of debates related to the choice of inferential techniques and of suitable characteristics to analyze. On these latter subjects, Haeckel, an ardent promoter in Germany of Darwin's ideas, introduced many new concepts. Ernst (von) Haeckel first coined the term "phylogeny" (i.e. the pattern of evolutionary descent – and, consequently, that reconstructing this pattern constitutes "phylogenetics"), first published a graph showing a phylogenetic tree (Haeckel, 1866), introduced the level of "Phylum" between Kingdoms and Classes in the Linnaean system, and proposed that embryogenesis should be studied as a characteristic that gives insight into the evolutionary pattern:

"I established the opposite view, that this history of the embryo (ontogeny) must be completed by a second, equally valuable, and closely connected branch of thought - the history of race (phylogeny). Both of these branches of evolutionary

science, are, in my opinion, in the closest causal connection; this arises from the reciprocal action of the laws of heredity and adaptation... 'ontogenesis is a brief and rapid recapitulation of phylogenesis, determined by the physiological functions of heredity (generation) and adaptation (maintenance).'" (Haeckel, 1899)

In later years, Haeckel has been derided on several grounds, such as his propensity to fudging his scientific drawings to conform to his idea that ontogeny recapitulates phylogeny, and his advancement of the notion that every human endeavor must be viewed in the light of biology: "*politics is applied biology*" (Haeckel, 1892), a pithy quote also popular among Nazi propagandists. However, despite these objections, contemporary biologists in the field of "Evo-Devo" remain interested in embryogenesis and how it gives insight into the evolution of complex forms (Raff, 1996).

In the decades following Haeckel's work, phylogenetics consisted sometimes solely of the presentation of the expert opinion of paleontologists, embryologists and taxonomists in the form of graphs lacking objective quantitative support or explicit methodological grounding. However, advances in statistics (Fisher, 1921) could be applied to the reconstruction of evolutionary trees in a probabilistic framework, and with the advent of scientific computing, approximation of the maximum likelihood tree came within reach for more than only trivial cases. Edwards and Cavalli-Sforza (1963) suggested that the most plausible estimate of tree shape is that which minimizes the amount of required evolution.

Around that same time, the work of Willi Hennig (Hennig, 1950; Hennig, 1966) was translated into English. From a less operational perspective, he advanced several principles for phylogenetics – "phylogenetic systematics" – which can be summarized as:

1. The relationships among life forms on earth should be viewed purely in genealogical terms, as the relationships among "clades". A hypothesis of the pattern of clade relationships may be proposed.
2. Only synapomorphies give evidence for patterns of clade relationships. Synapomorphies are shared, derived characteristics.
3. Proposed "cladograms", or phylogenetic hypotheses, must conform to the evidence provided by synapomorphies. When faced with a choice, the hypothesis that explains the largest number of synapomorphies as homologues is preferred.
4. Taxonomies must follow the best supported cladograms, such that taxa at any level must be "monophyletic", i.e. all members of the taxon must form a set that is identical to the set of descendants of the most recent common ancestor of the taxon's members.

The principles of Hennig are only operational at the level of clade relationships, which are selected based on the number of synapomorphies – but not at the level of the entire tree. In contrast, the notion of minimum evolution is based on the statistical argument that, under Brownian motion, the expected position of any intermediate point in evolutionary space between two known points lies on a point on the line joining the two known points (Edwards, 1996). By extension, the tree that minimizes the total amount of evolution (i.e. the sum of the line lengths through evolutionary space) is an

approximation of the maximum likelihood tree. This makes no mention of clade membership and is a principle more readily applicable in the development of computational techniques, such as phylogeny estimation using least squares (Cavalli-Sforza and Edwards, 1964). In the following years, Cavalli-Sforza and Edwards's EVOMIN computer program for just such analyses circulated in Europe and North America.

The concepts introduced by Willi Hennig in his phylogenetic systematics and those implemented using minimum evolution are both sometimes referred to as "maximum parsimony" techniques in the sense of "Occam's razor". The argument to justify this label goes that those phylogenies must be preferred that explain the largest number of analyzed characteristics as shared derived homological characters, and that in effect, this means that in Hennig's philosophy, evolutionary changes are events that should not be invoked unnecessarily. Therefore, the phylogenetic hypothesis that invokes the smallest number of such changes – the most parsimonious tree – is preferred. However, the original argument for minimum evolution was based on quantitative statistics, not Occam's razor – and subsequently developed methods also based on increasingly sophisticated statistics.

PRESENT PARADIGMS

In contrast to "cladistic" methods that deal with the pathways of evolution and shared descent, "phenetic" methods, such as clustering methods like UPGMA (Sneath and Sokal, 1973), analyze *overall* similarity regardless of evolutionary relation. One approach consists of the measurement of the pairwise distance – along an axis such as the difference between homological gene sequences – between taxa followed by the reduction of these distances to two-dimensional graphs. "Phenetic" and "distance" are sometimes, incorrectly, used interchangeably: phylogenetic inference using maximum likelihood (Felsenstein, 1981) can also be considered a phenetic method under some circumstances, though it is not a distance method. Maximum likelihood is a paradigm that requires the explicit, *a priori*, specification of the statistical model under which (usually molecular) evolution is assumed to take place. Given a data set, e.g. a DNA sequence alignment, the combination of substitution model parameters and phylogenetic tree shape that maximizes the likelihood is selected. The maximum likelihood approach has statistical properties that allow for the calculation of confidence intervals around parameters, and for hypothesis testing (Huelsenbeck and Crandall, 1997).

Bayesian systematics (Larget and Simon, 1999; Mau et al., 1999; Yang and Rannala, 1997) uses the same likelihood functions and substitution models as phylogenetic inference under maximum likelihood, however, the way in which the optimal solution, or set of solutions, is arrived at differs. Using maximum likelihood and maximum parsimony (both "optimality criteria"), the optimal solution is usually arrived at using hill-climbing algorithms. Bayesian systematics differs in that the optimality landscape is explored using a random walk technique ("Markov chain Monte Carlo") that

has several desirable statistical and computational properties, most notably the speed with which meaningful results are obtained, including support values around parameters that would otherwise require additional bootstrapping (Felsenstein, 1985) to achieve using hill-climbing approaches.

In practical considerations of different tree searching methods the distinction between cladistic and phenetic methods is not that useful. More instructive with regard to the trade-off between computational intensity and robustness of methods are the distinctions between distance methods, which are fast but reduce the data, and character matrix methods, which are more efficient with the data, and whether the method defines an optimality criterion – perhaps based on an expensive function – against which solutions must be enumerated.

Comparisons have been performed to assess the relative performance of the various approaches mentioned here (e.g., see Huelsenbeck, 1995). In terms of retrieving the generating topology from simulated data it has been demonstrated that maximum likelihood outperforms maximum parsimony, and maximum parsimony outperforms distance methods. As Bayesian estimation of phylogeny employs the same functions and models as maximum likelihood these two approaches should usually yield the same result. Should the outcomes of these comparisons then be construed as favoring the latter methods in all instances? Certainly not (indeed, several cladists are said to have nearly resorted to physical violence to make this point during an apocryphal meeting of the Willi Hennig society). Different methods of phylogenetic inference are suitable for different applications. For example, in supertree analyses (Bininda-Emonds et al., 2002), maximum parsimony is the most commonly used method of reconciling the phylogenetic

data contained in MRP matrices (Baum, 1992; Ragan, 1992). Indeed, its assumptions about the evolutionary process (change is rare) are best suited for the underlying data in supertree analyses, where the data are not expected to follow a complex model of evolutionary change. Likewise, phenetic methods comprise a suitable approach when the distance between taxa, irrespective of the evolutionary process (in fact, agnostic with respect to its existence), is analyzed.

In summary, then, the choice for a suitable method of phylogenetic inference is presently largely contingent on the underlying assumptions of the study, and of the type of data that is analyzed. Indeed, hybrid approaches are possible, such as algorithms where initial results based on distance methods are refined using hill-climbing techniques (Vos, 2003), or Bayesian analyses whose initial stages are comprised of accelerated explorations of the optimality landscape using techniques adapted from the parsimony framework (Nixon, 1999). I present such hybrid approaches in this dissertation.

WHY STUDY PHYLOGENETICS?

The concept that all life forms are related through shared ancestry, and the image that conveys, are fundamental aspects of our understanding of the origin of the diversity we see around us. We now know that life is not naturally organized in a strict hierarchy with "lower" and "higher" forms. The history of life on earth has proceeded along a pattern of the splitting of lineages over long timescales. All extant lineages have traveled the same distance through time, diverging into endless forms (Howard and Berlocher, 1998). Over time, there have been bursts of speciation where multiple lineages arose in very short order (Schluter, 2000). In some cases, only very few lineages remain – or apparently have not speciated over long time periods. Perhaps, if anything, at least these "living fossils" should be conserved, as they represent a long line of independent evolution (Mooers et al., 2005).

Using phylogenetics, we can gain insight into the evolutionary process over long timescales, by studying the dynamics of the rate of speciation over time (Nee et al., 1995; Nee et al., 1992). For example, in this dissertation I present results showing that some clades of Primates (Cercopithecines, Old World monkeys) have speciated at a higher rate than others (Vos and Mooers, 2004). This raises questions about the ecological conditions that may have promoted these elevated rates (Barraclough et al., 1998), perhaps giving rise to comparative studies in a phylogenetic context (Harvey and Pagel, 1991; Maddison, 1990).

Studying changes in speciation rates over time can give insight into the development of viral diseases. Viral genomes can be sampled over time to analyze population growth during epidemics (Holmes et al., 1995) and to track the behavior of

emerging diseases such as H5N1 "bird flu" (Chen et al., 2004). Since the case of the gastroenterologist who was convicted of attempted second-degree murder for injecting his ex-girlfriend with blood or blood products from an HIV type 1-infected patient under his care, phylogenetic evidence has been admissible before criminal court cases in the USA (Metzker et al., 2002).

Clearly then, phylogenetics has many applications of interest to biologists and to society as a whole, and novel applications for phylogenies are continuously being added to the field (Harvey et al., 1996).

THE BIG TREE PROBLEM

The amount of data, especially molecular data, to analyze has grown dramatically over the last years. With increasing computer power and the development of better search algorithms, larger phylogenies can be inferred, which in turn can be used to greater advantage in the types of analyses discussed previously. However, phylogenetics is a computationally intensive science. Even for sets of species as small as a dozen, the set of distinct evolutionary tree shapes to connect them exceeds what can be practically stored in computer memory. Naïve tree searching strategies quickly get bogged down in endless numbers of trees to enumerate. This is referred to as the "Big Tree Problem": the computational difficulties posed by the inference of very large phylogenies under optimality criteria, which means that many possible solutions must be visited, possibly in optimality landscapes with multiple optima (Maddison, 1991). Even given the continual exponential increase in computational power, the size of the tree of life (on the order of 10^6 species) means we will continually be confronted with too-demanding phylogenetics problems. Here, I will address this issue from several perspectives – from an algorithmic (II and III), an empirical (IV and V) and a computer science (VI, appendix) perspective.

THIS DISSERTATION

This dissertation is comprised of a number of individual, self-contained papers. The chapters are closely related conceptually, and form a logical progression. In Chapters II and III I present novel algorithms for the inference of large phylogenetic trees comprised of, potentially, hundreds of species. The first algorithm (in Chapter II) is applicable to the likelihood framework; the second algorithm (Chapter III) extends some of the concepts of the first to the Bayesian framework. The algorithms are examples of inferential techniques that play to the strengths of multiple paradigms of phylogenetic inference, using distance techniques, hill-climbing and Bayesian analysis.

In the following two chapters, I discuss the inference of one such large phylogeny: that of the Order Primates. In Chapter IV I present a novel approach to combining divergence dates estimated from multiple molecular sources in order to obtain branch lengths on supertrees (Baum, 1992; Ragan, 1992) as applied to the Primate supertree. In Chapter V I discuss the topology of the Primate supertree per se, also in relation to earlier work done on this clade (Purvis, 1995). This chapter also introduces a technique whereby the divergence dates that are expected under models of constant speciation can be generated via simulation.

The next chapter (Chapter VI) deals with practical issues surrounding computer analysis of phylogenetic problems, particularly that of the internal representation of tree shapes under various programming paradigms and in various programming languages.

In the final chapter (Chapter VII) I summarize the preceding chapters and discuss these contributions to the fields, their implications and areas for future research. The

dissertation then concludes with an appendix, documenting the application programming interface of software I wrote during the preparation of this dissertation.

REFERENCES

Barraclough, T. G., A. P. Vogler, and P. H. Harvey. 1998. Revealing the factors that promote speciation. Philos. Trans. R. Soc. Lond. B Biol. Sci. 353:241-249.

Baum, B. R. 1992. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon 41:3-10.

Bininda-Emonds, O. R. P., J. L. Gittleman, and M. A. Steel. 2002. The (super)tree of life: procedures, problems and prospects. Annu. Rev. Ecol. Syst.:265-289.

Cavalli-Sforza, L. L., and A. W. F. Edwards. 1964. Analysis of human evolution. Genet. Today 3:923-933.

Chen, H., G. Deng, Z. Li, G. Tian, Y. Li, P. Jiao, L. Zhang, Z. Liu, R. G. Webster, and K. Yu. 2004. The evolution of H5N1 influenza viruses in ducks in southern China. Proc. Nat. Acad. Sci. USA 101:10452-10457.

Darwin, C. 1872. The Origin of Species by means of Natural Selection; or, the Preservation of Favoured Races in the Struggle for Life, 6th edition.

Edwards, A. W. F. 1996. The Origin and Early Development of the Method of Minimum Evolution for the Reconstruction of Phylogenetic Trees. Syst. Biol. 45:79-91.

Edwards, A. W. F., and L. L. Cavalli-Sforza. 1963. The reconstruction of evolution. Heredity 18:553.

Felsenstein, J. 1981. Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach. J. Mol. Evol. 17:368-376.

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

Fisher, R. A. 1921. On the "probable error" of a coefficient of correlation deduced from a small sample. Metron 1:3-32.

Haeckel, E. 1866. Generelle Morphologie der Organismen, 1st edition.

Haeckel, E. 1892. Monism as Connecting Religion and Science, 10th edition.

Haeckel, E. 1899. Riddle of the Universe at the Close of the Nineteenth Century. 1st edition

Harvey, P. H., A. J. L. Brown, J. M. Smith, and S. Nee (eds) 1996. New Uses for New Phylogenies. Oxford University Press, Oxford.

Harvey, P. H., and M. D. Pagel. 1991. The comparative method in evolutionary biology. Oxford University Press, Oxford.

Hennig, W. 1950. Grundzüge einer Theorie der phylogenetischen Systematik. Deutscher Zentralverlag, Berlin.

Hennig, W. 1966. Phylogenetic Systematics. University of Illinois Press, Urbana.

Holmes, E. C., S. Nee, A. Rambaut, G. P. Garnett, and P. H. Harvey. 1995. Revealing the history of infectious disease epidemics through phylogenetic trees. Philos. Trans. R. Soc. Lond. B Biol. Sci.:33-40.

Howard, D. J., and S. H. Berlocher. 1998. Endless Forms: Species and Speciation. Oxford University Press, Oxford.

Huelsenbeck, J. P. 1995. Performance of phylogenetic methods in simulation. Syst. Biol. 44:17-48.

Huelsenbeck, J. P., and K. A. Crandall. 1997. Phylogeny estimation and hypothesis testing using maximum likelihood. Annu. Rev. Ecol. Syst. 28:437-466.

Larget, B., and D. L. Simon. 1999. Markov Chain Monte Carlo Algorithms for the Bayesian Analysis of Phylogenetic Trees. Mol. Biol. Evol. 16:750-759.

Linnaeus, C. 1735. Systema naturae per regna tria naturae, secundum classes, ordines, genera, species, cum characteribus differentiis, synonymis, locis, 1st edition.

Maddison, D. R. 1991. The discovery and importance of multiple islands of most-parsimonious trees. Syst. Zool. 40:315-328.

Maddison, W. P. 1990. A method for testing the correlated evolution of two binary characters: Are gains or losses concentrated on certain branches of a phylogenetic tree? Evolution 44:539-557.

Mau, B., M. A. Newton, and B. Larget. 1999. Bayesian Phylogenetic Inference via Markov Chain Monte Carlo Methods. Biometrics 55:1-12.

Metzker, M. L., D. P. Mindell, X.-M. Liu, R. G. Ptak, R. A. Gibbs, and D. M. Hillis. 2002. Molecular evidence of HIV-1 transmission in a criminal case. Proc. Nat. Acad. Sci. USA 99:14292-14297.

Mooers, A. Ø., S. B. Heard, and E. Chrostowski. 2005. Evolutionary heritage as a metric for conservation *in* Phylogeny and Conservation (A. Purvis, et al., eds.). Oxford University Press, Oxford.

Nee, S., E. C. Holmes, A. Rambaut, and P. H. Harvey. 1995. Inferring population history from molecular phylogenies. Philos. Trans. R. Soc. Lond. B Biol. Sci.:25-31.

Nee, S., A. Ø. Mooers, and P. H. Harvey. 1992. Tempo and mode of evolution revealed from molecular phylogenies. Proc. Nat. Acad. Sci. USA 89:8322-8326.

Nixon, K. 1999. The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis. Cladistics 15:407-414.

Purvis, A. 1995. A composite estimate of primate phylogeny. Philos. Trans. R. Soc. Lond. B Biol. Sci. 348:405-421.

Raff, R. A. 1996. The Shape of Life: Genes, Development, and the Evolution of Animal Form. University Of Chicago Press, Chicago.

Ragan, M. A. 1992. Phylogenetic inference based on matrix representation of trees. Mol. Phylogenet. Evol. 1:53-58.

Schluter, D. 2000. The Ecology of Adaptive Radiation. Oxford University Press, Oxford.

Sneath, P. H. A., and R. R. Sokal. 1973. Numerical taxonomy: The principles and practice of numerical classification. W.H. Freeman, San Francisco.

Vos, R. A. 2003. Accelerated Likelihood Surface Exploration: The Likelihood Ratchet. Syst. Biol. 52:368-373.

Vos, R. A., and A. Ø. Mooers. 2004. Reconstructing divergence times for supertrees: a molecular approach *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht.

Yang, Z., and B. Rannala. 1997. Bayesian Phylogenetic Inference Using DNA Sequences: A Markov Chain Monte Carlo Method. Mol. Biol. Evol. 14:717-724.

CHAPTER II - ACCELERATED LIKELIHOOD SURFACE EXPLORATION: THE 'LIKELIHOOD RATCHET'[1]

Rutger A. Vos

[1] This chapter has been published as: Vos, R. A. 2003. Accelerated likelihood surface exploration: The 'Likelihood Ratchet'. Syst. Biol. 52:368-373. Reproduced with permission.

ABSTRACT

Algorithms for finding maximum likelihood trees can sometimes find multiple solutions (Steel, 1994; Rogers and Swofford, 1999; Chor et al, 2000; Salter, 2001). Some of these findings pertain to multiple optima in branch length space (e.g. see Steel, 1994) while others show multiple optima in tree space (Salter, 2001), necessitating algorithms that explore a large part of tree space. However, due to computational constraints, commonly-used stepwise addition based tree searching methods do not allow for this in reasonable time. Here, I propose an algorithm that significantly increases the speed at which the likelihood landscape can be explored.

Keywords: heuristics; maximum likelihood; parsimony ratchet; phylogenetic inference; tree landscapes.

INTRODUCTION

The application of maximum likelihood to tree inference problems (Felsenstein, 1981) has gained wide acceptance (Swofford et al. 1996). However, the computationally intensive nature of the phylogeny problem (number of possible rooted, binary, labeled trees = $(2n\text{-}3)!/((n\text{-}2)!2^{n-2})$ for n taxa, Felsenstein, 1978a) is compounded by the fact that calculating any single tree's likelihood score can take considerable time under complex models of sequence evolution. This imposes limits on the size of phylogenies that can be inferred using exhaustive or branch-and-bound search strategies and maximum likelihood, often called the 'big tree problem.' Heuristics are employed as feasible alternative search strategies. Typically, such searches are comprised of a mixture of global and local optimization routines: a search commences by constructing starting trees using stepwise addition, and subsequently employs a rearrangement (branch swapping) algorithm to locally improve on the starting tree's topology. Rearrangements are accepted when the fit is improved. Although some novel search algorithms under maximum likelihood allow for non-significant decreases in tree score (Salter and Pearl, 2001), the usual modus operandi is that only increases in tree score are allowed: for hill-climbing strategies, the only way is up. If none of the possible rearrangements from a given tree improves upon the result the search terminates.

Hill-climbing strategies work under the assumption that tree scores are distributed in clusters over tree space when tree space is represented as a network with closely-related tree shapes in each other's vicinity (Hendy et al., 1988). This is both the strength and weakness of rearrangement algorithms: for hill-climbing strategies to be guaranteed to find the global optimum the optimality landscape must be unimodal, such that any

local optimum is also the global one. Under maximum parsimony, this condition is often not met (Maddison, 1991). The resulting local optima ("tree islands" or locally optimal trees, which form a connected set) may lead to deceiving results during heuristic searches. The likelihood landscape has not been thoroughly characterized as of yet, but theoretical (Steel, 1994) as well as empirical evidence using simulated (Rogers and Swofford, 1999; Chor et al, 2000) and real (Salter, 2001) data has demonstrated that multiple maxima also exist under ML. Therefore, heuristic searches through solution space should take the possibility of a multimodal tree landscape into account by starting hill-climbing replicates from disparate points in tree space. In this context, a commonly used strategy is stepwise addition of sequences in a random input order, which results in different starting points between search replicates. The construction of starting trees using stepwise addition for large numbers of taxa under maximum likelihood is a time-consuming process. The alternative, starting with random starting trees saves little: such random trees are expected to be so far from any optimum that an excessive number of swaps and subsequent likelihood fits result. An alternative approach to obtain approximations of the global optimum more rapidly is found in algorithmic methods such as neighbor-joining (NJ; Saitou and Nei, 1987). A drawback of such distance tree algorithms is that the same tree is obtained irrespective of the input order of sequences, and so this approach can not be used to obtain starting points from which different regions of a multimodal optimality landscape can be reached.

For parsimony searches, additional tree searching strategies to mitigate against tree island problems have been developed (Nixon, 1999; Goloboff, 1999; Ota and Li, 2000; Ota and Li, 2001; Moilanen, 2001; Quicke et al., 2001; Charleston, 2001). Some

of these strategies rely on iterative perturbations of the tree landscape in order to escape from local optima (Nixon, 1999; Quicke et al., 2001). For example, by reweighting a random sample drawn from the data set, a tree island may no longer be locally optimal and the search may continue uphill. After reaching a new optimum, the algorithm reverts to the initial weighting scheme and the search continues, hopefully out of the reach of the original local optimum. The advantages of this strategy (known as the "Parsimony Ratchet"; Nixon, 1999) are that reweighted hill-climbing cycles preserve some of the original phylogenetic signal (rather than losing it entirely as is the case when random starting trees are used), while greatly reducing the time spent in stepwise addition.

Parsimony ratchet approaches are implemented in DADA (Nixon, 1998) WinClada/NONA (Nixon, 1999, 2002; Goloboff, 1993-2000), TNT (Goloboff, 1999) and POY (Gladstein and Wheeler, 2001; Giribet, 2001; Janies and Wheeler, 2001). Ratcheting techniques have been used with success in supertree construction (Jones et al., 2002), and phylogenetic inference using morphological data sets (Quicke et al., 2001; Fontal-Cazalla et al., 2002; Faivovich, 2002), molecular data sets (Simmons et al. 2002; Malia et al., 2002) and combined molecular and morphological data sets (Giribet et al., 2002). Here, I propose a simple method that expands some of the concepts of this strategy to the likelihood framework.

THE LIKELIHOOD RATCHET

The algorithm proposed here extends the concept of the Parsimony Ratchet (Nixon, 1999) to phylogenetic inference under maximum likelihood: the 'Likelihood Ratchet'. The steps of the algorithm are outlined in Figure II-1.

1) A tree is generated using a fast distance algorithm (e.g. neighbor-joining).
2) This tree is used as a starting tree in a standard heuristic branch-swapping routine (e.g. NNI, SPR or TBR). The search continues until it converges on an optimum. Alternatively, a time limit or a maximum number of rearrangements can be specified after which point the search terminates and the optimal solution for this iteration is stored.
3) A random sample drawn from the data set is reweighted. This step will change the tree landscape and allow the search to move away from the optimum of step 2.
4) A distance tree based on the reweighted data is constructed.
5) The original weighting scheme is restored. The tree from step 4 is used as a starting tree in a standard heuristic branch-swapping routine. The search continues until it converges on an optimum. Alternatively, a time limit or a maximum number of rearrangements can be specified after which point the search terminates and the optimal solution for this iteration is stored.
6) The search returns to step 3. Steps 3 through 5 are repeated until a predefined number of iterations is reached.
7) When the predefined number of iterations is reached, the optimal solution(s) from among all iterations is selected.

The rationale for steps 3 through 5 is that, in order to escape from local optima, a perturbation of the tree landscape may turn 'hills' into 'valleys' and vice versa, such that a search that has converged on an optimum can escape from it while retaining much of the phylogenetic signal already identified. A different set of randomly sampled characters is drawn during each iteration because if the same set of characters is reweighted in the same way each time, the search could potentially cycle between two local optima: one in the unweighted landscape and one in the reweighted landscape.

IMPLEMENTATION

The likelihood ratchet algorithm is readily implemented using a modified input file for PAUPRat (Sikes and Lewis, 2001). The input file contains instructions for reweighted sample size, number of iterations, model of sequence evolution applicable and additional options for the branch swapping cycles (e.g. branch swapping algorithm and time or rearrangement limits per iteration). Based on the settings in the input file, PAUPRat constructs a script file which is executed in PAUP* (Swofford, 2002) in combination with the aligned sequence data set in NEXUS (Maddison et al., 1997) format. An example of a likelihood ratchet input file can be obtained from http://www.sfu.ca/~rvosa/likelihoodratchet.

The weighting scheme used in this study is the default "uniform" setting of PAUPRat (Sikes and Lewis, 2001). Under this scheme, the initial weight of all characters is set to 1 (other options are "additive" and "multiply", both of which preserve *a priori* defined weighting schemes such as codon position weighting though they differ in their upweighting methods). A user defined percentage of characters is drawn with replacement from the data. To the initial weight of each character chosen an additional

weight of 1 is added. Because characters are sampled with replacement, some characters may have their weights adjusted multiple times. For example, for

100 characters and an initially defined percentage of 25, the number of characters to sample would be set to *nmod*(0.25*100)=25 (where *nmod* is the modulo, i.e. the result of a division rounded down to the nearest integer) and a loop would be executed 25 times. Each time through this loop, 1 character is drawn at random and its weight increased by 1. Through this scheme, it is possible that 2 characters are selected for reweighting twice, resulting in 2 characters with weight 3 and 21 characters with weight 2. Thus, only 23 characters (not 25) will have weights greater than 1 due to the fact that 2 characters were chosen twice for reweighting (Lewis, pers. comm.).

Before turning to tests of concept, a remark on the fraction of characters to be reweighted is necessary. In the hypothetical case where none of the data are reweighted between cycles each iteration starts out from the same NJ tree. The likelihood landscape may be structured in such a way that this starting tree is incongruent with the generating tree, e.g. because of long branch attraction (Felsenstein, 1978b; Huelsenbeck and Hillis, 1993; Huelsenbeck, 1995), and that it is located on or near a local optimum such that branch swapping will not escape from it. Searches will consistently converge on a wrong result in this scenario (Figure II-2a). As the reweighted sample size increases, starting trees will become different such that the probability of consistently converging on the same suboptimal solution in a multimodal likelihood landscape decreases (Figure II-2b). However, after a certain point the data will be reweighted to such an extent that the time used during branch swapping to make up for the suboptimal topology of the starting tree

will take longer than constructing a starting tree using stepwise addition, in which case the purpose of the ratchet is defeated (Figure II-2c).

In the extreme case where the weighting scheme is altered such that the reweighted landscape loses all similarity with the original landscape each iteration will effectively start out from random starting trees, in which case many branch swapping cycles have to take place before reasonable results are obtained. Nonetheless, searches will not consistently converge on the wrong solution, if given enough time and enough iterations.

TESTS OF CONCEPT

To assess the performance of the likelihood ratchet algorithm relative to standard tree searching methods, I (i) compared the time needed by the likelihood ratchet to converge or improve on the optimal result obtained by a standard stepwise addition based search method on an ITS data set for 62 taxa; (ii) measured the time until convergence on the generating tree for the likelihood ratchet algorithm and standard stepwise addition based search methods on simulated data sets; (iii) tested the extent to which the likelihood ratchet explores tree space and effectively identifies known tree islands from Salter's data set of 30 papillomavirus sequences (Salter, 2001). Full results from these benchmarks can be found on www.sfu.ca/~rvosa/likelihoodratchet. All searches were given top cpu priority on empty nodes on a Beowulf cluster with 1.2 GHz AMD cpus running PAUP*4b10x86 under Linux.

Search time comparisons.—To compare search times between the likelihood ratchet and standard heuristic search strategies I used a 581 bp ITS data set for 62 bilimulid land snail taxa (supplied by Christine Parent and available from http://www.sfu.ca/~rvosa/likelihoodratchet). The data set contained low levels of average pairwise sequence divergence (~3.42% per site), and modest phylogenetic signal (g1=-0.86). One of five replicated heuristic searches using stepwise addition to generate starting trees returned the optimal result for this strategy (-lnL=2897.84772) after 60 hours, 28 minutes and 56 seconds of CPU time. This search was designed to perform 200 replicates with a time limit of 1080 seconds per replicate.

For the ratchet analysis of the ITS data set I ran 10 ratchet searches ranging in reweighted sample sizes from 5% to 50%. A likelihood ratchet search consisting of 200

iterations with the same time limit per iteration as the stepwise addition based searches and 10% reweighted sites needed only 24 hours, 23 minutes and 15 seconds to surpass (-lnL=2897.69274) the benchmark mentioned earlier. This finding, and results from earlier experimentation (data not shown), suggests that reweighting percentages should be fairly low. However, there is no reason to assume that the likelihood landscape for the ITS data set is particularly complex. In the remainder of this section I discuss data sets which I assume to be more complex. I have therefore analyzed the data sets discussed next using higher percentages of reweighted characters.

Inferring the generating tree.—To compare the search time until convergence on the generating tree between the likelihood ratchet algorithm and standard stepwise addition based search methods I ran a series of time limited searches using the likelihood ratchet and an equivalent series using standard stepwise addition based heuristic searching for a single 64-taxon pectinately branching non-ultrametric model tree. In order to provoke long branch attraction and thus increase the complexity of the likelihood landscape I made the terminal branches approximately 15 times longer than the internal branches (0.9375 versus 0.0645, respectively). On this tree I simulated five 1000 character sequence data sets using Seq-Gen v1.2.5 (Rambaut and Grassly, 1997). The data sets were simulated under the JC69 model of sequence evolution. For the first data set, I multiplied all branch lengths by 0.1, such that the probability of internodal change was approximately 0.0064 per site for the internal branches and approximately 0.0937 per site for the terminal branches. For the subsequent data sets I multiplied the branch lengths with 0.2 through 0.5. The simulated data sets were created with the intention of obtaining

more complex data sets. This is why I decided to increase the percentage of reweighted characters for the analysis of these data sets.

The ratchet consisted of 200 iterations on each of the data sets. Starting trees were constructed using neighbor-joining. During each reweighted cycle, a random sample of 15% of the total data set was drawn. Characters were reweighted along the same ratio as described earlier. A time limit of 200 seconds per iteration was imposed, the JC69 substitution model and the TBR branch swapping algorithm were used.

I compared the results with heuristic searches that consisted of 200 replicates that used for each replicate starting trees obtained using stepwise addition. The input order of the sequences was randomized at the start of each stepwise addition replicate. For these searches I used the same substitution model and branch swapping algorithm as that used by the likelihood ratchet. A time limit of 200 seconds of branch swapping per replicate was imposed. This limit did not include the time needed to construct the initial starting tree, which can be substantial for stepwise addition (ranging between on average 10 minutes per replicate for the least saturated data set to about 20 minutes per replicate for the most saturated data set).

I used the optimal solution obtained across the full 200 replicates of the stepwise addition searches as a benchmark. The likelihood ratchet algorithm converged faster on this result than did the stepwise addition searches. In the case where branch lengths were multiplied by 0.1 - the least saturated data set - the algorithm converged within 1 minute on a solution that took the stepwise addition search 15 minutes and 47 seconds to find. The optimal solution for the most saturated data set that took over 22 minutes to find using stepwise addition was found within 3 minutes using the 'ratchet' algorithm. For the

two least saturated data sets the generating tree was successfully returned by the likelihood ratchet and the stepwise addition searches. For the more saturated datasets, the ML trees obtained were still different from the generating tree, highlighting the difficulty of the problem. Since the objective for the analysis of the simulated data sets was to determine whether the likelihood ratchet could successfully return known trees - which it did where it was reasonable to expect - no further work was done to optimize the algorithm.

Tree landscape exploration.— I downloaded a 30 taxon 1,382-bp papillomavirus sequence data set from which all insertions and deletions are removed from ftp://ag.arizona.edu/dept/systbiol/issues/50_6/Salter.nexus. The first 449 bp of this data set consists of a long string of identical nucleotides. Cross-referencing the sequences with GenBank through a BLAST search revealed this part of the data set to be an artifact and so I removed it. From the 450th nucleotide onwards the data contains high levels of average pairwise sequence divergence (~45.9% per site). However, with a g1 skewness of -1.264 the phylogenetic signal is fairly high.

On this data set I ran a likelihood ratchet search consisting of 200 iterations, 300 seconds per iteration. To perturb the tree landscape, the algorithm drew a random sample of about 30% of the total data set. Out of this sample, about 90% of the characters had weight 2, about 9% had weight 3 and about 1% had weight 4 (exact numbers fluctuated between iterations but are available on request). The rationale for choosing the 30% reweighted characters setting for the papillomavirus data set is that the likelihood landscape for this data set is known to be complex. Under maximum likelihood, using a transition/transversion ratio of 2.0, no clock assumption and NNI branch swapping, the

tree landscape for the papillomavirus data set contains at least three unique islands (Salter, 2001). I selected this data set for analysis in order to asses whether the likelihood ratchet explores the landscape in such a way that it can identify different islands and move away from them. To facilitate that, a higher percentage of characters than in the other studies was selected for reweighting. Since the algorithm outperformed the standard stepwise addition searches no further work was done to optimize its settings. I compared the results with those obtained by a heuristic search using stepwise addition. This search consisted of 200 replicates with a time limit of 300 seconds per replicate.

For both searches I used the same substitution model as outlined by Salter (Salter, 2001). To identify the islands I used NNI as branch swapping algorithm. The ratchet algorithm took 1 hour, 1 minute and 47 seconds to identify the same two islands that took 3 hours, 57 minutes and 21 seconds using the stepwise addition search method. Neither one of the methods identified the third island within the given timeframe.

DISCUSSION

As for any search technique, the complexity of the model used, its fit to the generating function and the tree shape will all affect the efficiency. Results may vary and need therefore be treated with caution. Further work is needed to: i) determine how search time for the likelihood ratchet scales with the number of taxa; ii) determine the optimal percentage of sites to modify; iii) determine how reweighting schemes interact with user defined character sets such as codon positions; iv) see if aspects of the dataset (e.g. some measure of phylogenetic signal obtained directly from the data) itself can be used to optimize its implementation; and v) compare these sorts of searches to other search strategies, such as hot-swapping in the Bayesian framework, besides the standard stepwise addition method. It is conceivable that 'ratchet-like' approaches may do a better job here too.

The results presented in this study suggest that randomly reweighted NJ starting trees can be profitably used as a starting point to explore the likelihood landscape for large numbers of taxa. The results obtained using the likelihood ratchet are similar to, or better than those obtained using stepwise addition based starting trees, in less time. The trade-off in search time between using reweighted, suboptimal NJ starting trees combined with subsequent hill-climbing compares favorably with that of stepwise addition and hill-climbing, especially in data sets with little phylogenetic signal measured as tree score distribution skewness.

ACKNOWLEDGEMENTS

I thank Arne Mooers, Bernie Crespi, Mike Whitlock, the FAB-lab, Mike Sanderson and two anonymous reviewers for their helpful suggestions and feedback on this manuscript, Christine Parent for kindly providing me with her DNA sequence data, Biological Sciences, Simon Fraser University and NSERC/Canada for funding, and the high power computing facility HPC@SFU and Martin Siegert for use of the Beowulf infrastructure. My gratitude extends to Paul O. Lewis, who was very helpful in explaining the inner workings of PAUPRat.

REFERENCES

Charleston, M. A. 2001. Hitch-hiking: a parallel heuristic search strategy, applied to the phylogeny problem. J. Comput. Biol. 8:79-91.

Faivovich, J., 2002. A cladistic analysis of Scinax (Anura : Hylidae). Cladistics 18:367-393.

Felsenstein, J. 1978a. The number of evolutionary trees. Syst. Zool. 27:27-33

Felsenstein, J. 1978b. Cases in which parsimony or compatibility methods will be positively misleading. Syst. Zool. 27:401-410

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368-376

Fontal-Cazalla, F. M., M. L. Buffington, G. Nordlander, J. Liljeblad, P. Ros-Farre, J.L. Nieves-Aldrey, J. Pujade-Villar and F. Ronquist, 2002. Phylogeny of the Eucoilinae (Hymenoptera : Cynipoidea : Figitidae). Cladistics 18:154-199.

Giribet, G., 2001. Exploring the behavior of POY, a program for direct optimization of molecular data. Cladistics 17:S60-S70.

Giribet, G., G. D. Edgecombe, et al. 2002. Phylogeny and systematic position of Opiliones: A combined analysis of chelicerate relationships using morphological and molecular data. Cladistics 18:5-70.

Gladstein, D. and W. C. Wheeler, 2001. POY - Phylogeny Reconstruction via Optimization of DNA data. New York, NY, Division of Invertebrates, American Museum of Natural History.

Goloboff, P. A. 1999. Analyzing large data sets in reasonable times: Solutions for composite optima. Cladistics 15:415-428.

Goloboff, P. A., 1993-2000. PIWE / NONA / PHAST /SPA (Parsimony with Implied Weights). Copyright (C) P. Goloboff, Instituto Miguel Lillo, Tucuman, Argentina.

Hendy, M. P., M. A. Steel, D. Penny, and I. M. Henderson. 1988. Families of trees and consensus. Pages 355-362 in Classification and related methods of data analysis (H. H. Bock, ed.) Elsevier, New York.

Huelsenbeck, J. P. 1995. Performance of phylogenetic methods in simulation. Syst. Biol. 44:17-48

Huelsenbeck, J. P. and D. M. Hillis, 1993. Success of phylogenetic methods in the four-taxon case. Syst. Biol. 42:247-264

Janies, D. A. and W. C. Wheeler, 2001. Efficiency of parallel direct optimization. Cladistics 17:S71-S82.

Jones, K. E., A. Purvis, A. MacLarnon, O. R. P. Bininda-Emonds and N. B. Simmons, 2002. A phylogenetic supertree of the bats (Mammalia : Chiroptera). Biol. Rev. 77:223-259.

Maddison, D. R. 1991. The discovery and importance of multiple islands of most-parsimonious trees. Syst. Zool. 40:315-328.

Maddison, D. R., Swofford, D. L., and W. P. Maddison. 1997. NEXUS: an extensible file format for systematic information. Syst. Biol. 46:590–621

Malia, M. J., R. M. Adkins, et al. 2002. Molecular support for Afrotheria and the polyphyly of Lipotyphla based on analyses of the growth hormone receptor gene. Mol. Phylogenet. Evol. 24:91-101.

Moilanen, A. 2001. Simulated evolutionary optimization and local search: Introduction and application to tree search. Cladistics 17:S12-S25.

Nixon, K., 1998. Dada ver 1.9. Software and manual. Published by the author, Trumansburg, NY.

Nixon, K. 1999. The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis. Cladistics 15:407-414.

Ota, S., and W. H. Li. 2000. NJML: A hybrid algorithm for the neighbor-joining and maximum-likelihood methods. Mol. Biol. Evol. 17:1401-1409.

Ota, S., and W. H. Li. 2001. NJML+: An extension of the NJML method to handle protein sequence data and computer software implementation. Mol. Biol. Evol. 18:1983-1992.

Quicke, D. L. J., J. Taylor, and A. Purvis. 2001. Changing the landscape: A new strategy for estimating large phylogenies. Syst. Biol. 50:60-66.

Rambaut, A. and N.C. Grassly, 1997. Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13:235-238

Rogers, J., and D. Swofford. 1999. Multiple local maxima for likelihoods of phylogenetic trees: a simulation study. Mol. Biol. Evol. 16:1079-1085

Saitou, N., and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4:406–425.

Salter, L. A., 2001. Complexity of the Likelihood Surface for a Large DNA Dataset. Syst. Biol. 50: 970-978

Salter, L. A., and D. K. Pearl. 2001. Stochastic search strategy for estimation of maximum likelihood phylogenetic trees. Syst. Biol. 50:7-17.

Sikes, D. S., and P. O. Lewis. 2001. PAUPRat: A tool to implement Parsimony Ratchet searches using PAUP*, beta software, version 1.

Simmons, M. P., H. Ochoterena, et al. 2002. Amino acid vs. nucleotide characters: challenging preconceived notions. Mol. Phylogenet Evol. 24:78-90.

Steel, M. 1992. The complexity of reconstructing trees from qualitative characters and subtrees, J. Classif. 9:91-116.

Steel, M. 1994. The maximum likelihood point for a phylogenetic tree is not unique. Syst. Biol. 43:560-564

Sumrall, C. D., C. A. Brochu, and J. W. Merck. 2001. Global lability, regional resolution, and majority-rule consensus bias. Paleobiology 27:254-261.

Swofford, D. L., G. Olsen, P. Waddell and D. Hillis. 1996. Phylogenetic inference. Pp. 407-509 in D. Hillis, C. Moritz and B. Mable, eds. Molecular Systematics. 2nd edition. Sinauer, Sunderland, Mass.

Swofford, D. L. 2002. PAUP*: phylogenetic analysis using parsimony, version 4.0b10.

FIGURE II-1 FLOWCHART DIAGRAM OF THE LIKELIHOOD RATCHET STRATEGY.

See text for details.

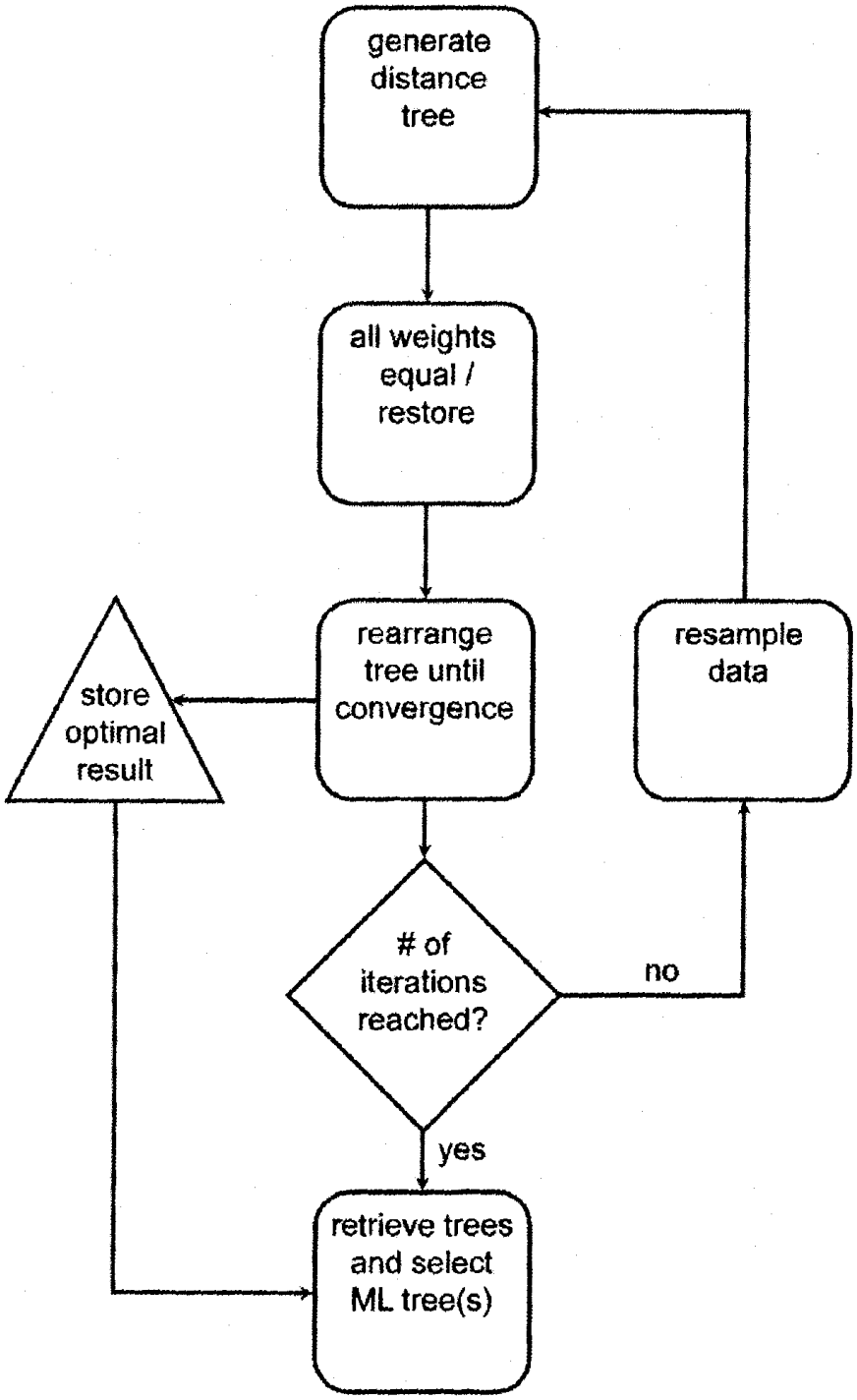
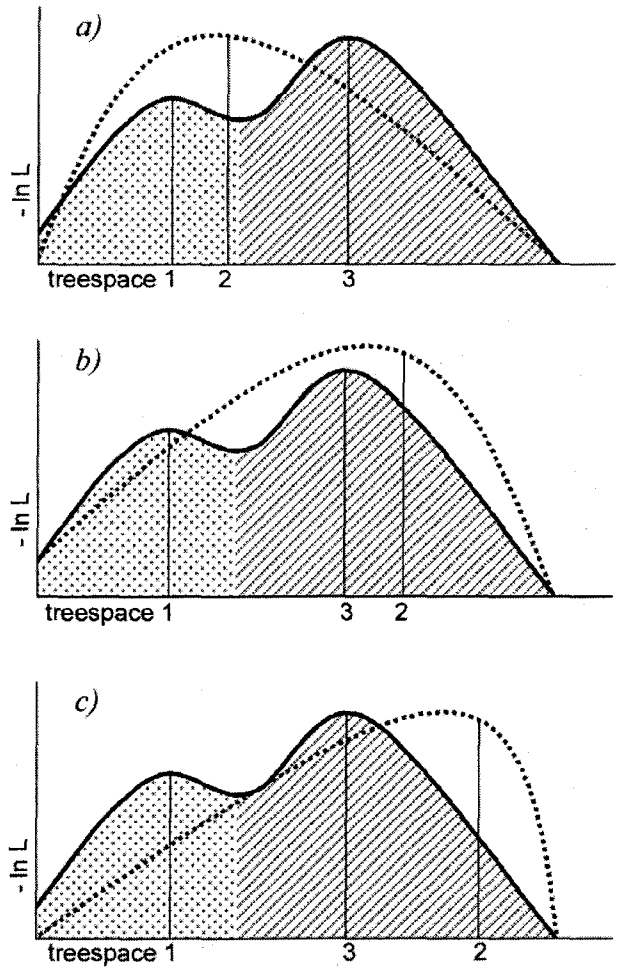


FIGURE II-2 ESCAPING FROM LOCAL OPTIMA BY CHANGING A TREE LANDSCAPE

The x-axis represents a hypothetical tree space. Points near each other on this axis represent trees that are in each other's vicinity in terms of the number of rearrangements needed to go from one tree to the next. The y-axis represents the fit of the trees to the data (in this case the –lnL). The dotted area represents the set of trees from which only the locally optimal tree 1 can be reached. The hatched area represents the set of trees from which the global optimum (tree 3) can be reached. The dashed lines represent changed tree landscapes obtained by reweighting. Note that, although the analogy of a "tree landscape" may hold to a certain extent for the case of maximum likelihood, this is not the case for distance algorithms. The reweighted starting trees in (a-c), albeit obtained by distance methods, are therefore identified within tree landscapes under maximum likelihood given the same weighting scheme and substitution model as were used to construct the distance trees. In (a), the reweighted landscape is structured such that starting tree 2 obtained from it will lead the search back to locally optimal tree 1. As the percentage of reweighted sites is increased, starting trees – tree 2 in (b) and (c) - may be obtained that allow the search to move towards the global optimum. Depending on the structure of the reweighted landscape this is done in a more (b) or less (c) efficient way. See text for details.



CHAPTER III - ACCELERATED METROPOLIS-HASTINGS COUPLED MARKOV CHAIN MONTE CARLO BURN-IN BY ITERATIVE JACKKNIFING[2]

Rutger A. Vos

[2] This chapter is in revision with *Systematic Biology*

ABSTRACT

The burn-in phase of phylogenetic inference using Metropolis-Hastings coupled Markov chain Monte Carlo (MC3) is typically discarded, but is computer-intensive, especially with large data sets. Here I introduce an approach that accelerates this phase by iteratively jackknifing randomly drawn samples from the character set, so that the optimality landscape changes in cycles and Markov chains may move away more easily from local optima. The method performs well in comparison with commonly used – and more computationally intensive – settings for the MC3 approach: with the present algorithm, solutions with a greater likelihood are found in less time and so it can therefore be used to infer the MAP tree, which can be used as the input tree for runs that approximate the posterior distribution. The method proposed here may be of particular use in exploring complex tree landscapes of conflicting phylogenetic signal.

Keywords: Bayesian systematics, burn-in, likelihood ratchet, Metropolis-Hastings coupled Markov chain Monte Carlo, *rbc*L, conflicting phylogenetic signal

INTRODUCTION

There is increasing interest in the large-scale phylogenetic analysis of DNA sequences (e.g. 'AToL' projects, http://www.nsf.gov/pubs/2003/nsf03536/nsf03536.htm). However, the number of possible solutions – phylogenies – to evaluate under optimality criteria increases hyper-exponentially as sequences are added such that it is impossible to evaluate all trees for more than about 50 taxa (Felsenstein, 1978). If the optimality landscape is unimodal only a small subset of possible trees needs to be evaluated using for example a hill-climbing algorithm, and phylogenetic analysis of large data sets is guaranteed to return the optimal solution under such conditions. However, these conditions are often not met. Maddison (1991) has shown how "tree islands" (Maddison, 1991) exist under maximum parsimony, and though the likelihood landscape has not been fully characterized, theoretical (Steel, 1994) and empirical evidence using both simulated (Chor et al., 2000; Rogers and Swofford, 1999) and real (Salter, 2001) data has demonstrated that multiple, local maxima of likelihood also exist. In the case where these optima exist in tree space (Salter, 2001; as opposed to multiple optima in branch length space; see Steel, 1994) such multiple maxima pose problems for heuristic hill climbing search strategies, which can arrive at locally optimal rather than globally optimal trees. One common method to mitigate against this is to start hill-climbing replicates from disparate points in tree space (Felsenstein, 2003). These starting points are often obtained by optimal stepwise addition of sequences in a random input order; unfortunately the construction of starting trees using the stepwise addition of large numbers of sequences can be a time-consuming process under certain optimality criteria (Felsenstein, 2003). A speedier alternative, starting with random starting trees, saves little: such random trees

are expected to be so far from any optimum that an excessive number of refinements and subsequent fits to the optimality criterion result (Vos, 2003).

More complex heuristic search strategies to avoid getting "stuck" in local optima have been developed (Charleston, 2001; Goloboff, 1999; Moilanen, 2001; Nixon, 1999; Ota and Li, 2000; Ota and Li, 2001; Quicke et al., 2001). Among these are strategies that rely on iterative perturbations of the tree landscape (Nixon, 1999; Quicke et al., 2001). For example, by temporarily assigning a different weight to a random sample drawn from the data set, the optimality landscape will change, allowing the search to escape from a local optimum. After reaching a new optimum, the initial weighting scheme is restored and the search continues, hopefully out of the reach of the original local optimum. This strategy, known as the "Parsimony Ratchet" (Nixon, 1999), has the advantage that episodes of reweighted hill-climbing preserve some of the original phylogenetic signal (rather than losing it entirely as happens when random starting trees are used), while greatly reducing the time spent in generating viable starting points. This approach has been found to be quite efficient for big trees under maximum parsimony (Nixon, 1999), and some of its concepts have been extended to the likelihood framework (Vos, 2003).

THE BAYESIAN FRAMEWORK

A relatively new approach to phylogenetic inference uses Bayes' theorem, which, in a phylogenetic context, can be stated as: $f(\tau,v,\theta|X) = (f(\tau,v,\theta) * f(X|\tau,v,\theta)) / f(X)$; where τ = topology of the tree; v = vector of branch lengths; θ = vector of model parameters; X = data matrix. The property of interest here is the *posterior probability* of tree shape $f(\tau|X)$, branch lengths $f(v|X)$ and model parameters $f(\theta|X)$, given the data. This posterior probability is heavily (indeed, with flat priors, solely) influenced by the likelihood of the

data. Phylogenetic inference within the likelihood framework (Felsenstein, 1981) is well-established, and much of its adaptation to the Bayesian framework has been straightforward. However, the exploration of tree space and the optimality landscape is different from the hill-climbing approach commonly used in the likelihood framework. In Bayesian phylogenetic inference, tree space and the optimality landscape are typically explored using the Markov chain Monte Carlo (Larget and Simon, 1999; Mau et al., 1999; Yang and Rannala, 1997), a proposal mechanism whereby a new state (ψ') replaces the current state (ψ) with probability $R = \min [\ 1, (f(X|\psi') / f(X|\psi) * f(\psi') / f(\psi) * q(\psi|\psi') / q(\psi'|\psi'))\ ]$, where ψ = current state, combination of τ, v and θ; $q(\psi)$ = probability of proposing current state; ψ' = new state, combination of τ, v and θ; $q(\psi')$ = probability of proposing new state. Whether the new state is accepted is determined by generating a random variable U, uniformly distributed on interval (0,1). If $U < R$; $\psi = \psi'$.

If a Markov chain is constructed properly, and run long enough, the frequencies with which states are visited converge on values proportional to the states' posterior probabilities, and hence serve as valid approximations of them. However, unless the Markov chain is run for an infinite number of generations, only the times states are visited after the *burn-in* phase, i.e. the first part of the chain before stable sampling from the target distribution, are of interest. Burn-in can take a long time, and better areas of tree space may be found long after apparent stabilization has set in (Huelsenbeck et al., 2002).

A recent approach to speeding up the exploration of the optimality landscape comes from the Metropolis-Hastings-coupled Markov chain Monte Carlo (Huelsenbeck et al., 2001), which uses multiple chains that search the landscape in parallel. All but one

of the chains are "heated", such that for chain i the posterior probabilities are raised to the power β, where $\beta = 1 / (1 + (i - 1) * \lambda)$; i = the index of the chain; λ = the temperature. This has as an effect that the probability of accepting "worse" solutions increases with increasing temperatures (λ) and chain indices (i), so that these chains accept suboptimal proposals more often, and so may cross suboptimal valleys. The different chains attempt to exchange their results with a certain probability, so that the cold chain can reach stationary sampling from the target distribution after fewer generations, at a cost of greater computational intensity per generation. Luckily, given the modularity of the approach, the MC3 framework can be parallelized to great effect (Altekar et al., 2004). Here, I present a different approach to accelerating the burn-in phase, an approach that can be used in conjunction with MC3. The algorithm draws on some concepts that have proven useful in the parsimony (Nixon, 1999), and likelihood (Vos, 2003) framework. At its core is a series of short Monte-Carlo chains run on subsets of the original columns of input data.

THE ALGORITHM

The algorithm is depicted in Figure III-1. The steps of the algorithm are:

1. Obtain a starting state (i.e. a tree shape, a set of branch lengths and model parameters). These can all be random values – though prior knowledge can also be incorporated.
2. From this starting state, initiate and run a Markov chain for a predefined number ("X" in Figure III-1) of generations, storing the visited states.
3. Unless a predefined number of Y iterations has been reached, return to step 2 by:
    a. Selecting the most recently visited state from the previous iteration as a starting point.
    b. Every second iteration (i.e. when "Y" in Figure III-1 is an even number), randomly jackknifing a sample of predefined size ("Z") from the characters.

Once the predefined number of iterations is reached, the best tree from the set of stored trees is selected, which can either be used as an approximation of the MAP tree, or as a starting point from which to commence a properly constructed Markov chain from which posteriors on parameter values, trees and branch lengths can be approximated.

IMPLEMENTATION

The algorithm is implemented by a wrapper, written in Perl (v.5.6.1), that creates handles to MrBayes' v.3.0B4 (Ronquist and Huelsenbeck, 2003) standard input and standard output. The wrapper writes the jackknifing and MC3 commands to MrBayes' standard input. The wrapper reads the last accepted tree from the *.t file and writes it to

MrBayes' standard input as a user tree and starts another chain, issuing jackknifing commands every other iteration. The script for implementing the IJ with MrBayes is available at http://search.cpan.org/~rvosa/Bio-Phylo/.

RESULTS AND DISCUSSION

To test the performance and parameters of the algorithm presented here I used two data sets. The first is *TreeZilla* (*sensu* Anne Yoder), a 1428 nucleotide data set of 500 sequences of the *rbc*L locus of seed plants (Chase et al., 1993). The TreeZilla data set is known to be difficult to analyze, though perhaps mostly due to its size and it has become a *de facto* benchmark data set for testing heuristic search strategies on large data sets (Chase et al., 1993; Nixon, 1999; Savolainen et al., 2000).

The second data set consists of two concatenated, simulated alignments of 500 sequences, 1000 nucleotides each. These two alignments were simulated on different generating trees (which were obtained from a randomly labeled Yule process). Hence, the phylogenetic signals contained in the two concatenated data sets yields a combined data set with conflicting signal. Recent theoretical work (Mossel and Vigoda, 2005) has shown that MCMC analyses on such mixed data may be misleading, with poor mixing, slow convergence and inflated posterior probabilities. I included mixed data to explore this issue empirically, from the perspective of the generating trees being 'islands' in the tree landscape and to test whether the iterative jackknifing algorithm explores this bimodal landscape more effectively than does the standard MCMC approach.

Simulations were done using Mesquite (Maddison and Maddison, 2001). All calculations were done using the MPI version of MrBayes' v.3.0B4 (Ronquist and Huelsenbeck, 2003) on a cluster of 96 dual 1.2GHz Athlon processors. Uniform priors were used for all parameters; the first starting trees were random trees.

To illustrate how the two approaches, iterative jackknifing and MC3, differ in their exploration of tree space I calculated the average Symmetric Difference Metric (Steel, 1988) over a sliding window of five trees through a typical analysis run on TreeZilla using both approaches. Sets of trees that differ more in topology will have higher SDM values. A representative example of the behavior is shown in Figure III-2. The solid line shows an MC3 run using 4 chains and a temperature of 0.2 (the default temperature in MrBayes). Note how, on a number of occasions, there is a sudden peak in the tree-to-tree distances. This is associated with a swap between the cold chain and a heated chain that has "discovered" a better area of the optimality landscape, with radically different tree shapes. The cyclical "seesaw" pattern in the single-chain iterative jackknifing approach (dashed line) is associated with the iterations: when a new iteration commences, the optimality landscape changes because a fraction (in this example 20%) of the sites is jackknifed or restored, and the chain quickly moves to a new area of tree space.

The overall average tree-to-tree distance is higher for iterative jackknifing (~366.08) then for the MC3 approach (~195.47), i.e. the iterative jackknifing approach explores more different trees than the MC3 approach. This general pattern – local refinement interspersed with sudden peaks for the MC3 approach, and a "seesaw" pattern with more different trees for the iterative jackknifing approach – is repeated under all conditions, both with respect to the percentage of jackknifed sites and with respect to various settings for the number of chains and the temperature (full results available in the

online appendix on this journal's website). The IJ approach accomplishes its main objective of searching larger areas of tree space.

In addition, the iterative jackknifing approach using a single chain finds better solutions in less time than the MC3 approach using commonly used numbers of chains and temperatures for the two datasets tested here. Representative results for the TreeZilla dataset are shown in Figure III-4: the dashed line shows the iterative jackknifing approach, the solid line shows the MC3 approach with 4 chains and a temperature of 0.2 for the three hot chains for the same dataset. The (even) cycles, where the iterative jackknifing approach reaches the highest log likelihood values, should be disregarded here because they represent the log likelihood values on the jackknifed (here 20%) data set and so the values are necessarily higher, as fewer sites are evaluated. However, after the ninth iteration, when the cumulative number of generations is 450,000, the last visited states using the iterative jackknifing approach have a higher likelihood than those of the last visited states of the MC3 approach after 450 000 generations; and in subsequent iterations the IJ approach continues to visit trees with higher –lnL scores than the best visited by the MC3 approach. The important thing to note in this comparison is that the IJ approach uses only a single chain while the MC3 approach uses 4 chains. The number of processor cycles scales roughly linearly with the number of chains; hence the Iterative Jackknifing approach needs approximately 1/4 the total number of cycles to surpass the standard MC3 approach.

Mixed Signal Data

The expectation for the mixed data set was that it would contain a 'bottleneck' (Mossel and Vigoda, 2005) such that visiting both areas of optimality would require

either many random starting points, or strong reweighting or jackknifing so that the tree landscape would be altered to such an extent that the search would be able to escape from the 'pull' of one area of optimality and move to the other from iteration to iteration. Therefore, I analyzed this data set also under higher fractions of jackknifed characters (up to 50%).

Figure III-5a & b show how the iterative jackknifing algorithm and the MCMC approach move through tree space. The charts show, respectively, the distance from the first and the second generating tree, through time. The black diamonds represent trees sampled during the course of iterative jackknifing. (Only those trees sampled on the 'true' data set are shown, the jackknifed iterations are omitted and their time frames are condensed. Hence, 30,000 generations of the IJ algorithm are equivalent to 60,000 MCMC generations.) Compared to the gray squares, i.e. the trees sampled during a standard Markov chain, the IJ trees again show a more 'see-saw' pattern, albeit less regular due to the conflicting signal. Especially near the end of the analysis, the iterations are cycling between the vicinities of the two generating trees, approaching both of them more closely than does the regular MC3 approach.

The likelihood progress curve in Figure III-6 for the mixed signal data set shows a similar pattern as that for TreeZilla (Figure III-4) insofar that the IJ algorithm again finds better solutions than the regular MCMC approach. Notable however is the noisier curve for regular MCMC: in addition to the macro-pattern of improving likelihood there is a micro-pattern presumably caused by the irregularity – multimodality – of the optimality landscape of the mixed signal data.

"Ratchet" approaches, and the IJ method presented here, are understood to work because the optimality landscape changes during the reweighted (or jackknifed) cycles, so that local searches can move away from local optima. The success of this approach depends on the extent to which the landscape is altered: if the change is slight, the local optimum might remain in place, and the search will not escape from it. If many characters are reweighted (or jackknifed), the altered optimality landscape loses all similarity with the unaltered one. The search may move away from what was a local optimum on the unaltered landscape, but in the direction of suboptimal solutions that are meaningless on the unaltered optimality landscape, so that the starting trees obtained in this way will effectively be random requiring a lot of refining during the unweighted cycles. The optimal sample size should therefore be an intermediate value; for example, in Figure III-3 for the TreeZilla data set the optimum lies at 25%. In the test case using simulated data with conflicting phylogenetic signal I found that higher values (up to 50%) yielded better results.

CONCLUSION

The iterative jackknifing approach introduced here finds better solutions in the same number of generations than MC3 with a standard number chains and standard temperatures. Though the MC3 approach with more chains or higher temperatures may eventually be expected to outperform the IJ solutions, it will do so at a concomitant cost in greater computational intensity. I propose the IJ approach as a method to speed up burn-in for large datasets. As datasets with hundreds and even thousands of sequences become the norm, the IJ may be of great value. It is important to emphasize however that the series of visited states during the IJ run does not constitute a pure Markov chain and the resulting trees cannot be summarized in a consensus to obtain posteriors (e.g. on clades). However, burn-in trees are discarded in any case, and the results should be used as a starting point for further analysis using properly constructed chains. Alternatively, the IJ approach can be used as a means of approximating the maximum a posteriori (MAP) estimate of phylogeny – in which case no further pure Markov chain searches are necessary.

ACKNOWLEDGEMENTS

The author would like to thank Rod Page, Paul Lewis and an anonymous reviewer of *Systematic Biology* for their helpful comments on this manuscript; the tree builders at SCS (former CSIT), Arne Mooers for suggesting expanding the ratchet to the MC3 framework and the SFU FAB* lab for helpful comments and feedback in the course of drafting this manuscript. I am thankful to Martin Siegert of HPC@SFU for providing access to the high-performance computing facility at Simon Fraser University.

REFERENCES

Altekar, G., S. Dwarkadas, J. P. Huelsenbeck, and F. Ronquist. 2004. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics 20:407-415.

Charleston, M. A. 2001. Hitch-hiking: a parallel heuristic search strategy, applied to the phylogeny problem. J. Comput. Biol. 8:79-91.

Chase, M. W., D. E. Soltis, R. Olmstead, D. Morgan, D. Les, B. D. Mishler, M. Duvall, R. Price, H. Hills, Y.-L. Qiu, K. Kron, J. Rettig, E. Conti, J. Palmer, J. Manhart, K. Sytsma, H. Michaels, W. J. Kress, M. J. Donoghue, W. D. Clark, M. Hedren, B. S. Gaut, R. Jansen, K.-J. Kim, C. Wimpee, J. Smith, G. Furnier, S. Straus, Q.-Y. Xiang, G. Plunkett, P. S. Soltis, S. Swensen, L. Eguiarte, G. Learn Jr., S. Barret, S. Graham, S. Dayanandan, and V. Albert. 1993. Phylogenetics of seed plants: an analysis of nucleotide sequences from the plastid gene *rbc*L. Ann. MO Bot. Gard. 80:528-580.

Chor, B., M. P. Hendy, B. R. Holland, and D. Penny. 2000. Multiple Maxima of Likelihood in Phylogenetic Trees: An Analytical Approach. Mol. Biol. Evol. 17:1529-1541.

Felsenstein, J. 1978. The number of evolutionary trees. Syst. Zool. 27:27-33.

Felsenstein, J. 1981. Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach. J. Mol. Evol. 17:368-376.

Felsenstein, J. 2003. Inferring phylogenies. Sinauer Associates, Inc., Sunderland, Massachussets.

Goloboff, P. A. 1999. Analyzing large data sets in reasonable times: Solutions for composite optima. Cladistics 15:415-428.

Huelsenbeck, J. P., B. Larget, R. E. Miller, and F. Ronquist. 2002. Potential Applications and Pitfalls of Bayesian Inference of Phylogeny. Syst. Biol. 51:673-688.

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

Larget, B., and D. L. Simon. 1999. Markov Chain Monte Carlo Algorithms for the Bayesian Analysis of Phylogenetic Trees. Mol. Biol. Evol. 16:750-759.

Maddison, D. R. 1991. The discovery and importance of multiple islands of most-parsimonious trees. Syst. Zool. 40:315-328.

Maddison, W. P., and D. R. Maddison. 2001. Mesquite: a modular system for evolutionary analysis.

Mau, B., M. A. Newton, and B. Larget. 1999. Bayesian Phylogenetic Inference via Markov Chain Monte Carlo Methods. Biometrics 55:1-12.

Moilanen, A. 2001. Simulated evolutionary optimization and local search: Introduction and application to tree search. Cladistics 17:S12-S25.

Mossel, E., and E. Vigoda. 2005. Phylogenetic MCMC algorithms are misleading on mixtures of trees. Science 309:2207-2209.

Nixon, K. 1999. The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis. Cladistics 15:407-414.

Ota, S., and W. H. Li. 2000. NJML: A hybrid algorithm for the neighbor-joining and maximum-likelihood methods. Mol. Biol. Evol. 17:1401-1409.

Ota, S., and W. H. Li. 2001. NJML+: An extension of the NJML method to handle protein sequence data and computer software implementation. Mol. Biol. Evol. 18:1983-1992.

Quicke, D. L. J., J. Taylor, and A. Purvis. 2001. Changing the landscape: A new strategy for estimating large phylogenies. Syst. Biol. 50:60-66.

Rogers, J., and D. L. Swofford. 1999. Multiple local maxima for likelihoods of phylogenetic trees: a simulation study. Mol. Biol. Evol. 16:1079-1085.

Ronquist, F., and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572-1574.

Salter, L. A. 2001. Complexity of the likelihood surface for a large DNA dataset. Syst. Biol. 50:970-978.

Savolainen, V., M. W. Chase, S. B. Hoot, C. M. Morton, D. E. Soltis, and C. Bayer. 2000. Phylogenetics of Flowering Plants Based on Combined Analysis of Plastid *atpB* and *rbcL* Gene Sequences. Syst. Biol. 49:306-362.

Steel, M. A. 1988. Distribution of the symmetric difference metric on phylogenetic trees. Siam J. Disc. Math 1:541-551.

Steel, M. A. 1994. The maximum likelihood point for a phylogenetic tree is not unique. Syst. Biol. 43:560-564.

Vos, R. A. 2003. Accelerated Likelihood Surface Exploration: The Likelihood Ratchet. Syst. Biol. 52:368-373.

Yang, Z., and B. Rannala. 1997. Bayesian Phylogenetic Inference Using DNA Sequences: A Markov Chain Monte Carlo Method. Mol. Biol. Evol. 14:717-724.

FIGURE III-1 THE ITERATIVE JACKKNIFING ALGORITHM

Flowchart diagram of the iterative jackknifing algorithm for Bayesian searches of treespace. See text for details.

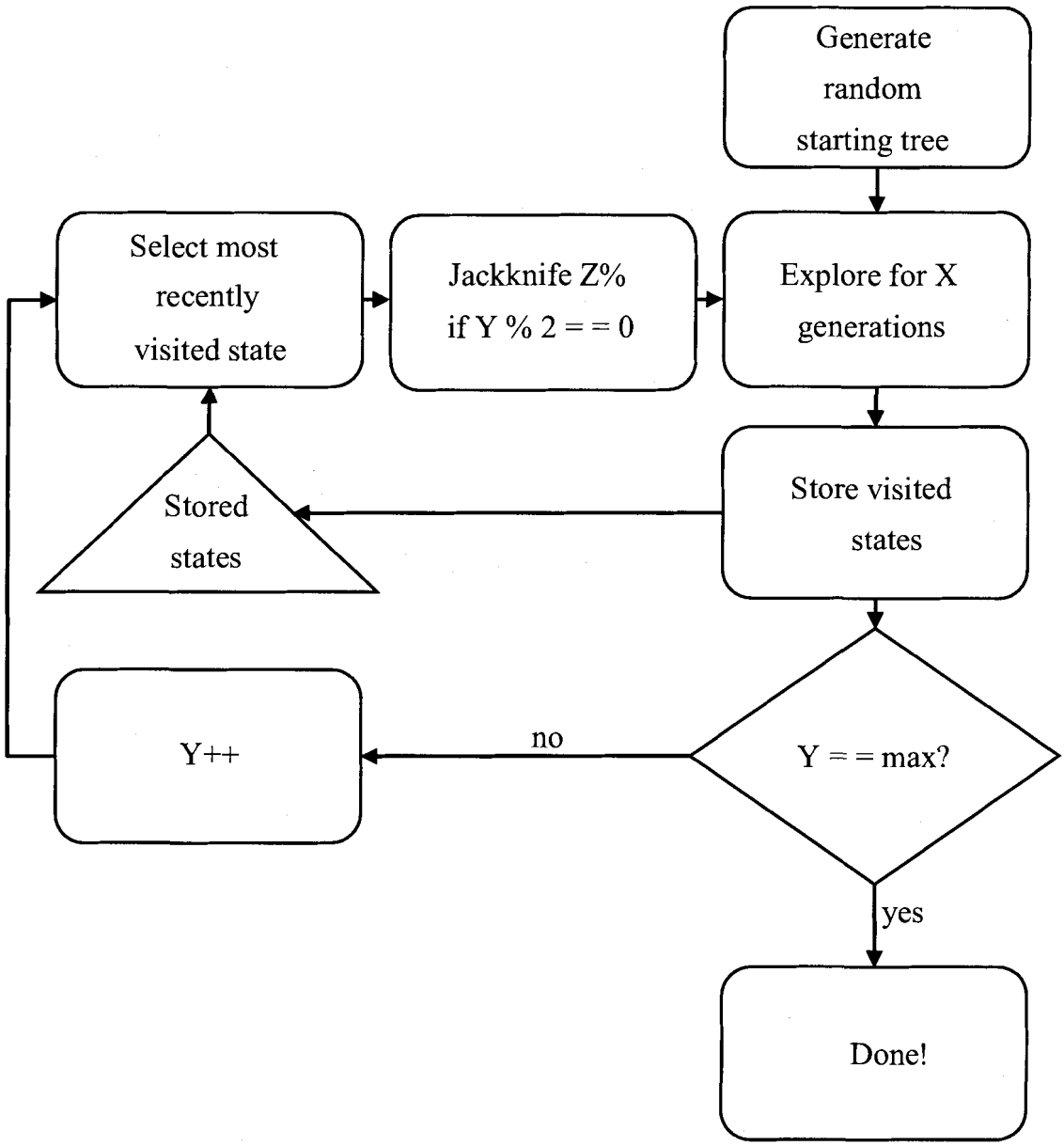


FIGURE III-2 SLIDING WINDOW ANALYSIS OF TREE-TO-TREE DISTANCES

Sliding window analysis of symmetrical difference metric tree-to-tree distances over the course of a representative run using iterative jackknifing (dashed line) and MC3 (solid line). Higher values mean the sampled trees are topologically more different, implying the run is sampling more disparate areas of treespace.

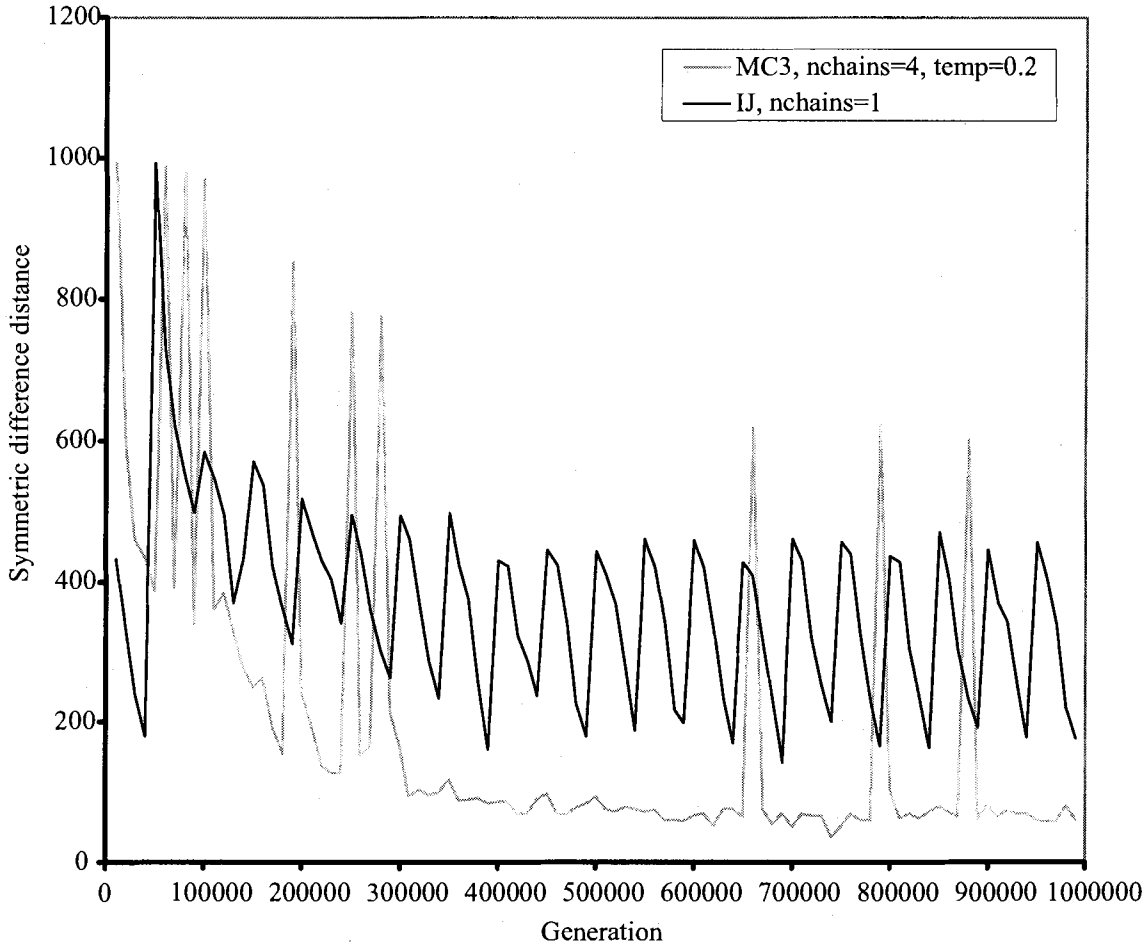


FIGURE III-3 OPTIMAL LIKELIHOOD SCORES FOR *RBC*L AFTER 10^6 GENERATIONS

Optimal solutions (ML fit values) found after 10^6 generations (divided over 10 iterative jackknifing cycles) as a function of jackknifed sample size for the *rbc*L dataset.

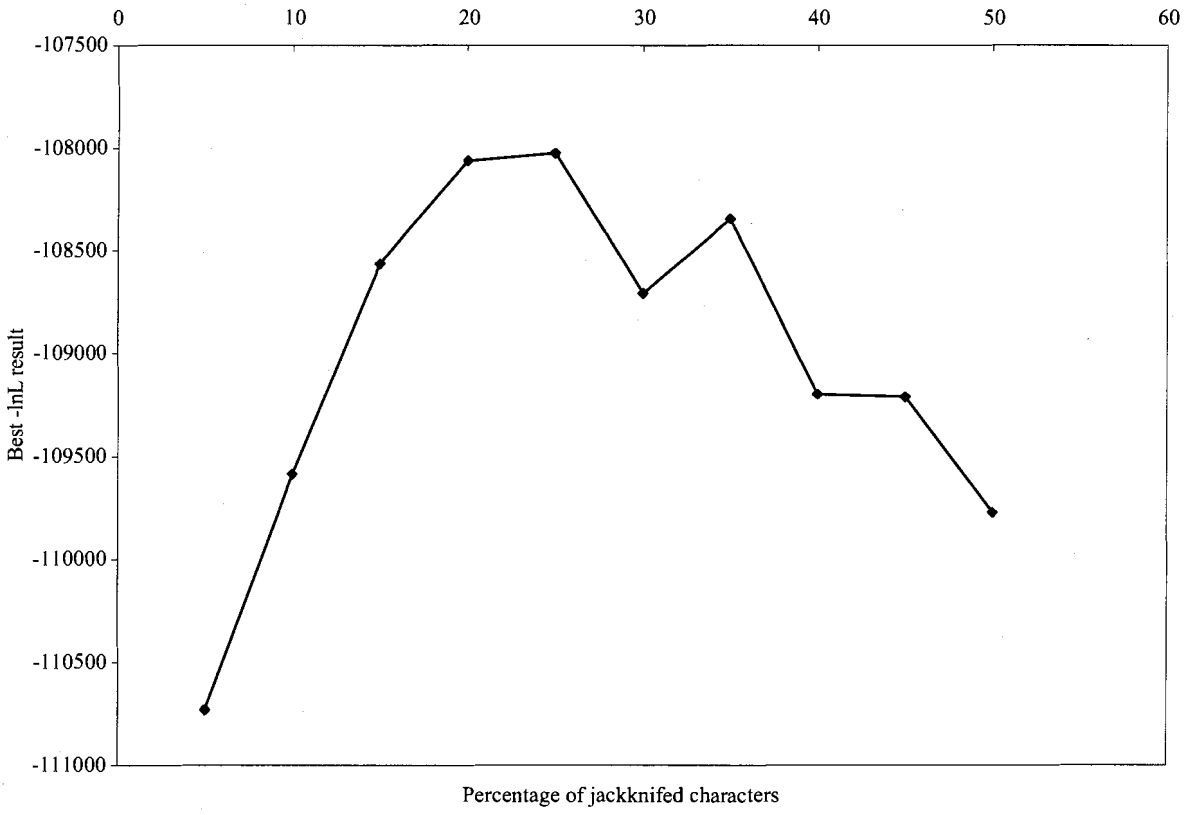


FIGURE III-4 CYCLICAL LIKELIHOOD OPTIMIZATIONS

Log likelihood tree scores over the course of a representative run using iterative jackknifing (dashed line) and MC3 (solid line) for the *rbc*L dataset. The see-saw pattern in the iterative jackknifing analysis is caused by the jackknifing; with fewer characters in the matrix the likelihood scores become inflated. The high peaks in the pattern are those obtained during jackknifed iterations; and are therefore not informative. Of interest are the lower peaks, which by generation 450,000 surpass the scores obtained using MC3. See text for details.

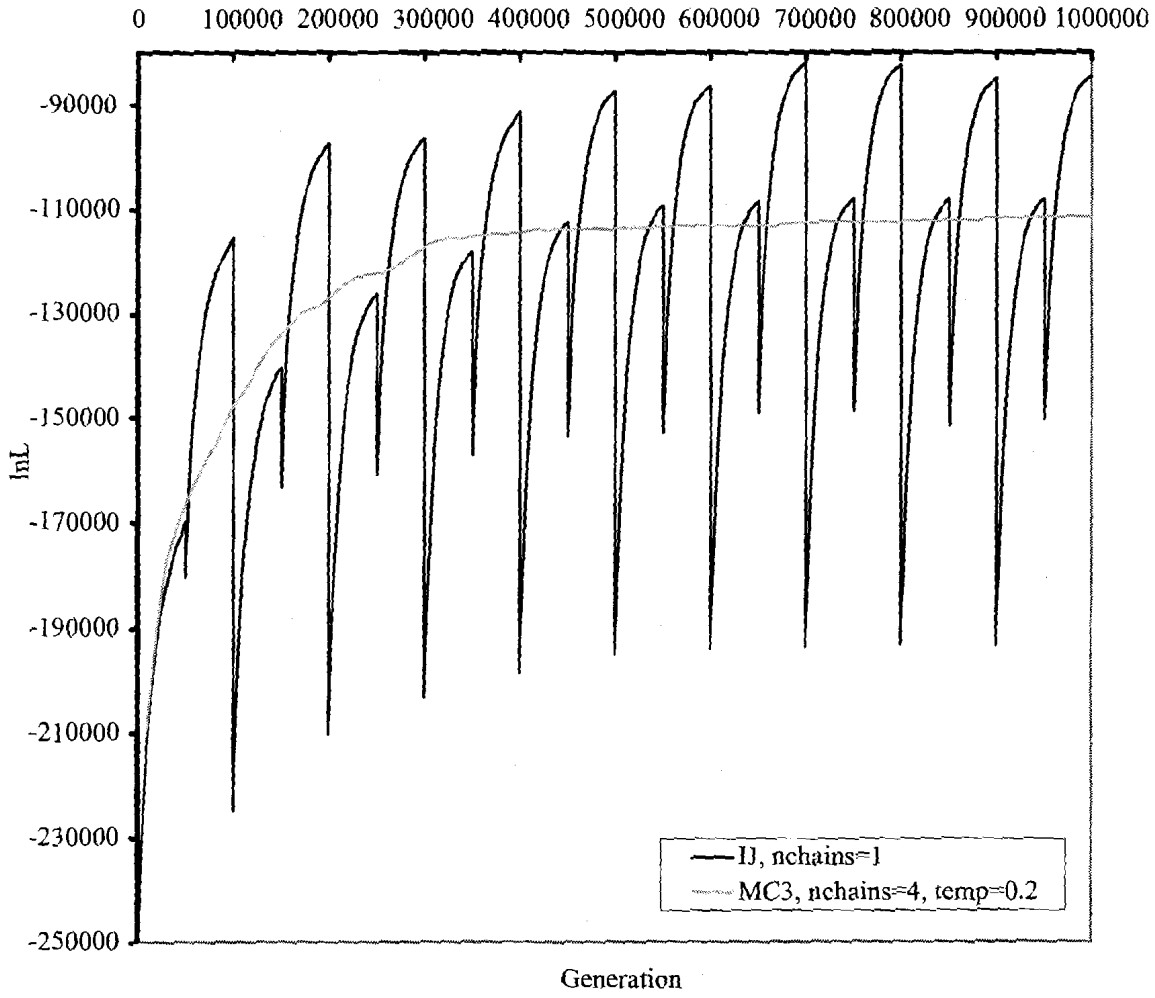
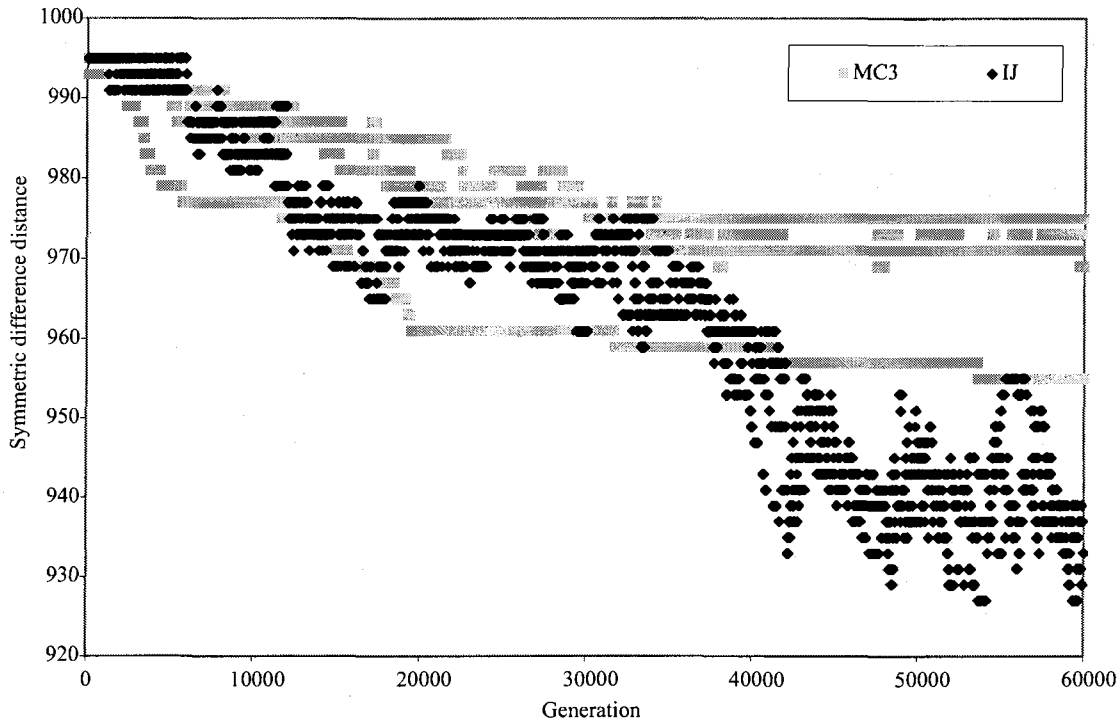


FIGURE III-5 DISTANCE FROM GENERATING TREES

Distance from first (5a) and second (5b) generating tree that contributed to the mixed data set. Gray squares are MCMC samples (every 1000 generations), black diamonds are IJ samples (every 1000 generations, jackknifed iterations omitted). See text for details.



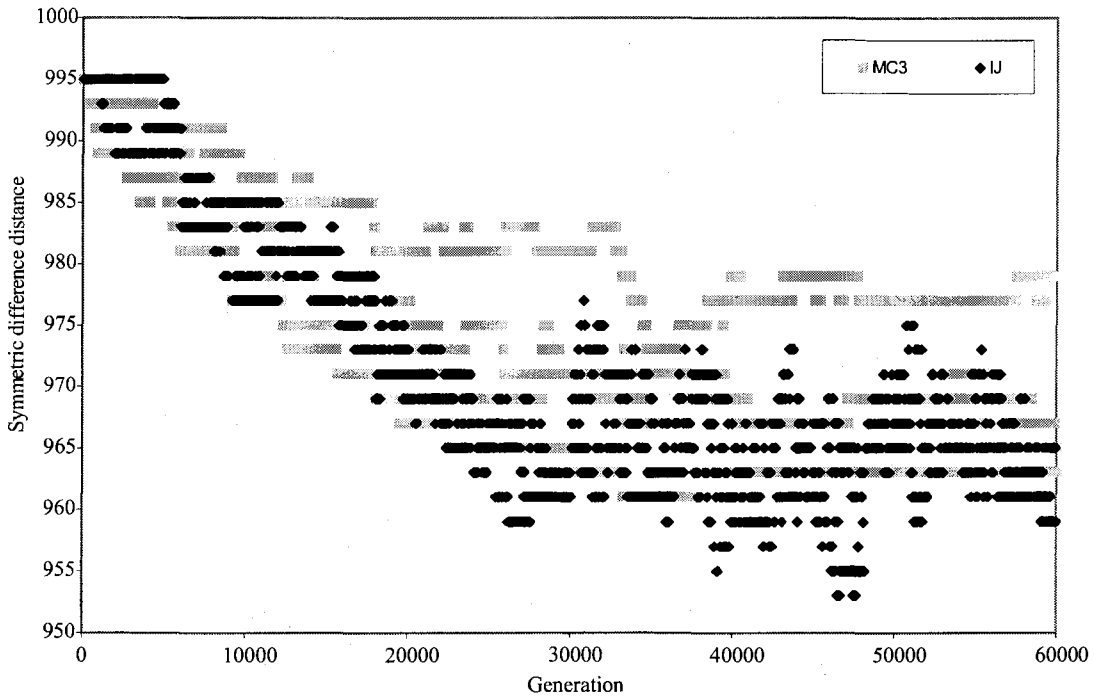
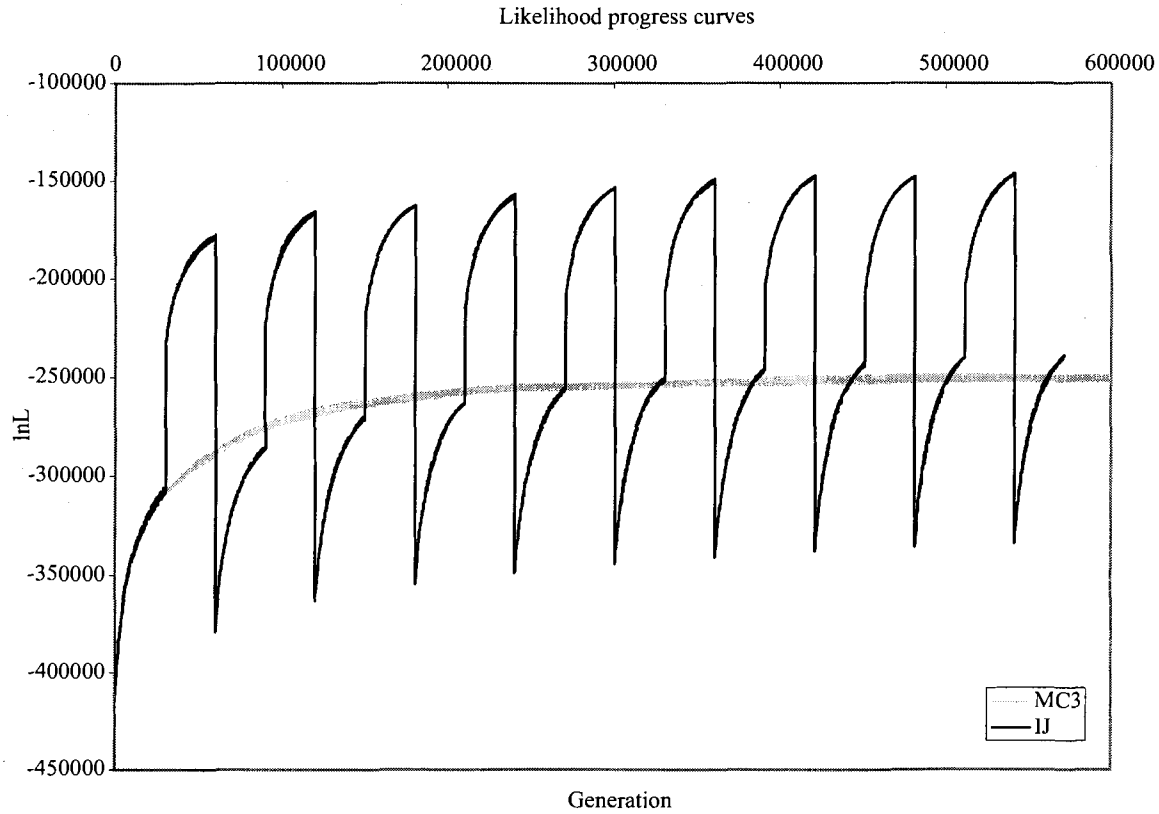


FIGURE III-6 LOG LIKELIHOOD TREE SCORES.

Log likelihood scores measured over a 50% jackknifed IJ run and a standard MCMC run for the simulated mixed data set. See text for details.



CHAPTER IV - RECONSTRUCTING DIVERGENCE TIMES FOR SUPERTREES: A MOLECULAR APPROACH[3]

Rutger A. Vos and Arne Ø. Mooers

[3] This chapter is published as: Vos, R. A., and A. Ø. Mooers. 2004. Reconstructing divergence times for supertrees: a molecular approach *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht. Reproduced with permission.

ABSTRACT

We present a formal approach to estimating divergence dates derived from aligned DNA sequence data on MRP supertrees, using a new supertree for the Primates as a case study. We selected 40 sequence data sets that conform under various models of sequence evolution to the molecular clock. Each of these data sets covers only a subset of the taxa on the supertree, and so composite date estimates were obtained by calibrating the data sets on common nodes and subsequently combining the estimates from different genes for the same node. The internal consistency of our estimates is high. The estimates presented here also fit well with those from Purvis' 1995 primate supertree, although estimates for deeper splits are progressively older.

Keywords: divergence times; fossils; maximum likelihood; molecular clock; Primates; supertree techniques

INTRODUCTION

Supertrees can be applied usefully to research beyond that of descriptive systematics (Bininda-Emonds *et al.*, 2002; Gittleman *et al.*, 2004), including comparative studies of character evolution (Gittleman *et al.*, 2004); studies of speciation, extinction, and diversification rates (Purvis *et al.*, 1995; Moore *et al.*, 2004); or establishing conservation priorities (e.g., based on the "evolutionary heritage" concept, the amount of independent evolutionary history embodied within a taxon; Mooers *et al.*, in press). These applications require phylogenies for which divergence dates, relative or absolute, are established. Although estimates of relative branch lengths from consensus techniques are possible (see Bryant *et al.*, 2004), the most widely used technique for the amalgamation of source trees, matrix representation with parsimony analysis (MRP; Baum, 1992; Ragan 1992), does not result in branch lengths that can be interpreted as a temporal dimension. Instead, divergence dates on supertrees are added afterwards, if at all. In some of the currently published supertrees, divergence dates were obtained through a combination of fossil dates, indirect estimates of sequence divergence by measuring branch lengths from published sources, and models for the expected age of clades given the number of taxa of that clade relative to its dated parent clade (Purvis, 1995; Bininda-Emonds *et al.*, 1999). In other studies (e.g., Wojciechowski *et al.*, 2000; Liu *et al.*, 2001; Jones *et al.*, 2002; Pisani *et al.*, 2002), no effort was made to establish divergence dates. In any case, objective and robust methods to reconstruct divergence dates for MRP supertrees directly from molecular data sets have yet to be established. Here, we will comment on the advantages and pitfalls of different techniques and data sources, and then discuss a molecular approach as applied to a new supertree of the order Primates.

FOSSILS AS TOOLS FOR CALIBRATION

If a fossil can be ascribed clearly to a clade, it can offer a minimum estimate of the age of that clade. The application of fossils in estimating divergence dates is twofold: a fossil date can not only be used to define the minimal age of a single node or clade (and its sister group) in a tree, but also to calibrate the absolute depths of other nodes in the same tree if the relative depths of these nodes have been inferred (e.g., from gene sequence data). This distinction is worth mentioning in the context of supertrees: relative node ages are unknown for MRP supertrees and fossils can supply information only in the former manner (i.e., as an indicator of the minimum age of clades and their sister groups) without recourse to added data. The paucity of fossil data is therefore an especially big problem in this type of supertree construction and subsequent dating.

The data on the ages of taxa provided by the fossil record has conflicted with molecular phylogenetic data on several occasions. A textbook example of such conflict is the initial identification of *Ramapithecus* as a 9–12 million year old hominid, constraining the split between humans and the (non-hominid) chimpanzees to be older than that. The subsequent reclassification of *Ramapithecus* as being more closely related to orangutans reconciled the fossil-constrained age of the hominids with the mounting molecular evidence of a more recent origin (Ridley, 1996). Clearly, a misidentified fossil leads to correlated errors for all the node depth calibrations based on it. The reliability and independence of fossil dates should therefore be evaluated critically, as stressed by, for example Lee (1999), who showed that recent molecular evidence for the earliest metazoan split (Xun, 1998) was calibrated on only two "fossil" dates — one of which

was actually obtained from the other "with an additional (molecular) layer of uncertainty introduced" (Lee, 1999:387).

The carnivore supertree (Bininda-Emonds *et al.*, 1999) is an example where fossils were used to derive the minimum age of sister groups: the time of first occurrence of either descendant lineage was used to date nodes. It is agreed generally that because fossils can be classified only once clade-defining morphological synapomorphies have arisen (Archibald, 1999), it is likely that the fossils of the earliest members of a clade are often overlooked as members of the clade (if these fossils have formed and were discovered at all). Thus, fossil dates will be too-young estimates of the age of clades. A famous example of this is the "Cambrian explosion" scenario, a hypothesized evolutionary burst (e.g., Gould, 1989; Lipps and Signor, 1992) that hinges on the assumption that sudden cladogenesis and trait evolution followed from the sudden appearance of most animal phyla in the Cambrian fossil record. Molecular studies, however, consistently support an extended period of Precambrian metazoan diversification (Bromham *et al.*, 1998; Bromham and Hendy 2000) along "ghost lineages" (Novacek and Wheeler, 1992; Fortey *et al.*, 1996), giving further evidence that fossils should not be considered as fixed ages of nodes, but rather as constraints on the minimum ages of nodes. However, despite the difficulties in working with fossils in terms of their rarity and their interpretation, the key attraction to fossils is that they are the only way, ultimately, that absolute ages of clades can be determined.

Relative divergence dates inferred from molecular phylogenies

DNA sequence data can provide information on when species have diverged, not only on the branching order that can be inferred from the phylogenetic signal they

provide, but also on the relative timing of these branching events. For the latter to work, the locus under study must conform to the "molecular clock" (Zuckercandl and Pauling, 1965), which in practice means that substitution rates must be constant along all lineages, resulting in an ultrametric tree (i.e., a tree with the same root-to-tip path length for all lineages). Whether or not a particular locus conforms to the molecular clock can be tested by comparing the likelihood (Felsenstein, 1981) of the optimal topology under unconstrained rates to the likelihood of the same tree constrained to be ultrametric. The ultrametric tree will have a worse score, but, if it is not significantly worse, the locus is considered to conform to the molecular clock hypothesis.

Clocklike loci are a useful source of information from which divergence dates for supertrees can be obtained. However, the MRP supertree technique does not allow for branch length information to be encoded in such a way that the resulting supertree reproduces meaningful divergence date estimates. Therefore, in earlier MRP supertree studies, molecular data on divergence times was used indirectly (Purvis, 1995; Bininda-Emonds *et al.*, 1999) by rescaling previously published molecular phylogenies, calibrating them subsequently using fossil data, and then sticking the divergence dates so obtained on the supertree. This approach has two drawbacks. First, the rescaling process (as described in Purvis, 1995) is essentially a method by which source phylogenies are "ultrametricized" without recourse to the underlying sequence data. It is therefore not certain whether or not the particular locus actually conforms to the molecular clock. Second, the source trees sometimes do not match the topology of the supertree, rendering the source tree in whole or in part unusable. Given these drawbacks, we argue that using

sequence data directly is an approach that warrants further research, a case study of which is discussed in this chapter.

Obtaining composite estimates of divergence dates from sequence data

Relative branch lengths from a set of congruent phylogenies that each cover a subset of taxa usually cannot be combined to derive the branch lengths for the phylogeny that covers the bigger set from which the subsets were drawn. However, the depths of nodes in the set of congruent phylogenies can be combined. For instance, Figure IV-1a shows a topology for which the divergence dates are unknown. The four trees in Figure IV-1b each cover a subset of the taxa of the tree in Figure IV-1a, and are congruent with the topology of that tree. The branch lengths for these ultrametric trees might have been derived from disparate data sources, such as different genes that conform to the molecular clock hypothesis. By calibrating these trees on a shared node — such as node 2 for trees II–IV in this example — the node depths of these trees can be combined to obtain the branch lengths for the topology of Figure IV-1a (as shown to the right in Figure IV-1c). From the example in Figure IV-1, it is evident that this method can be used only to combine divergent dates from multiple sources that share at least one node. However, this is not the only consideration that needs to be taken into account in choosing calibration points.

The location of the calibration point relative to the other nodes in the source trees has an effect on how variation in the estimates is distributed over the tree. Figure IV-2 illustrates this via a simulation. Branch lengths on 1000 ultrametric and fully-unbalanced (i.e., comb-like) 32-taxon trees were simulated based on a pure birth model for clade growth (Harding, 1971). This is a common process for generating divergence times on

trees, with the useful property that the waiting times between successive branching events are drawn from a negative exponential distribution with parameter n, where n is the number of extant taxa at any time (Nee *et al.*, 1992; Nee, 2001). Relative waiting times (and so relative branch lengths) can therefore be simulated simply as $t = -\ln(p) / n$, where p is a uniformly distributed random number between 0 and 1 that represents the uniform distribution of probabilities. Although the trees so generated are all the same size and shape, they differ in their total depth as a result of the stochastic nature of the birth process.

Figure IV-2a–c depict different calibration scenarios for these simulated trees. In Figure IV-2a, all 1000 trees were calibrated on the root, forcing them all to have the same total depth. The graph plots the median depth over the sets of equivalent estimated nodes (i.e., the most recent split, the second-most, the third-most, through to the root) as the x-axis and the coefficient of variation over each of these sets as the y-axis. The data point with the largest depth (i.e., the root) had a coefficient of variation of zero because it was used as the calibration point. As we moved away from the calibration point (i.e., leftward), the coefficient of variation increased because of the cumulative effect of the randomness propagating through the tree.

Figure IV-2b shows how the coefficient of variation behaves when a node of intermediate depth was chosen as calibration point. Once again, the coefficient of variation increased as we moved away from the calibration point, both when we moved nearer to the tips or to the root. However, the mean coefficient of variation over all nodes was lower (here, 0.309 versus 0.368 when calibrated on the root). Figure IV-2c shows the behaviour when a recent node was used as calibration point: the mean coefficient of

variation over all nodes was the highest of all scenarios (2.324). This is probably because of the Central Limit Theorem: the depth of the first split, unlike all that follow, is not the result of the sum of a series of draws from the exponential distribution, but rather of a single draw, and so the variation over a set of such nodes is accordingly higher than that over any set of deeper nodes. Thus, constraining a set of these first, more variable, splits to the same depth will increase the variation over each set of deeper nodes.

In comb-like trees, all nodes are ordered consistently and linearly, and so the trees in our simulation provide a highly simplified and somewhat extreme example of the effect of choosing a single calibration point on the overall variation over all other nodes. Nevertheless, we expect that the same effect will hold for real datasets, albeit to a lesser extent because most real trees are not fully unbalanced.

Because it is desirable to choose a calibration point that minimizes the total variation over node depth estimates, the best choice would be to choose an intermediate node for calibration. However, even if the variation over different estimates is so minimized, it is still likely to be high as a result of outliers caused by, for instance, 1) saturated genes reducing the estimated depth for deeper nodes or 2) genes that give highly discrepant estimates for other reasons such as different strengths and modes of selection along different lineages. In earlier studies where divergence dates were combined in supertrees (e.g., Purvis, 1995), the influence of such outliers was minimized by taking the median instead of the mean over the set of estimates. We do the same here.

From the simulations, it is evident that overall variation can be reduced by choosing an optimal calibration point. However, even if one were to choose the node that is located optimally within the topology of the tree, stochasticity will still propagate

through the tree such that nodes located away from the calibration point will be highly variable. By using multiple calibration points located in disparate regions of the tree, we can minimize this effect. This approach has the added merit of including more previously known information on divergence dates.

Consider Figure IV-1 again. In this example, the trees were calibrated on the shared node 2. All prior information on the other divergence dates is thus disregarded when obviously we should strive to incorporate all available, robust, information in the estimates. We will do this by averaging all divergence date estimates for a given node across all different calibration points for which prior information is available. For instance, if node 1 in Figure IV-1 would also be used as a calibration point, we would get two data points for node 2: one where it was used as a calibration point as shown in Figure IV-1c, and one from tree II calibrated on node 1. Similarly, we would get two estimates for node 3 (one from the median of the estimates obtained by calibrating trees III and IV on node 2, and one from tree I calibrated on node 1) as well as for node 1 (one obtained by calibrating tree II on node 2 and one where it is used as a calibration point for trees I and II). We then average over the data points for each the respective nodes and incorporate the results into the supertree. We apply this method below.

METHODS

PHYLOGENY CONSTRUCTION

The primate phylogeny we used in this study will be presented in full in a companion article (Vos and Mooers, in prep.), and so we offer only the briefest outline here. We collected 217 source trees from 126 articles published after 1993 and combined these with the data from the primate supertree of Purvis (1995). We then combined all these datasets into one large MRP matrix using RadCon (Thorley and Page, 2000) and used the parsimony ratchet (Nixon, 1999) strategy as implemented in the program PAUPRat (http://viceroy.eeb.uconn.edu/paupratweb/pauprat.htm) to search tree space under various models of character state change. Finally, we constructed majority-rule and strict consensus trees over each of the resulting sets of unique optimal trees.

MOLECULAR DATA COLLECTION

To collect suitable candidate genes for the inference of relative divergence dates we downloaded the Primates section of the NCBI-GenBank Flat File Release 132.0 from ftp.ncbi.nlm.nih.gov. We indexed this data set using the standalone BLAST tool formatdb and performed keyword frequency ("grep") searches to collect genes that were sequenced over a broad taxonomic range. We refined these results using BLAST (Altschul *et al.*, 1990) searches. This yielded 55 candidate genes. We aligned these sequence data sets using ClustalW's default settings and method (Thompson *et al.*, 1994) and subsequently by hand. We then ran ModelTEST (Posada and Crandall, 1998) on each data set using the likelihood-ratio test statistic $\delta = -2 \log \Lambda$ to identify the appropriate nucleotide substitution model from a nested set.

Subsequently, we tested whether the molecular clock could be rejected using the same statistical approach, but with a liberal alpha for rejection of 0.001. We chose this alpha level for two reasons. First, given that the likelihood-ratio test for rate constancy is a test of significance, the usual alpha level of 0.05 will reject the clock by chance alone once in every twenty tests on average, even if all loci behave in a clocklike manner (i.e., a Type I error). Lowering the alpha level reduced this risk and so served as a correction for multiple comparisons. Second, lowering the alpha level to 0.001 allowed us to include data sets that evidently deviate somewhat from rate constancy such that they would have been rejected under the more commonly used level of 0.05.

Because this approach by itself yielded too few data sets, we developed a program that iteratively prunes from the non-clocklike data sets those taxa that are the most divergent from the mean root-to-tip path length, and subsequently tests whether the data set then conforms to the molecular clock. The routine stops once $p > 0.001$. Essentially, this program removes those lineages from a data set within which substitution rates have increased or decreased significantly relative to the average of that data set. Data sets where the program stopped when three taxa remained were discarded because conforming to the molecular clock with so few taxa is essentially meaningless.

Using this approach, which could be described as "gene shopping" followed by "taxon shopping", 40 loci conformed to the molecular clock. The loci analyzed in this study are listed in Table IV-1; those that conform to the molecular clock and that we used to obtain divergence dates are indicated by an asterisk.

We labeled each node in the topology of the supertree by appending a serial number — and, to remain compliant with the NEXUS format (Maddison *et al.*, 1997), the word "node" — to each closing bracket of the tree description. The result is similar to the labeling on the tree in Figure IV-1a. For each aligned clock-like sequence data set, we then pruned all taxa that were absent in that data set from the supertree so as to obtain constraint trees congruent with the consensus supertree, while keeping track of the initial node-labeling scheme. This resulted in a set of trees with labeled nodes like those shown in Figure IV-1b. The labeling and pruning was done using Perl scripts, which are available from the authors upon request. We then estimated the branch lengths on these constraint trees under the appropriate models using PAUP* (Swofford, 2002). We calculated relative node depths from these branch lengths using the ape package (http://stat.ethz.ch/R-CRAN/doc/packages/ape.pdf) for the R program. The routine that calculates these depths visits all labeled nodes and, for each, calculates the path length from that focal node to the tips and writes it to a table. Because the routine does not take all possible paths into consideration, it gives meaningful results only for ultrametric (i.e., clocklike) trees. We then combined the results from the individual genes into a larger table to calibrate these multiple loci on shared nodes. We surveyed the recent literature for estimates of the timing of major, uncontested splits in the evolutionary history of the primates that could function as calibration points (e.g., Gingerich and Uhen, 1994; Adachi and Hasegawa, 1995; Adachi and Hasegawa, 1996; Arnason *et al.*, 1996a, b, 1998, 2000; Easteal and Herbert, 1997; Porter *et al.*, 1997; Yoder, 1997; Goodman *et al.*, 1998; Kumar and Hedges, 1998; Stauffer *et al.*, 2001; Nei and Glazko, 2002).

RESULTS

The majority-rule consensus tree that we dated was based on a search using irreversible character-state changes, and had a resolution of 0.917 (over 15 242 unique optimal trees), and a consistency index of 0.82.

Figure IV-3 presents the relationship between the median depth of a set of equivalent estimated nodes and coefficient of variation over that set under three calibration scenarios. Figure IV-3a depicts the variation over the divergence dates if the depths were calibrated on the split between *Homo* and *Pan*, which is a recent split in the context of primate phylogeny. The total variation was highest under this scenario (mean coefficient of variation, CV = 0.622). Variation was lowered when all node depths were calibrated on the root (mean CV = 0.513; Figure IV-3c). The coefficient of variation was lowest when the split between the Colobinae and Cercopithecinae was used for calibration (mean CV = 0.345; Figure IV-3b). The results shown in Figure IV-3 demonstrate that the actual data behaved as we assumed from the results of our simulations: the lowest overall variation was obtained by calibrating on a node of intermediate depth, whereas recent nodes used as calibration points led to the highest variation. Note that the comparison is not exact for several reasons: first, different numbers of genes were common to each calibration; second, the topology of the supertree is not comb-like; and finally, the model used in our simulations was a simplified approximation of the actual process of clade growth (of which a molecular phylogeny is again an approximation).

The depths of the calibration points used in Figure IV-3 were obtained by taking the median over the estimates we found in a search through the recent literature (Table

IV-2). These previously published dates were obtained through a variety of methods and data sources: from fossils (Goodman *et al.*, 1998); from a coalescence model for species diversity (Gingerich and Uhen, 1994); from maximum likelihood estimates using mtDNA calibrated on divergences outside the order (Arnason *et al.*, 1996a; Arnason *et al.*, 1998), inside the order (Adachi and Hasegawa, 1995, 1996; Yoder, 1997), or calibrated on geological data (Arnason *et al.*, 1996b; Stauffer *et al.*, 2001) or using the method of Li *et al.* (1987; Arnason *et al.*, 2000); from nuclear sequences calibrated on nodes outside (Easteal and Herbert, 1997; Kumar and Hedges, 1998) or inside the order (Porter *et al.*, 1997); from amino acid sequences calibrated inside and outside the order (Nei and Glazko, 2002); and using the mixed fossil and rescaled phylogenies technique outlined earlier (Purvis, 1995). The estimates, all in millions of years ago (MYA), are listed in Table IV-2. We calibrated our data on the median values over these estimates and averaged over the nine resulting sets of estimates (i.e., one for each calibration point), some of the results of which are listed in the bottom row of Table IV-2. Figure IV-4 presents date estimates for the same nine splits we found using our method by calibrating trees first on each of these published estimates in turn and then averaging the results.

DISCUSSION

The divergence dates estimated using the method described here generally fit well with previously published estimates from different sources (see the examples in Table IV-2). The correlation between dates estimated here and those for the equivalent nodes in the only other large-scale study (that of Purvis, 1995) is strong (Figure IV-5). Note that the topology of our supertree is different from that of Purvis in this comparison, and so we compared only those nodes that were unambiguously equivalent (the subtrees descending from these nodes could be different, however). The comparison is therefore not exact, and any differences observed could still be a result of different methods, different topologies, or both.

Compared with the date estimates in Purvis (1995), the estimates presented in this paper were increasingly older with their depth in the tree. We suspect that this is a result of a trend in primate phylogenetics that can be ascribed to both newly discovered, older fossil finds as well as the use of more sophisticated models of sequence evolution in more recent studies.

One potential weakness of our approach is that we have not been able to cover every node in the supertree with the currently available data. On the most resolved topology, 55% of the nodes had date estimates, with all the missing data concentrated around recent nodes in rarely studied clades. Although the amount of sequence data in public databases is growing rapidly, some way of incorporating more non-clocklike loci would seem desirable, perhaps using methods akin to those pioneered by Sanderson (1997, 2002). Even so, missing data points will probably remain in our tree that would

have to be interpolated based on models for clade growth such as those used in previous supertree studies (Purvis, 1995; Bininda-Emonds *et al.*, 1999).

More comparisons of our approach with that of Bininda-Emonds *et al.* (1999) will be necessary, as will further exploration of the relative power of this hybrid MRP + model-based method and traditional tree-building algorithms that consider the genetic data directly, incorporate multiple genes and multiple models, and, most dauntingly, mixed clock and nonclock scenarios for different data partitions. This, however, is for the future.

ACKNOWLEDGEMENTS

We would like to thank Andy Purvis for kindly providing the source data used for his primate supertree research; Vincent Nijman and Eva Chrostowski for assistance with data collection; the members of FAB-lab and Eirikur Palsson for valuable input; Olaf Bininda-Emonds for inviting us to contribute, for his patience, and for his keen editing; and Paul-Michael Agapow and Kate Jones for in-depth reviews of the manuscript.

REFERENCES

Adachi, J. and Hasegawa, M. 1995. Improved dating of the human/chimpanzee separation in the mitochondrial DNA tree: heterogeneity among amino acid sites. Journal of Molecular Evolution 40:622–628.

Adachi, J. and Hasegawa, M. 1996. Tempo and mode of synonymous substitutions in mitochondrial DNA of Primates. Molecular Biology and Evolution 13:200–208.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. 1990. Basic local alignment search tool. Journal of Molecular Biology 215:403–410.

Archibald, J. D. 1999. Molecular dates and the mammalian radiation. Trends in Ecology and Evolution 14:278–278.

Arnason, U., Gullberg, A., Burguete, A. S., and Janke, A. 2000. Molecular estimates of primate divergences and new hypotheses for primate dispersal and the origin of modern humans. Hereditas 133:217–228.

Arnason, U., Gullberg, A., and Janke, A. 1998. Molecular timing of primate divergences as estimated by two nonprimate calibration points. Journal of Molecular Evolution 47:718–727.

Arnason, U., Gullberg, A., Janke, A., and Xu, X. 1996a. Pattern and timing of evolutionary divergences among hominoids based on analyses of complete mtDNAs. Journal of Molecular Evolution 43:650–661.

Arnason, U., Xu, X. F., Gullberg, A., and Graur, D. 1996b. The "Phoca standard": An external molecular reference for calibrating recent evolutionary divergences. Journal of Molecular Evolution 43:41–45.

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

Bininda-Emonds, O. R. P., Gittleman, J. L. and Purvis, A. 1999. Building large trees by combining phylogenetic information: a complete phylogeny of the extant Carnivora (Mammalia). Biological Reviews 74:143–175.

Bininda-Emonds, O. R. P., Gittleman, J. L., and Steel, M. A. 2002. The (super)tree of life: procedures, problems and prospects. Annual Review of Ecology and Systematics 33:265–289.

Bromham, L. D., Rambaut, A., Fortey, R., Cooper, A. and Penny, D. 1998. Testing the Cambrian explosion hypothesis by using a molecular dating technique. Proceedings of the National Academy of Sciences of the United States of America 95:12386–12389

Bromham, L. D. and Hendy, M. D. 2000. Can fast early rates reconcile molecular dates with the Cambrian explosions? Proceedings of the Royal Society of London B 267:1041–1047

Bryant, D., Semple, C., and Steel, M. 2004. Supertree methods for ancestral divergence dates and other applications. In O. R. P. Bininda-Emonds (ed). Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, volume 3, pp. 129-150. Kluwer Academic, Dordrecht, the Netherlands.

Colless, D. H. 1980. Congruence between morphometric and allozyme data for Menidia species: a reappraisal. Systematic Zoology 29:288–299.

Easteal, S. and Herbert, G. 1997. Molecular evidence from the nuclear genome for the time frame of human evolution. Journal of Molecular Evolution 44(Suppl. 1):S121–S132.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution 17:368–376.

Fortey, R. A., Briggs, D. E. G., and Wills, M. A. 1996. The Cambrian evolutionary "explosion": decoupling cladogenesis from morphological disparity. Biological Journal of the Linnean Society 57:13–33.

Gingerich, P. D. and Uhen, M. D. 1994. Time of origin of primates. Journal of Human Evolution 27:443–445.

Gittleman, J. L., Jones, K. E., and Price, S. A. 2004. Supertrees: using complete phylogenies in comparative biology. In O. R. P. Bininda-Emonds (ed.), Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, volume 3, pp. 439–460. Kluwer Academic, Dordrecht, the Netherlands.

Goodman, M., Porter, C. A., Czelusniak, J., Page, S. L., Schneider, H., Shoshani, J., Gunnell, G. F., and Groves, C. 1998. Toward a phylogenetic classification of primates based on DNA evidence complemented by fossil evidence. Molecular Phylogenetics and Evolution 9:585–598.

Gould, S. J. 1989. Wonderful Life. Norton, New York.

Harding, E. F. 1971. The probabilities of rooted tree shapes generated by random bifurcation. Advanced Applied Probability 3:44–77.

Hasegawa, M., Kishino, H., and Yano, T.-A. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. Journal of Molecular Evolution 22:160–174.

Jones, K. E., Purvis, A., MacLarnon, A., Bininda-Emonds, O. R. P., and Simmons, N. B. 2002. A phylogenetic supertree of the bats (Mammalia: Chiroptera). Biological Reviews 77:223–259.

Kimura, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. Journal of Molecular Evolution 16:111–120.

Kluge, A. and Farris, S. 1969. Quantitative phyletics and the evolution of anurans. Systematic Zoology 18:1–32.

Kumar, S. and Hedges, S. B. 1998. A molecular timescale for vertebrate evolution. Nature 392:917–920.

Lee, M. S. Y. 1999. Molecular clock calibrations and metazoan divergence dates. Journal of Molecular Evolution 49:385–391.

Li W.-H., Wolfe, K. H., Soudis, J., and Sharp, P. 1987. Reconstruction of phylogenetic trees and estimation of divergence times under nonconstant rates of evolution. Cold Spring Harbor Symposia on Quantitative Biology 52:847–856.

Lipps, J. H. and Signor, P. W. 1992. Origin and Early Evolution of Metazoa. Plenum, New York.

Liu, F.-G. R., Miyamoto, M. M., Freire, N. P., Ong, P. Q., Tennant, M. R., Young, T. S., and Gugel, K. F. 2001. Molecular and morphological supertrees for eutherian (placental) mammals. Science 291:1786–1789.

Maddison, D. R., Swofford, D. L., and Maddison, W. P. 1997. NEXUS: an extensible file format for systematic information. Systematic Biology 46:590–621.

Mooers, A.O., Heard, S. B., and E. Chrostowski, E. 2005. Evolutionary heritage as a metric for conservation. In A. Purvis, T. L. Brooks, and J. L. Gittleman (eds), Phylogeny and Conservation. Oxford University Press, Oxford.

Moore, B. R., Chan, K. M. A., and Donoghue, M. J. 2004. Detecting diversification rate variation in supertrees. In O. R. P. Bininda-Emonds (ed.), Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, volume 3, pp. 487-533. Kluwer Academic, Dordrecht, the Netherlands.

Nee, S. 2001. Inferring speciation rates from phylogenies. Evolution 55:661–668.

Nee, S., Mooers, A. Ø., and Harvey, P. H. 1992. Tempo and mode of evolution revealed from molecular phylogenies. Proceedings of the National Academy of Sciences of the United States of America 89:8322–8326.

Nei, M. and Glazko, G. V. 2002. Estimation of divergence times for a few mammalian and several primate species. Journal of Heredity 93:157–164.

Nixon, K. 1999. The parsimony ratchet, a new method for rapid parsimony analysis. Cladistics 15:407–414.

Novacek M. J. and Wheeler, Q. D. 1992. Introduction: extinct taxa. In: Novacek M. J. and Q. D. Wheeler (eds), Extinction and Phylogeny: New York: Columbia University Press, 1–16.

Pisani, D., Yates, A. M., Langer, M. C., and Benton, M. J. 2002. A genus-level supertree of the Dinosauria. Proceedings of the Royal Society of London B 269:915–921.

Porter, C. A., Page, S. L., Czelusniak, J., Schneider, H., Schneider, M. P. C., Sampaio, I., and Goodman, M. 1997. Phylogeny and evolution of selected primates as determined by sequences of the □-globin locus and 5' flanking regions. International Journal of Primatology 18:261–295.

Posada, D. and Crandall, K. A. 1998. MODELTEST: testing the model of DNA substitution. Bioinformatics 14:817–818.

Purvis, A. 1995. A composite estimate of primate phylogeny. Philosophical Transactions of the Royal Society of London B 348:405–421.

Purvis, A., Nee, S. and Harvey, P. H. 1995. Macroevolutionary Inferences from Primate Phylogeny. Proceedings of the Royal Society of London B 260:329–333.

Ragan, M. A. 1992. Phylogenetic inference based on matrix representation of trees. Molecular Phylogenetics and Evolution 1:53–58.

Ridley, M. 1996. Evolution, 2nd edition. Blackwell Science, Inc., Cambridge, Massachusetts.

Rodríguez, F., Oliver, J. L., Marín, A., and Medina, J. R. 1990. The general stochastic model of nucleotide substitution. Journal of Theoretic Biology 142:485–501.

Salamin, N., Hodkinson, T. R., and Savolainen, V. 2002. Building supertrees: an empirical assessment using the grass family (Poaceae). Systematic Biology 51:112–126.

Sanderson, M. J. 1997. A nonparametric approach to estimating divergence times in the absence of rate constancy. Molecular Biology and Evolution 14:1218–1231.

Sanderson, M. J. 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. Molecular Biology and Evolution 19:101–109

Stauffer, R. L., Walker, A., Ryder, O. A., Lyons-Weiler, M., and Hedges, S. B. 2001. Human and ape molecular clocks and constraints on paleontological hypotheses. Journal of Heredity 92:469–474.

Swofford, D. L. 2002. PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4. Sinauer, Sunderland, Massachusetts.

Thompson, J. D., Higgins, D. G., and Gibson T. J. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Research 22:4673–4680.

Thorley, J. L. and Page, R. D. M. 2000. RadCon: phylogenetic tree comparison and consensus. Bioinformatics 16:486–487.

Vos, R. A. and A. Ø. Mooers. in prep. A new dated supertree of the primates. for Syst. Biol.

Wilson, D. E. and Reeder, D. M. (eds). 1993. Mammal Species of the World. Smithsonian Institution Press, Washington DC.

Wojciechowski, M. F., Sanderson, M. J., Steel, K. P., and Liston, A. 2000. Molecular phylogeny of the "temperate herbaceous tribes" of papilionoid legumes: a supertree approach. In P. Herendeen and A. Bruneau (eds), Advances in Legume Systematics 9:277–298. Royal Botanic Garden, Kew.

Xun, G. 1998. Early metazoan divergence was about 830 million years ago. Journal of Molecular Evolution 47:369–371.

Yang, Z., Goldman, N., and Friday, A. 1994. Comparison of models for nucleotide substitution used in maximum likelihood phylogenetic estimation. Molecular Biology and Evolution 11:316–324.

Yang, Z. 1996. Among-site variation and its impact on phylogenetic analyses. Trends in Ecology and Evolution 11:367–371.

Yoder, A. D. 1997. Back to the future: a synthesis of strepsirrhine systematics. Evolutionary Anthropology: Issues, News, and Reviews 6:11–22.

Zuckercandl, E. and Pauling, L. 1965. Evolutionary divergence and convergence in proteins. In V. Bryson and H. J. Vogel (eds), pp. 97–165 in Evolving Genes and Proteins. Academic Press, New York.

TABLE IV-1

LOCI USED IN THIS STUDY

The abbreviated models are the following: HKY85: Hasegawa, Kishino, Yano (Hasegawa *et al.*, 1985); K80: Kimura two-parameter (Kimura, 1980); GTR: General Time Reversible (Rodríguez *et al.*, 1990; Yang *et al.*, 1994); +Γ: variation in rates among sites modeled using a gamma distribution (Yang, 1996); +I; a proportion of sites modeled as invariant (Hasegawa *et al.*, 1985). The number of taxa after pruning (see text) is given.

Gene	Model	Clock test p-value	No. of taxa
alpha-1,3-Galactosyltransferase	GTR+I	1.09752 x 10-23	19
ATP6	GTR+G+I	3.72376 x 10-10	17
ATP7A	HKY85+G	0.13700341*	7
ATP8	GTR+G+I	3.06905 x 10-10	17
BRCA1	HKY85+G	0.01145129*	7
Calmodulin	HKY85	0.815719539*	6
CCR5	K80+G+I	0.00046635	67
CD4	GTR+G	0.00189824*	22
COII	GTR+G+I	4.56 x 10-8	57
CXCR4	HKY85+G+I	8.91 x 10-5	42
DRD4	HKY85+G	0.07035867*	14
FUT1	HKY85+G	7.3035 x 10-149	32
Gamma1 globin	HKY85+G	0.0093968*	13
G6PD	HKY85+G	0.00121083*	23
IL-2	HKY85	0.92550696*	11
IL-3	GTR+I	0.020268243*	4
IL-4	GTR	0.13912934*	8
IL-6	HKY85	9.88 x 10-14	8
IL-10	HKY85	0.16400763*	9
IL-16	GTR+I	0.08135042*	7
Interferon gamma	HKY85+G	0.19943861*	13
IRBP (intron 1)	K80+G	0.00010886	37
IRBP (partial cds)	HKY85+G	0.01551987*	23
LZM	HKY85+G	0.000427075	17
nd1	GTR+G+I	0.36658552*	12
ND2	GTR+G+I	0.00033689	13
ND3	GTR+G+I	0.13023191*	36
ND4L	GTR+G+I	0.25098333*	45
ND5	GTR+G+I	0.00024931	27
ND6	GTR+G+I	0.26791855*	12
NRAMP1	HKY85	0.13333399*	14
PLCB4	GTR	0.85023707*	7
PNOC	GTR+G	0.28885825*	7

Gene	Model	Clock test p-value	No. of taxa
SRY	HKY85+G	0.01145427*	59
TSPY	HKY85+G	0.00116896*	41
tRNA-ala	GTR+G	0.101676857*	10
tRNA-arg	HKY85+G+1	0.1082015*	36
tRNA-asn	HKY85+G	0.000246114	10
tRNA-asp	HKY85+G	0.054571469*	10
tRNA-cys	GTR+G	0.597145544*	10
tRNA-gln	HKY85+G	0.406863018*	9
tRNA-glu	GTR+G	0.006563358*	10
tRNA-gly	HKY85+G	0.005765017*	30
tRNA-ile	GTR+1	0.003574332*	10
tRNA-lys	HKY85+G	0.007495502*	10
tRNA-met	GTR+I	0.552294012*	10
tRNA-phe	HKY85	4.28761 x 10-9	12
tRNA-pro	GTR+G	0.22147066*	11
tRNA-thr	GTR+I	0.010557853*	12
tRNA-trp	HKY85+G	7.52579 x 10-6	10
tRNA-tyr	GTR+G	0.012303697*	10
tRNA-val	HKY85+G	0.048727903*	26
ZFX	HKY85+G+I	0.0130834*	18
ZFY	GTR+G	0.00138935*	13
vWF	HKY85+G	0.031415638*	17

TABLE IV-2 RECENT ESTIMATES OF MAJOR PRIMATE SPLITS

1 = Apes-Old World monkeys; 2 = *Homo-Pan*; 3 = (*Homo*, *Pan*)-*Gorilla*; 4 = ((*Homo*, *Pan*),*Gorilla*)-*Pongo*; 5 = Great apes-Gibbons; 6 = Old World Monkeys-New World Monkeys; 7 = Root; 8 = Lemurs-Lorisiforms; 9= Colobinae-Cercopithecinae. All ages are in millions of years ago.

Source	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Nei and Glazko (2002)	23	6	7			33			
Stauffer *et al.* (2001)	23	5.4	6.4	11	15				
Gingerich and Uhen (1994)							63		
Yoder (1997)								54	
Arnason *et al.* (1998)	50					60	80		30
Porter *et al.* (1997)	25								
Goodman *et al.* (1998)						38			
Adachi and Hasegawa (1995)		4		16					
Easteal and Herbert (1997)				8.5					
Kumar and Hedges (1998)	23.3	5.5	6.7	8.2	14.6	47.6			
Arnason *et al.* (1996b)		6.1							
Adachi and Hasegawa (1996)		4.3							
Arnason *et al.* (2000)		13	16	30	35	70			
Arnason *et al.* (1996a)		10.4	14.2	19.2	32.4				
Purvis (1995)	27.5	7	8.3	14.5	18.2	40.5	57.5	45.1	14.4
Median of published studies	24.1	6	7.6	14.5	18.2	44	63	49.5	22.2
Present estimates	32.8	5.9	6.3	15.2	18.8	49.8	77.5	51.6	16.8

FIGURE IV-1 COMBINING AND CALIBRATING DIVERGENCE DATES.

a) Hypothetical MRP supertree topology, for which relative branch lengths and labeled node depths are undefined. b) Aligned sequence data sets that conform to the molecular clock when fitted to the topology of the supertree. Node labels correspond with those in (a). c) Sequence data sets II, III and IV are calibrated on their shared node 2. Based on these combined data sets, the depths for all three nodes can be reconstructed in the composite estimate. Because there are two data points for node 3, there is a range (hatched area) from which the median is selected for the composite estimate.

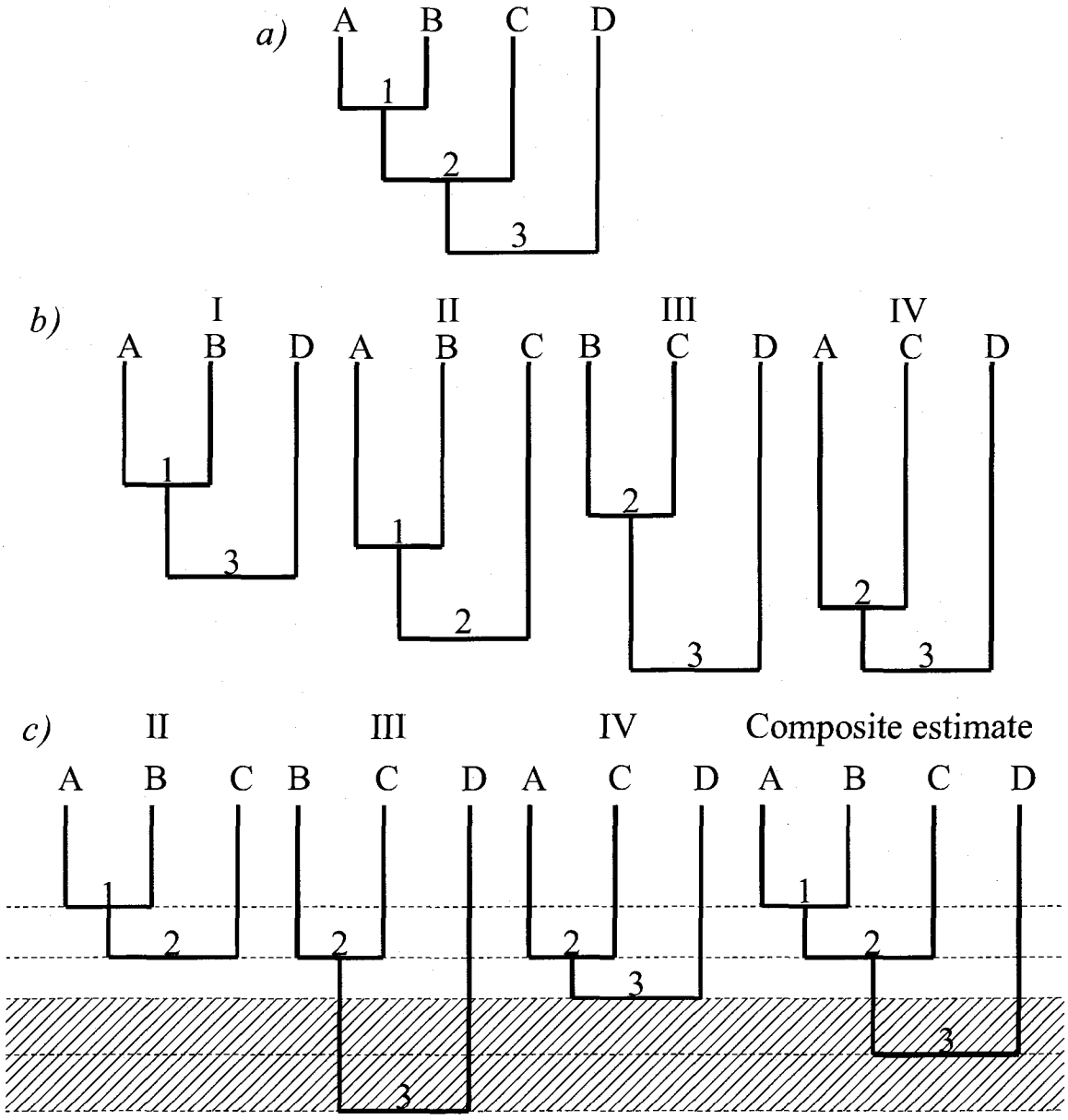


FIGURE IV-2 SIMULATED CALIBRATION SCENARIOS.

a) calibration on the root, b) calibration on an intermediate node, and c) calibration on a recent node. Each data point represents a set of equivalent nodes over 1000 comb-like trees. For instance, the rightmost point represents a set of a thousand roots, whereas the leftmost point represents the set of nodes that splits the most recent pair of sister species. Median depth over each set is plotted on the horizontal axis such that values of 0 and 1 correspond with the tips and the root, respectively. On the vertical axis, the coefficient of variation over each set is given, give the following calibration scenarios An example of a 32-taxon ultrametric tree with branch lengths simulated under a Yule model, such as the trees used in these calibration scenarios, is given in (d). Its orientation is identical to the data points in (a)–(c) (i.e., with the oldest nodes on the right and the newest on the left).

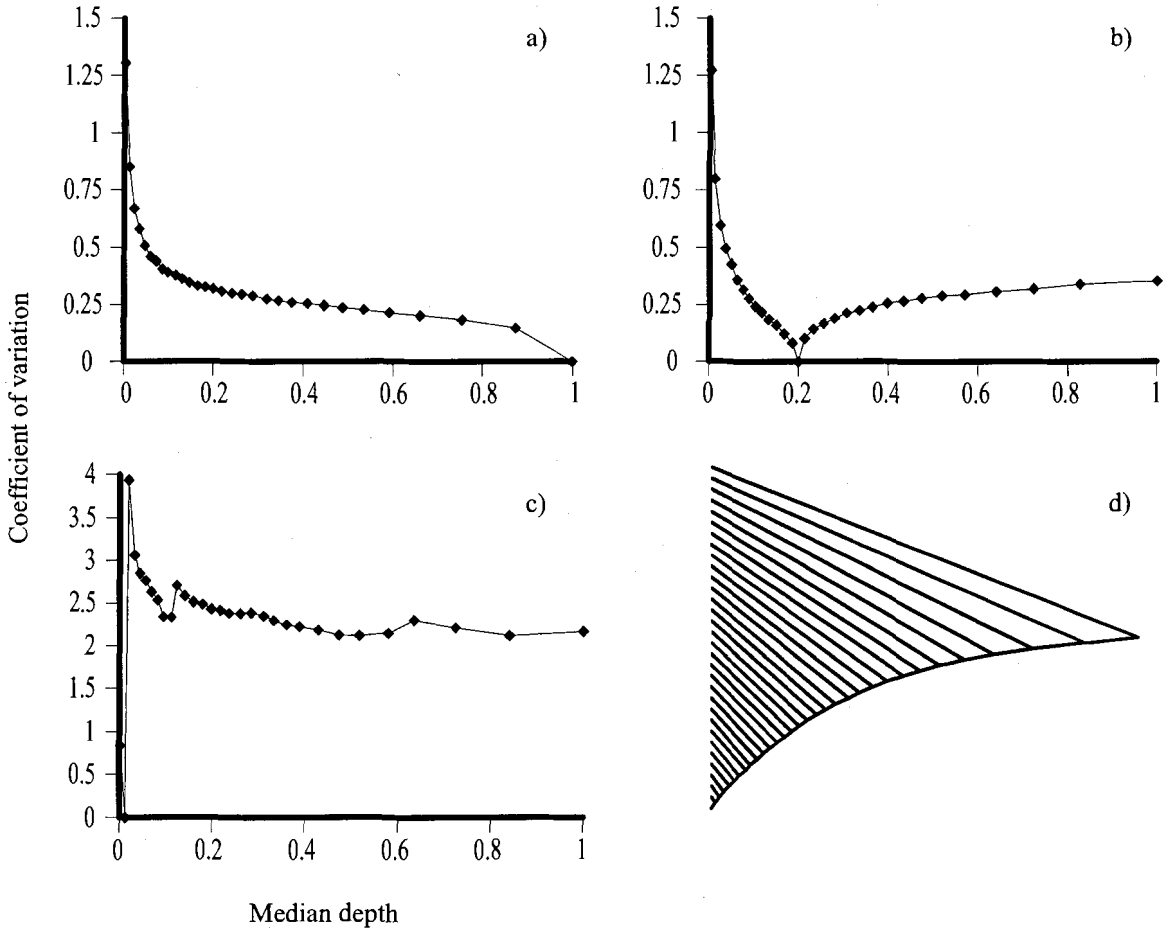


FIGURE IV-3 THREE CALIBRATION SCENARIOS

a) calibration on the split between Homo sapiens and Pan (the calibration point lies at a depth of 6 MYA); b) calibration on the split between the Cercopithecinae and the Colobinae (22.2 MYA); and c) calibration on the root (63 MYA).

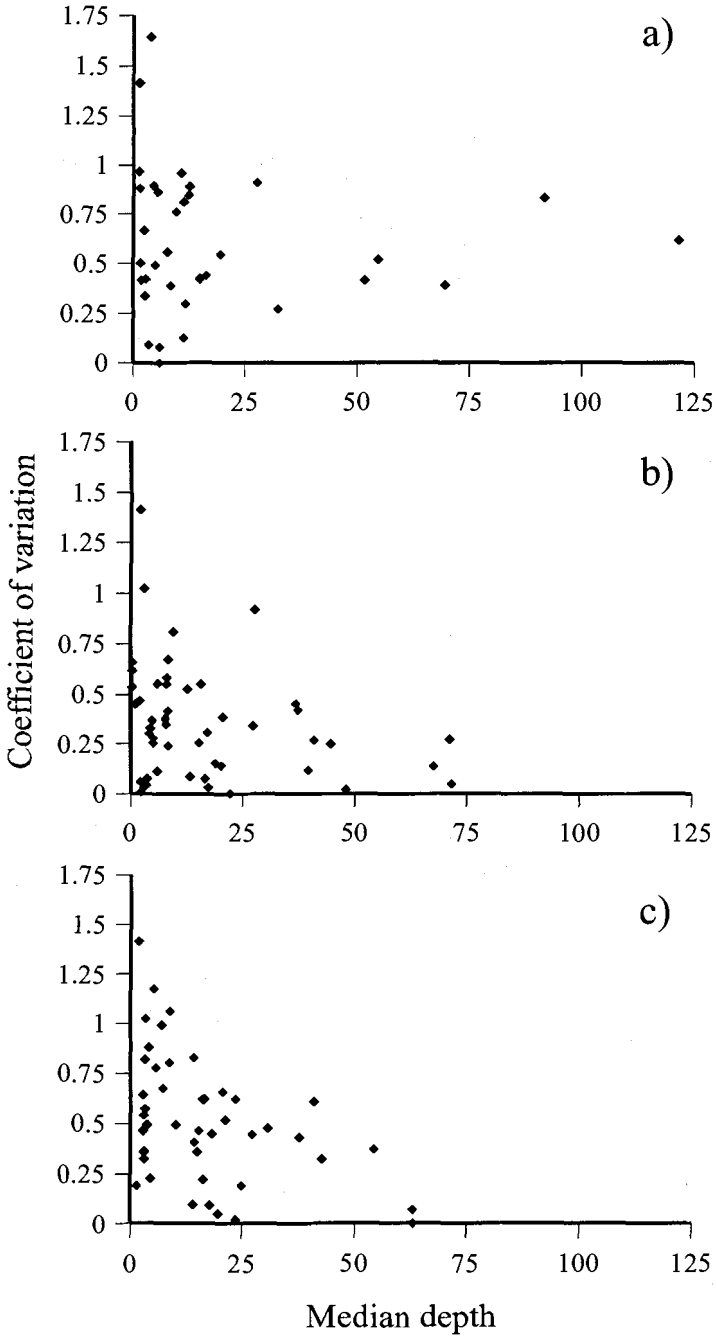


FIGURE IV-4 SELECTED DATES OF PRIMATE DIVERGENCES.

Dates inferred using the methods outlined in the text. Numbers above nodes are from Table IV-2; numbers in front of nodes are divergence dates in MYA.

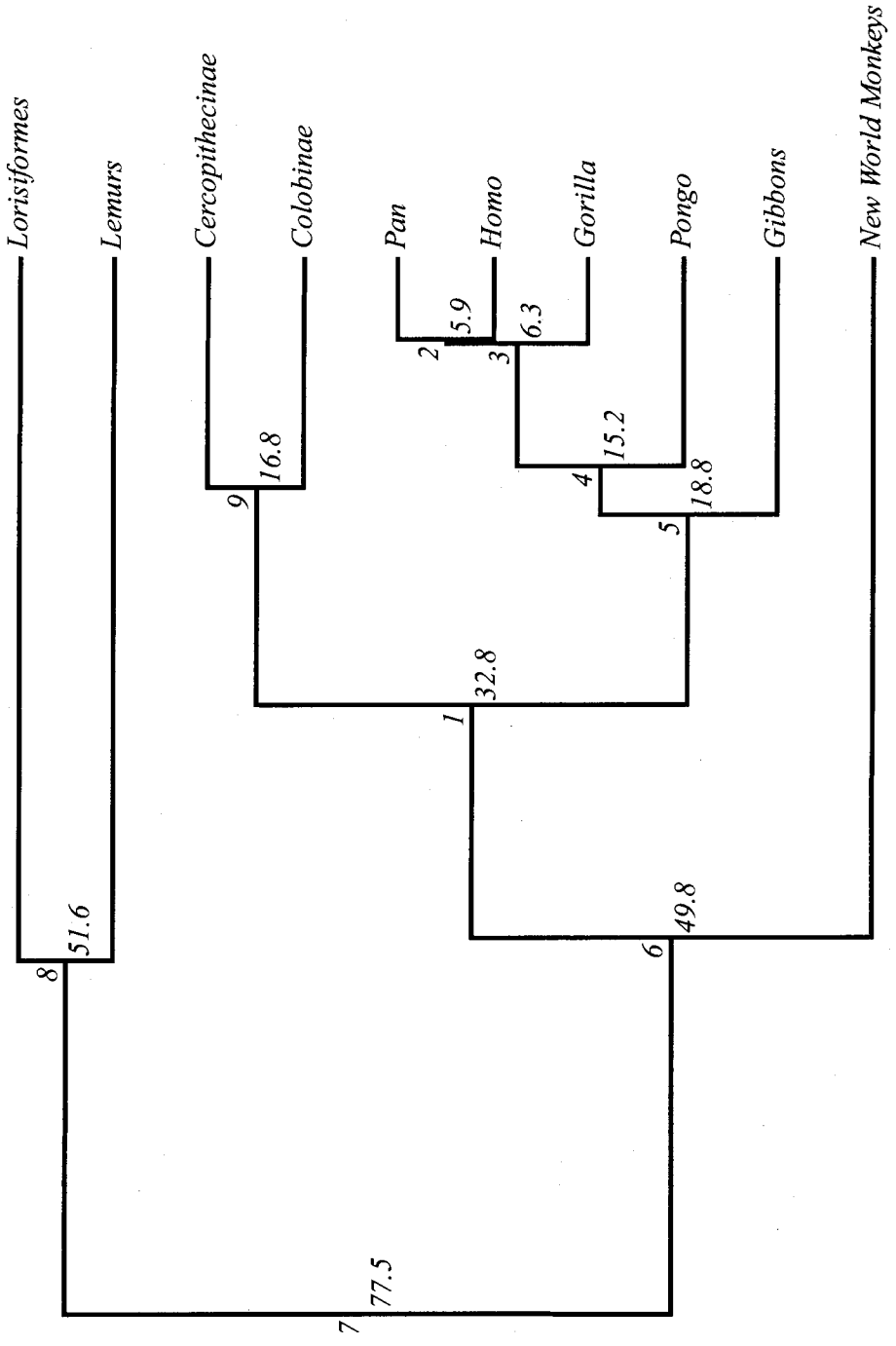
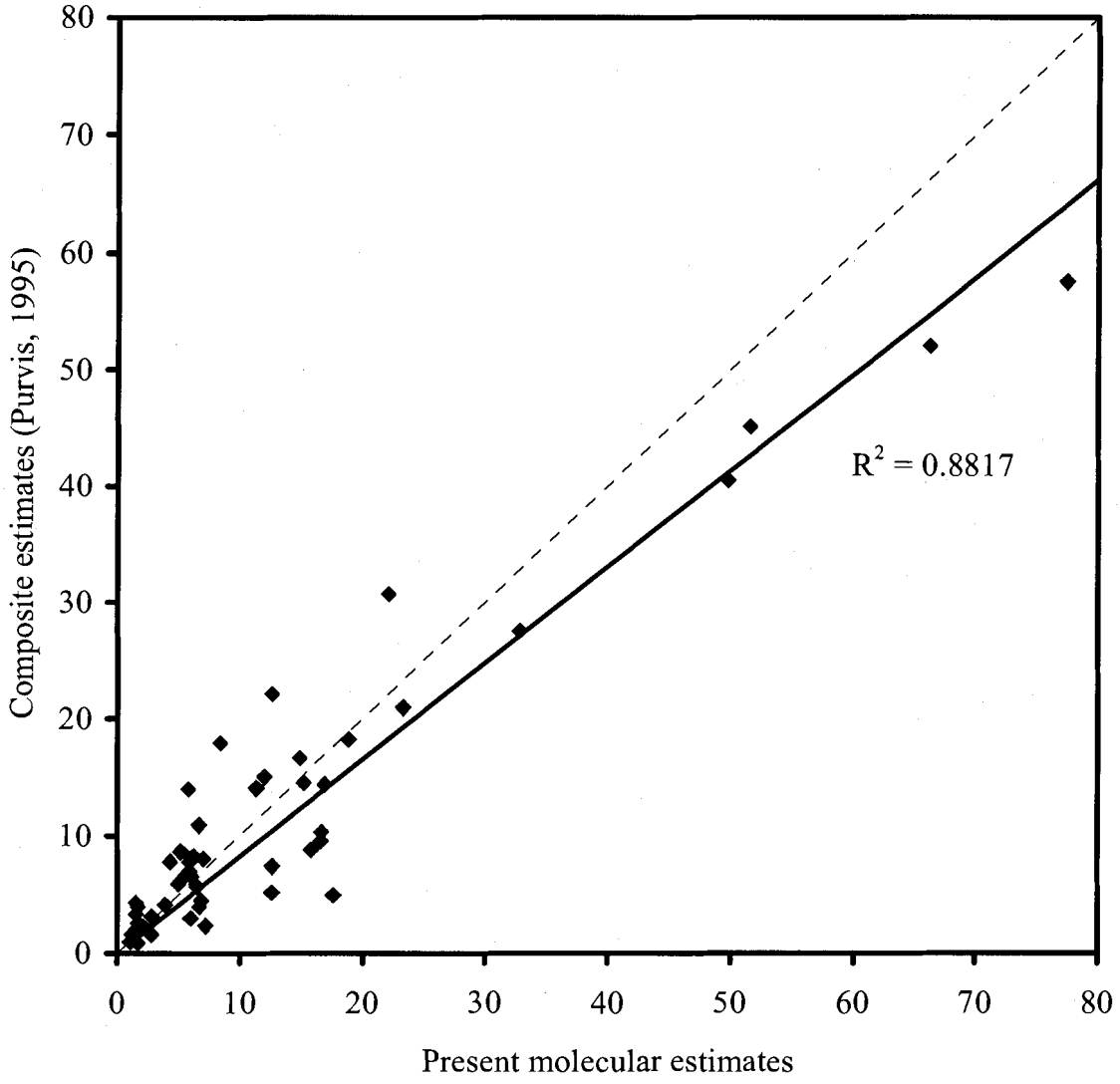


FIGURE IV-5 COMPARISON OF DIVERGENCE DATE ESTIMATES

Comparison of previously published composite estimates of divergence dates (from Purvis, 1995) with present estimates. Dotted line indicates 1:1 relationship. See text for further details.



CHAPTER V - A DATED MRP SUPERTREE FOR THE ORDER PRIMATES[4]

R. A. Vos and A. Ø. Mooers

[4] This chapter is in revision with *Systematic Biology*

ABSTRACT

Supertrees are phylogenetic trees whose topologies are derived from a set of partially overlapping, smaller trees. Here we present a supertree for the order Primates, constructed by analyzing source trees we binary-coded using Matrix Representation using Parsimony analysis (MRP). The source trees are derived from previously published studies, and are based on a variety of data types, reconstructed using a variety of methods. In terms of the number of species included in the analysis, the composite estimate presented here is the largest phylogeny of the Primates available to date; it is well-resolved and generally fits with our understanding of the systematics of this order. Areas with few or incongruent data are identified using strength-of-grouping and rQS values. In addition to this updated topology, we present divergence dates estimated using a new method that utilizes overlapping fossil-calibrated clock-like DNA sequence data. For the nodes for which no molecular estimates are available we provide expected ages under a pure birth model. We analyze the divergence date estimates and find that, overall, the Order Primates has diversified at a constant rate of cladogenesis, though looked at in more detail we note, in agreement with earlier studies, a significantly elevated rate in the cercopithecines.

Keywords: MRP; Maximum Parsimony; Phylogeny; Divergence Dates; Primates; Supertrees.

INTRODUCTION

The phylogeny of the order Primates, the order to which we belong, encompasses some of the most intensively studied problems in systematics. Many novel techniques (e.g. modeling of substitution rate heterogeneity) have first been applied to parts of the primate tree, for example to the 'hominid trichotomy', that is, the topology of the set of *Homo*, *Pan* and *Gorilla* (Adachi and Hasegawa, 1995; Arnason et al., 1998; Penny et al., 1995; Rogers, 1993; Suzuki et al., 1994). It is therefore perhaps surprising that only one species-level phylogeny of the whole order has been available, in a study that was published over a decade ago (Purvis, 1995, see also Purvis and Webster, 1999). The scope and methodological sophistication of phylogenetics has grown in the years since, and a new estimate that incorporates the subsequent accumulation of phylogenetic knowledge is timely. Here we present such a study, in the form of a supertree analysis of previously published phylogenies of the order Primates. In addition, we present divergence dates estimated from calibrated, clock-like DNA sequences, and expected node ages under a pure birth model for those nodes for which no molecular data are available.

SUPERTREES

A sobering fact in phylogenetics is that few estimates of large trees, regardless of methodology, accurately represents the branching order or divergence times of the part of the true tree of life it intends to approximate (though see Rokas et al., 2003). The reasons for this are many, ranging from the fact that sometimes the pattern of evolution of the underlying comparative data is different from that of the divergence of the taxa under study (e.g., the 'gene tree' versus 'species tree' problem *sensu* Maddison, 1997) to

methodological issues such as inconsistency in some of the most commonly used methods of inference (Gaut and Lewis, 1995; Huelsenbeck, 1995; Huelsenbeck and Lander, 2003).

Supertrees are phylogenetic trees whose topologies are derived from a set of usually smaller, overlapping trees - here referred to as 'source trees'. The accuracy of the estimates obtained using supertree methods is therefore sensitive to all the methodological issues associated with the source trees, even if the supertree method itself were without methodological problems of its own. Supertree methods have received criticism on this potential 'garbage in, garbage out' problem (Gatesy et al., 2002; Springer and De Jong, 2001). Careful selection of source trees and of the method of analysis should however address most of the issues raised (Bininda-Emonds et al., 2002; Bininda-Emonds et al., 2003). We therefore contend that supertrees can be usefully applied to a number of different issues in phylogenetics and evolutionary biology that require large phylogenies, at least while other methods of combined data analysis (such as the 'supermatrix' approach) are in their infancy or are unable to achieve the same level of taxon coverage with the presently available data. For example, a supertree based on reasonably congruent source trees that sufficiently cover the groups under study can be used for comparative studies (e.g. in Nunn, 1999; Thierry et al., 2000) or macro-evolutionary studies (e.g. in Gittleman and Purvis, 1998; Purvis et al., 1995) of a broader taxonomic scope than any single presently available phylogeny. Also, a supertree can be interpreted as a summary of what is currently known about the phylogeny of a given group – and so might steer research priorities, or might serve as a 'snapshot' of the data currently contained in a phylogenetic database (Page, 2004). The supertree that we

present here is intended to serve these purposes: to be a useful tool for primatologists and evolutionary biologists while indicating which parts of the primate tree need further investigation.

MATRIX REPRESENTATION USING PARSIMONY ANALYSIS

Different methods for the amalgamation of source trees are available; these can be classified into those that i) directly combine source tree topologies, such as the MinCut algorithm (Page, 2002; Semple and Steel, 2000); or ii) two-step methods that combine source trees through some form of matrix representation (Baum, 1992; Ragan, 1992; Ronquist et al., 2004) of their shape, where the set of representations is analyzed under an optimality criterion and conventional search strategies, or, as more recently proposed, using Metropolis-Hastings coupled Monte Carlo Markov chains (Ronquist et al., 2004).

The most commonly used supertree method, Matrix Representation using Parsimony analysis, or MRP (Baum, 1992; Ragan, 1992), follows a two-step approach. MRP is in essence a method by which tree shapes can be coded into binary matrices that can be interpreted as 'pseudo-character state matrices' and thus can be analyzed using the methods available for such data – in practice usually heuristic searching under Maximum Parsimony (e.g. in Bininda-Emonds et al., 1999; Jones et al., 2002; Liu et al., 2001; Purvis, 1995).

MRP, and by extension the supertrees built using this method, have met with considerable skepticism (Gatesy et al., 2002; Gatesy and Springer, 2004; Springer and De Jong, 2001), and defense (Bininda-Emonds et al., 2004; Bininda-Emonds et al., 2003). Our rationale for choosing MRP is as follows:

i) Combined MRP matrices can be analyzed using fairly well characterized methods such as heuristic searching under Maximum Parsimony; that is, the properties of the inferential techniques employed in supertree reconstruction using MRP are well characterized;

ii) The methods available for the analysis of MRP matrices have been implemented in commonly used software packages for phylogenetic inference such as PAUP*4b10 (Swofford, 2003) or MrBayes (Huelsenbeck and Ronquist, 2003);

iii) Additional methods have been developed to calculate support on MRP supertrees, such as the rQS method (Bininda-Emonds et al., 2005), employed here.

DIVERGENCE DATE ESTIMATES ON SUPERTREES

For some robust types of comparative analyses, setting all branch lengths to the same arbitrary non-zero value (or no value at all) serves as sufficient correction for shared ancestry (Maddison, 1990). However, for some other types of analyses (such as those of speciation and extinction rates) trees must be ultrametric and the relative node depths must be proportional to the ages of the clades they subtend. As the purpose of this study is to present an estimate that is of use for these kinds of studies we present divergence dates estimated using a novel method based on overlapping clock-like genes (Vos and Mooers, 2004), combined with estimates of clade growth that are discussed below.

METHODS

R.V. collected 200 source trees from 130 references dating from 1993 onwards covering 219 species. Source trees were located through Web of Science database searches using "phylogen* AND primates" or "taxonom* AND primates" as search terms. Subsequently, R.V. scanned through the references of the articles as well as through all issues of the *American Journal of Primatology*, the *American Journal of Physical Anthropology*, *Evolutionary Anthropology*, *Folia Primatologia*, the *International Journal of Primatology*, the *Journal of Human Evolution*, the *Journal of Molecular Evolution*, *Molecular Biology and Evolution*, *Molecular Phylogenetics and Evolution*, *Primates* and *Systematic Biology*.

A criticism that has been leveled at supertrees is the potential for data duplication: multiple, published trees might be based on the same data set, thus biasing the supertree topology toward that of overrepresented source topologies (Gatesy et al., 2002; Springer and De Jong, 2001). A tree selection protocol should therefore be employed. We followed the recommendations made by Bininda-Emonds *et al.* (Bininda-Emonds et al., 2004), in summary:

- Within publications, source trees may be accepted that:
  - Are based on independent data sources; or, if multiple trees are presented where one is based on a superset of the other (e.g. all changes versus transversions only) the tree based on more inclusive data is used. As a second best, the tree explicitly preferred by the authors is used (usually this is the case where different optimality criteria are used, such as maximum parsimony and maximum likelihood, and the authors prefer the

'more sophisticated' maximum likelihood), or, as a last resort, a consensus over trees based on non-independent data sources.

  - Analyze non-overlapping taxon sets. If there is overlap, the most comprehensive tree is used.

- Between publications, source trees may be accepted that:
  - In addition to the conditions for within-publication source trees, are the result of the more recent analysis. Research groups sometimes publish a series of articles based on a growing set of sequences. In cases like this, the most recent source is preferred.

In addition to this tree selection process, candidate trees must be modified in several ways to prepare them for the supertree analysis. In many phylogenetic analyses, multiple outgroups are used, which are constrained to be monophyletic with respect to the ingroup. Since this constraint is somewhat subjectively imposed, we pruned all outgroup taxa from the source trees. Lastly, we collapsed all clades consisting of below species-level OTUs (e.g. subspecies, haplotypes). In many cases, this is straightforward: in a source tree (((A1,A2),B),C) where A1 and A2 are two below species-level instances of taxon A, the resulting tree is ((A,B),C). Where haplotypes or subspecies do not form monophyletic groups, the source tree is collapsed in a conservative, agnostic manner: if the source tree's topology is (((A1,B),C),A2), the result is the polytomy (A,B,C). Synonymous taxa were identified using the Primates section of the *Mammal Species of the World* taxonomy (Groves, 1993); the names we use here follow the conventions therein. The choice for this taxonomy, and its online database version (http://www.nmnh.si.edu/msw/), which we used for disambiguation, was made in order to

retain compatibility with the consortium producing a supertree of all Mammals, to which we have contributed our data.

All source trees we collected were taken from articles published after 1993 to avoid overlap with an additional 174 source trees (counting the Purvis's compartmentalization of the Order) included in this study from a previously published primate supertree (Purvis, 1995). We converted this pooled collection of 374 source trees into MRP (Baum, 1992; Ragan, 1992) matrices using RadCon (Thorley and Page, 2000). No formal attempt was made to correct for non-independence between the two sets of source trees (e.g due to recycling of data: some of the trees in the post-1993 dataset used similar data from the studies published pre 1993). However, as the newly collected source trees were to a much larger extent based on molecular data (88% of the trees based on molecular data, versus 33% for the Purvis data set) non-independence should be low even in the worst case. The data sets, commented to indicate where and why changes (of the types described above, i.e. collapsing of haplotype trees and taxonomic disambiguation) were made, are available from this journal's website.

PHYLOGENETIC INFERENCE

To infer the supertree topology we ran Parsimony Ratchet (Nixon, 1999) searches on the pooled data set. This approach is different from that taken by Purvis (1995), who analyzed the 'major' clades separately due to computing power constraints. The 'traditional' approach to heuristic searching consists of performing one or more independent searches, starting from a topology obtained through the stepwise addition of taxa in a randomized order. Rather than perform independent heuristic searches, the Parsimony Ratchet performs a single long search comprised of a series of short bouts of

optimization, alternating with searches on a perturbed tree landscape to escape from local optima (Nixon, 1999). Using this strategy, some of the true phylogenetic signal is retained during reweighted hill-climbing cycles, while greatly reducing the time spent in stepwise addition. Ratcheting techniques have been used with success in supertree construction (Jones et al., 2002), and phylogenetic inference using morphological data sets (Faivovich, 2002; Fontal-Cazalla et al., 2002; Quicke et al., 2001), molecular data sets (Malia et al., 2002; Simmons et al., 2002) and combined molecular and morphological data sets (Giribet et al., 2002).

We used the Parsimony Ratchet strategy as implemented in PAUPRat (Sikes and Lewis, 2001). PAUPRat constructs a script file that is executed in PAUP* (Swofford, 2003) in combination with the MRP data set in NEXUS (Maddison et al., 1997) format. The weighting scheme used in this study is the default 'uniform' setting of PAUPRat (Sikes and Lewis, 2001). Under this scheme, the initial weight of all characters is set to 1 (other options are 'additive' and 'multiply', both of which preserve *a priori* defined weighting schemes though they differ in the way the predefined weights under these schemes are altered). A user-defined percentage of characters is sampled with replacement from the data and the weight of these characters is incremented by one. Because characters are sampled with replacement, some characters may have their weights adjusted multiple times.

In our study, experimentation with different reweighting schemes showed no improvement in the length of the shortest retrieved tree anywhere above 15% upweighted characters, which is the percentage used to obtain the results we report here.

Each search consisted of 200 iterations. In a previous supertree study (Salamin et al., 2002) it was shown that treating MRP character states as irreversible could increase the resolution of the resulting phylogeny, and irreversible MRP appears to be as accurate as standard MRP (Bininda-Emonds and Sanderson, 2001). We therefore ran the searches using this modification of maximum parsimony. We constructed majority rule and strict consensus trees over the resulting set of unique optimal trees. The majority rule consensus tree and MRP matrix have been submitted to TreeBASE under accession number "SN2421".

Inference of Bremer values and rQS values.

A Bremer value (Bremer, 1994) on a clade is the difference in the number of steps between the most parsimonious tree and the shortest tree that does not include that clade and can be interpreted as the net support for the most parsimonious grouping of a set of taxa compared to the second most parsimonious grouping. In a supertree context, Bremer values are difficult to interpret because of the non-independence of the MRP pseudocharacters. Nevertheless, we calculated Bremer values using a script we wrote that traverses through the supertree's topology, and for each node writes the Parsimony Ratchet (Nixon, 1999) commands modified to include an 'inverse constraint' requirement for the taxa the focal node subtends (an 'inverse constraint' on a heuristic search is a rule to reject all proposed topologies that include a predefined taxon bipartition or constraint tree shape). The script is available in the Bio::Phylo package at http://search.cpan.org/~rvosa/. As an alternative, perhaps more suitable measure of source tree support for supertree topology we also calculated the reduced qualitative

support (rQS) index (Bininda-Emonds et al., 2005) using a script kindly provided by Olaf Bininda-Emonds.

DIVERGENCE DATE ESTIMATION

Although several approaches now exist to combine branch lengths on source trees in a supertree analysis (Bryant et al., 2004; Semple et al., 2004), most of our source trees have no time-based branch lengths, and our aim in this study is to present a supertree based on previously published sources. Hence, although we could have pooled additional, unpublished, molecular phylogenies into the supertree analysis, we chose instead to estimate divergence dates in a separate procedure. The protocol we followed is discussed in more detail in Vos & Mooers (2004). In short, we collected sequence data from the GenBank flat file release 132.0. We parsed the sequence meta data for the Primates to collect suitable candidate genes, focusing on loci with high taxon coverage for the Order, which we aligned using ClustalW (Thompson et al., 1994) and subsequently by hand using Se-Al v2.0a11 (http://evolve.zoo.ox.ac.uk/software.html?id=seal).

For each of the alignments we selected the appropriate substitution model - constrained to the supertree's topology - using MODELTEST (Posada and Crandall, 1998). We then tested, using a likelihood ratio test with a liberal alpha-level of 0.001 whether the gene's evolution on the primate supertree was consistent with a molecular clock. The liberal alpha dealt with possible Type II error due to multiple genes and the known illiberality of likelihood-based molecular clock tests (Norman and Ashley, 2000; Yang et al., 1995; Zhang, 1999).

For candidate loci that did not conform to the molecular clock we tested whether removing some deviating taxa from the alignment would create a subset that did conform

to a clock. We did this by iteratively pruning the tips that most deviate from the average root-to-tip path length, performing the likelihood ratio test after each iteration (though retaining the original model of evolution; this may offer a slight bias, if lineages evolving at different rates also evolve under different molecular processes). A script that automates this procedure is available from the authors. This yielded, in total, 55 datasets containing on average 17 sequences (median = 12, range 4-59) after pruning on average three to four sequences (median = 0, range 0-63) (see Table V-2). For each of these loci we estimated the branch lengths under the appropriate substitution model. An alternative approach that disregards the molecular clock entirely would be to obtain ultrametric trees by non-parametric rate smoothing (Sanderson, 1997) or penalized likelihood (Sanderson, 2002) – however, NPRS seems to produce biased estimates, with nodes concentrated nearer the root (Ruber and Zardoya, 2005): penalized likelihood could have been used on a locus-by-locus approach, but we would still have had to combine the resulting subtrees.

In order to combine the dates from these 55 ultrametricized trees, we did the following:

1. Nine nodes were chosen that were both present in a large number of the subtrees and that could be dated from the literature. These nodes were: i) calibrated on the split between the apes and the Old World monkeys; ii) on the split between *Homo* and *Pan*; iii) on the split between *Homo* and *Pan*, and *Gorilla*; iv) the basal split of the great apes; v) the split between the great apes and the gibbons; vi) the split between the Old World monkeys apes, and the New World monkeys; vii) the split between the lemurs and the lorisiforms; viii) the split between the colobines and the cercopithecines; ix) and the root. We dated each of these nodes using several

(partially overlapping but widely-cited) estimates from the recent literature (Adachi and Hasegawa, 1995; Adachi and Hasegawa, 1996; Arnason et al., 2000; Arnason et al., 1998; Arnason et al., 1996; Arnason et al., 1996; Easteal and Herbert, 1997; Gingerich and Uhen, 1994; Goodman et al., 1998; Kumar and Hedges, 1998; Nei and Glazko, 2002; Porter et al., 1997; Purvis, 1995; Stauffer et al., 2001; Yoder, 1997). Where multiple estimates for the age of a particular split (node) were available from the literature, the median was used (see Table V-2).

2. Each of the 55 ultrametric trees was then scaled using these anchor points: some trees will have only a single anchor, while others may have several – in the latter case, separate trees were constructed, one for each anchor. This gives us a large set (252) of dated trees.
3. We constructed subsets of trees, grouped according to the anchor used. Within each, the anchors were fixed, and the other nodes adjusted accordingly: for any node with more than one estimate (i.e. found in more than one tree in the subset), we took the median, an approach also taken in earlier studies that combine different divergence date estimates for the same node (e.g. in Purvis, 1995).
4. Finally, because the node ages were well-behaved, we took the mean (and Standard Error) of the age of each node across the nine subsets corresponding to the nine different anchors. This entire procedure is designed to weight the dates from the literature equally.

However , as this approach combines estimates from different loci (each with their own rates and models of evolution) there is no obvious way in which the robustness of the composite dates can be quantified.

For a large number of clades (corresponding to 116 out of 210 or 55% of the nodes in the final tree) no molecular estimates were available. The most common method for inferring such unknown ages is to assume a model of diversification (Purvis, 1995, who applied this method to 70 of 160 or 44% of the nodes). We approximated the expected age of the nodes in these clades by randomly drawing, with replacement, 10^6 labeled histories (that is, phylogenies with a chronological ordering of internal nodes). For each of these histories we considered the set of expected time intervals between speciation events ($1/n$ where n is the number of accumulated lineages), and thus node ages, if speciation had proceeded under a pure birth model since the age of the most recent common ancestor for which a molecular estimate was available (this method now has an analytical solution, Gernhard and Steel, personal communication; Gernhard, 2006). As an aside, this approach could be modified to generate more sophisticated waiting times expectations, for example considering extinction. We subsequently averaged over the set of histories to arrive at the approximations presented here.

We performed the calculations using a module we wrote for the *Mesquite* program (Maddison and Maddison, 2001) that is available from the authors on request. This method avoids an unrecognized property of a simpler one used for previous supertrees (Purvis et al., 1995; and Bininda-Emonds et al., 1999), where clade age was made proportional to the natural logarithm of clade size relative to that of an ancestral, dated clade: even on a fully pectinate tree, the simpler method, applied iteratively, produces node ages that actually model a slow-down in diversification rate (i.e. progressively longer waiting times) whereas waiting times are understood to shorten as cladogenesis proceeds under constant speciation, and even more so if extinction is taken

into consideration. The method used here incorporates a constant diversification rate that is identical for all tree shapes (see Figure V-1).

Results and Discussion

The MRP data set we analyzed consisted of 218 taxa (excluding the hypothetical outgroup) and 2368 binary pseudo-characters. For all reweighting fractions of at least 10% the shortest trees found had a length of 3214. 200 ratchet iterations (15% reweighted) yielded 187 distinct most parsimonious trees. However, constructing a majority rule consensus tree yields a result with very little conflict: on average, every node in the majority rule consensus tree (which we present here) is present in 96.55% of the 187 most parsimonious trees. The supertree is generally well resolved with a resolution (expressed as the number of internal branches the tree contains divided by the number of maximum possible (n-2 for binary trees, Colless, 1980) of 96.77%. (The corresponding resolution of the strict consensus tree is 85.32%). The deepest splits in the topology are shown in Figure V-2, and the triangular tips it subtends are expanded in the subsequent Figure V-3 through Figure V-13. The numbers on the nodes correspond with those in Table V-3, which shows the estimated and interpolated divergence dates and the support values.

Resolution and support

In the context of supertrees, Bremer support values may be interpreted as indicative of the net number of source trees that unequivocally support a node. Low values may indicate either a low total number of source trees that include at least two taxa on either side of the focal node in the supertree or incongruence among studies. However, the groupings of taxa, the nodes that are reconstructed in an MRP supertree, are not only determined by the groupings proposed by the source trees and their support but also by the relative support for neighboring nodes. The source trees may suggest a certain

grouping unequivocally, yet the topology of the supertree may not include that grouping because of the stronger support for a surrounding topology that precludes it. Conversely, a grouping may be reconstructed in the supertree that has no support from source trees yet is necessitated by the surrounding reconstructed topology. This explains why MRP supertrees sometimes include nodes that have no support from source trees or are even contradicted by them. In our study, ten nodes were introduced in the majority rule supertree for which no supporting source trees exist in our data set (see Table V-3), and three such nodes appear in the strict supertree.

For most of the topology, the genera identified in the taxonomy (Groves, 1993) are reconstructed as monophyletic. The only exceptions are *Trachypithecus* (Figure V-5) and *Galago* (Figure V-13). Bremer support values for these clades are low, most notably for nodes 149-156 in Figure V-5 (*Trachypithecus*) and Table V-3 and nodes 201, 203, 205 and 206 in Figure V-13 (*Galago*). The largest polytomy in the tree (node 148 in Figure V-5, a split of four taxa) indicates another weakly supported area of the Primate phylogeny. Other areas in the tree however are quite strongly supported: the 'Hominid trichotomy' (nodes 88, 89 and 90) as it is shown in Figure V-6 has Bremer support values of 79 for the root of the subtree, and 49 and 47 for the subsequent splits, the highest values in the tree. Average support over the whole tree was 5.77, with a standard deviation of 2.06.

Comparison with the Purvis Supertreee

Both the present study and that undertaken by Purvis (1995) set out to infer a composite tree using previously published phylogenetic information. Hence, although the intention is to cover as many species as possible, not all known Primate species are

included in either of these studies. The disparity in taxon coverage between the two (Purvis 1995: 203 taxa, present study: 218 taxa) is caused by the fact that many phylogenies have been published in the meantime, some of which included data on species for which Purvis had no data available. Conversely, Purvis distinguished some taxa that are considered to belong to the same species, or are subspecies, in the taxonomy we followed (Wilson & Reeder, 1993).

Methodologically, the two studies differ in that the Purvis study analyzed subsets of the primate supertree (putative clades) which were then attached to a backbone, while the advances in computer power, as well as the introduction of the Parsimony Ratchet algorithm (Nixon, 1999), allowed us to analyze the complete dataset at once, freeing us from having to make *a priori* assumptions about monophyletic subdivisions in the primate supertree.

Another difference lies in the estimation of divergence dates: Purvis incorporated disparate data on divergence timing such as rescaled source tree topologies and karyological clocks; the present study uses molecular data directly. Where no data on divergence timing were available, Purvis used a different method to interpolate dates (clade age was made proportional to the natural logarithm of clade size), while here we choose a technique that more closely approximates the clade growth curve that is expected under a pure birth model.

In order to compare the two studies, we first disambiguated classification conflicts using the Mammal Species of the World online database (http://nmnhgoph.si.edu/msw/, accessed 12/16/2003) and pruned the two strict consensus trees to the same set of 191 taxa.

We compared the two trees on a node-by-node basis, identifying the nodes that subtend the same set of terminal taxa as a match (i.e. bipartitions, the subtree topology may still differ), and the nodes in the present tree that subtend a set absent in the Purvis (1995) tree as a conflict (see Table V-4). The most notable conflicts are nodes 76 and 42 in the New World Monkeys, and node 194 in the strepsirrhines (see Table V-3). The former two nodes indicate a rearrangement in the placement of *Callicebus* - Purvis (1995) places these as the sister group of *Aotus* (see Figure V-7, Figure V-8, Figure V-9). The latter conflict is caused by the placement of *Daubentonia madagascariensis,* which the present study places as basal to all Malagasy Strepsirrhines, while Purvis (1995) places it as basal to the Indridae. Nevertheless, the number of matching clades is 112 (out of 150 nodes in the Purvis (1995) tree), with the remaining conflicts confined to shallow nodes.

Finally, we asked whether there was a difference in the support of source trees for the different topologies. We calculated, for each source tree, its fit to the supertrees using c.i. (the compatibility index; Rodrigo, 1992). We would expect that pre-1993 trees should fit the Purvis tree better than the present tree, and that post-1993 trees should fit the present tree better. This is true: the mean c.i. of the pre-1993 source trees on the present supertree is 0.72, that of the post-1993 source trees is 0.85; conversely, the mean c.i. of the pre-1993 trees on the Purvis supertree is 0.74, and that of the post-1993 trees is 0.73. Using the same approach, we compared the fit of source trees based on molecular data, and source trees based on morphological data. The c.i. of all source trees based on molecular data is 0.86, that of morphological data 0.73. In the pooled MRP data set, 1639 out of 2289 pseudocharacters are nodes from source trees based on molecular data. As the newly collected source trees are mostly molecular (88%), whereas Purvis's data was

mostly morphological, (33% molecular), we can conclude that the growth in molecular phylogenetics in recent years has swamped and 'outvoted' the earlier Purvis data set.

IMPLICATIONS FOR SYSTEMATICS

In our tree, Cercopithecidae divides into the monophyletic traditional subfamilies Colobinae (i.e. the genera *Trachypithecus*, *Presbytis*, *Semnopithecus*, *Pygathrix*, *Nasalis*, *Colobus* and *Procolobus*) and Cercopithecinae (the genera *Macaca*, *Cercocebus*, *Mandrillus*, *Papio*, *Theropithecus*, *Lophocebus*, *Cercopithecus*, *Chlorocebus*, *Erythrocebus*, *Miopithecus* and *Allenopithecus*). Within the family, all but the genus *Trachypithecus* are reconstructed as monophyletic clades.

The results that we present here with respect to the Great Apes (Figure V-2, Figure V-6) conform to the overwhelming consensus of recent years; that is, the 'hominid trichotomy' is rooted on *Gorilla*, and *Pongo* is the most basal of the great apes.

The New World monkeys, the platyrrhines (Figure V-7, Figure V-8, Figure V-9 and Figure V-10), are a group whose deeper, intergeneric, relationships are still uncertain. In our results all genera are monophyletic, and can be grouped into three clades: Cebidae (*Cebus*, *Saimiri*, *Aotus* and the callitrichines); Atelidae (the atelines) and Pitheciidae (the pithecines).

The phylogenetic position of tarsiers (*Tarsius*, Figure V-2, Figure V-11) has long been controversial (Martin, 1990; Shoshani et al., 1996). Some authors have grouped tarsiers with the strepsirhines in the suborder Prosimii (Fleagle, 1988; Schwartz, 1986; Simpson, 1945), or as basal to the entire primate tree (Gingerich, 1973; Gingerich, 1975). Recently, the prevailing view is to group the tarsiers with the anthropoids (Groves, 1993; Nowak, 1991), and most molecular studies support this (but see (Jaworski, 1995). As a

result of these new studies, the tree presented here includes this grouping, with a relatively strong Bremer support of 9 (Table V-3).

All the Malagasy strepsirhines form a strongly supported (Table V-3, Figure V-2, Figure V-12) monophyletic group, congruent with the hypothesis of a single colonization of the island (e.g. Yoder et al., 1996). Within the Malagasy strepsirhines the tree is well resolved, and all genera (*sensu* Groves, 1993) are monophyletic. In our results, *Lemur catta* is reconstructed as basal to the genus *Hapalemur,* and *Daubentonia* is basal to the whole clade.

Recent findings suggest that the species diversity is perhaps far greater among the mouse lemurs *Microcebus* than previously thought (Rasoloarison et al., 2000; Yoder et al., 2000). As well, the subspecies diversity among *Eulemur* is large and complex (Djelati et al., 1997; Pastorini et al., 2000; Tattersall and Sussman, 1998; Wyner et al., 1999). In this study we have taken the conservative approach of only including those taxa recognized by Groves (1993), noting however that there may be more Lemurs than are presented here.

The other members of the strepsirhines, the lorisoids (Figure V-2, Figure V-13) are members of weakly supported groupings (Table V-3). For example, the genus *Galago* forms a polyphyletic group (Figure V-13). More data need to be brought to bear on the phylogenetic affinities within this group.

RATES OF CLADOGENESIS

The molecular estimates of divergence time were calculated under a constraint of on average equal rates across all lineages, i.e. a molecular clock. Alternative approaches to derive divergence dates from DNA sequences are possible, such as a two-step

approach where branch lengths obtained without the assumption of rate constancy are 'linearized' using one of several available approaches such as penalized likelihood, (Sanderson, 2002) or non-parametric rate smoothing, (Sanderson, 1997). However, these techniques yield transformed estimates that, at best (for PL), are similar to estimates using a molecular clock, and at worst (for NPRS) are biased towards a clustering of nodes near the root (Ruber and Zardoya, 2005). Hence, we prefer to report estimates based on data that are consistent with a molecular clock, supplemented with interpolated expected dates under a pure-birth model.

The age of many of the nodes near the tips of the supertree are based solely on pure-birth modeling, as homologous clock-like sequences for sister species, given the sparse sampling in the database, are rarer in GenBank than those for more distantly related species that speak to deeper nodes on the supertree. However, the estimates based on molecular data are not concentrated in any one clade, and are found distributed over all depths in the tree (see Figure V-14).

For the combined molecular and interpolated divergence dates (the gray curve in Figure V-15), the slope of the lineage-through-time plot suggests constant growth. A one-tailed test of the gamma statistic (Pybus and Harvey, 2000, implemented in Paradis *et al.*, 2003) confirms this (γ=2.4766, n=218, $p<0.05$). Expected waiting times for the interpolated divergence dates were generated using this same model, which might favor the selection of the constant growth model. By pruning those lineages for which no sequence data (and so no molecular estimate of divergence time) is available we obtain a subtree of molecular estimates. If we confine ourselves to this subset of lineages the same clade growth model is selected (γ=3.51723, n=110, $p<0.05$), though consequently with a

lower estimated net growth rate – see the black curve in Figure V-15 – hence the constant growth model is in any case not favored due to the pure birth interpolation.

A look in more detail at the major clades broadly confirms earlier analyses (Chan and Moore, 2002; Moore et al., 2004; Paradis, 1998; Purvis et al., 1995) suggesting a significantly elevated rate of cladogenesis in Cercopithecinae (Table V-5) compared to all others (when fitting a birth-death model by maximum likelihood to the branching times, using the method of Nee et al., 1994, as implemented in Paradis *et al.*, 2003). Whether this increased rate is due to key innovations, ecological opportunity, or some combination of factors remains an open question.

CONCLUSIONS

The supertree presented here is the largest supertree estimate of phylogeny for the order Primates to date (though see Purvis et al., 1999, where a composite tree of 233 species is presented). The findings presented here agree to a large extent with earlier findings – as well they should, considering that this study essentially summarizes prior research. In that context the present study also exposes a dearth of phylogenetic and molecular data for some groups. Notable among these are Galaginae (Figure V-13), for which some paraphyletic genera were reconstructed. Another problematic area of the Primate supertree is that of the deep nodes in the New World monkeys (e.g. Figure V-7, Figure V-8), where contradictory source trees contribute to the phylogenetic instability. Indeed, this ability to highlight areas that need more work may be one of the main uses of the supertree method to systematists. Supertree results can also be viewed in this light as indicators of where molecular data collection should be directed.

Our study also illustrates an approach to combining divergence dates derived from disparate molecular data under the assumption of rate constancy. The assumption of a molecular clock is a contentious issue in phylogenetics. It is possible that rate smoothing approaches (Sanderson, 1997; Sanderson, 2002) may serve as an alternative means of deriving composite molecular estimates of divergence dates, and we call for more work here. Any approach must deal with the robustness and the quantitative measure of robustness of composite estimates (see, eg., Magallon and Sanderson, 2001), and the ability to include partially overlapping data.

To interpolate divergence dates for splits for which no molecular estimates were available we employed a simulation technique whereby large sets of labeled histories and

their expected waiting times under a pure birth model were sampled. As we know very little about the actual time course of macroevolution, we have little evidence for this being a reasonable general model for waiting times (e.g. compared with adaptive radiation or saturated community models; see Mooers et al., in press). More work is desperately needed here.

ACKNOWLEDGEMENTS

The authors would like to thank Rod Page, Colin Groves, Phyllis Lee, Wayne Maddison, the SFU FAB*-lab, and especially Olaf Bininda-Emonds and Andy Purvis for input and very helpful comments on the many earlier versions of this manuscript. Our work was supported by both a SFU Graduate Fellowship and a Graduate Research Award from the SSB to RAV, and by an NSERC Canada Discovery Grant to AOM.

REFERENCES

Adachi, J., and M. Hasegawa. 1995. Improved Dating of the Human Chimpanzee Separation in the Mitochondrial-DNA Tree - Heterogeneity among Amino-Acid Sites. J. Mol. Evol. 40:622-628.

Adachi, J., and M. Hasegawa. 1996. Tempo and mode of synonymous substitutions in mitochondrial DNA of Primates. Mol. Biol. Evol. 13:200-208.

Adkins, R. M., and R. L. Honeycutt. 1994. Evolution of the Primate Cytochrome *c* Oxidase Subunit II Gene. J. Mol. Evol. 38:215-231.

Arnason, U., A. Gullberg, A. S. Burguete, and A. Janke. 2000. Molecular estimates of primate divergences and new hypotheses for primate dispersal and the origin of modern humans. Hereditas 133:217-228.

Arnason, U., A. Gullberg, and A. Janke. 1998. Molecular timing of primate divergences as estimated by two nonprimate calibration points. J. Mol. Evol. 47:718-727.

Arnason, U., A. Gullberg, A. Janke, and X. Xu. 1996. Pattern and timing of evolutionary divergences among hominoids based on analyses of complete mtDNAs. J. Mol. Evol. 43:650-661.

Arnason, U., X. F. Xu, A. Gullberg, and D. Graur. 1996. The "Phoca standard": An external molecular reference for calibrating recent evolutionary divergences. J. Mol. Evol. 43:41-45.

Baum, B. R. 1992. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon 41:3-10.

Bininda-Emonds, O. R. P., and H. N. Bryant. 1998. Properties of Matrix Representation with Parsimony Analysis. Syst. Biol. 47:497-508.

Bininda-Emonds, O. R. P., J. L. Gittleman, and S. A. Price. 2005. A complete phylogeny of the whales, dolphins and even-toed hoofed mammals (Cetartiodactyla). Biol. Rev. 80:445-473.

Bininda-Emonds, O. R. P., J. L. Gittleman, and A. Purvis. 1999. Building large trees by combining phylogenetic information: a complete phylogeny of the extant Carnivora (Mammalia). Biol. Rev. 74:143-175.

Bininda-Emonds, O. R. P., J. L. Gittleman, and M. A. Steel. 2002. The (super)tree of life: procedures, problems and prospects. Annu. Rev. Ecol. Syst.:265-289.

Bininda-Emonds, O. R. P., K. E. Jones, S. A. Price, M. Cardillo, R. Grenyer, and A. Purvis. 2004. Garbage in, garbage out: data issues in supertree construction. *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Publishers, Dordrecht.

Bininda-Emonds, O. R. P., K. E. Jones, S. A. Price, R. Grenyer, M. Cardillo, M. Habib, A. Purvis, and J. L. Gittleman. 2003. Supertrees Are a Necessary Not-So-Evil: A Comment on Gatesy et al. Syst. Biol. 52:724-729.

Bininda-Emonds, O. R. P., M. J. Sanderson. 2001. Assessment of the Accuracy of Matrix Representation with Parsimony Analysis Supertree Construction. Syst. Biol. 50:565-579.

Bremer, K. 1994. Branch support and tree stability. Cladistics 10:295-304.

Bryant, D., C. Semple, and M. A. Steel. 2004. Supertree methods for ancestral divergence dates and other applications. Pages 129-150 *in* Phylogenetic supertrees: combining information to reveal the tree of life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Publishers, Dordrecht.

Chan, K. M. A., and B. R. Moore. 2002. Whole-tree methods for detecting differential diversification rates. Syst. Biol. 51:855-865.

Colless, D. H. 1980. Congruence between morphometric and allozyme data for *Menidia* species: A reappraisal. Syst. Zool. 29:288-299.

Djelati, R., B. Brun, and Y. Rumpler. 1997. Meiotic study of hybrids in the genus *Eulemur* and taxonomic considerations. Am. J. Primatol. 42:235-245.

Easteal, S., and G. Herbert. 1997. Molecular Evidence from the Nuclear Genome for the Time Frame of Human Evolution. J. Mol. Evol. 44(Suppl 1):S121-S132.

Faivovich, J. 2002. A cladistic analysis of Scinax (Anura: Hylidae). Cladistics 18:367-393.

Fleagle, J. G. 1988. Primate adaptation and evolution. Academic Press, San Diego, CA.

Fontal-Cazalla, F. M., M. L. Buffington, G. Nordlander, J. Liljeblad, P. Ros-Farre, J. L. Nieves-Aldrey, J. Pujade-Villar, and F. Ronquist. 2002. Phylogeny of the Eucoilinae (Hymenoptera: Cynipoidea: Figitidae). Cladistics 18:154-199.

Gatesy, J., C. Matthee, R. Desalle, and C. Hayashi. 2002. Resolution of a supertree/supermatrix paradox. Syst. Biol. 51:652-664.

Gatesy, J., and M. Springer. 2004. A critique of matrix representation with parsimony supertrees *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht.

Gaut, B. S., and P. O. Lewis. 1995. Success of Maximum Likelihood Phylogeny Inference in the Four-Taxon Case. Mol. Biol. Evol. 12:152-162.

Gernhard, T. 2006. Stochastic Models for Speciation Events in Phylogenetic Trees. Dissertation. Zentrum Mathematik Technische Universität München, Munich.

Gingerich, P. D. 1973. Anatomy of the temporal bone in the oligocene anthropoid *Apidium* and the origin of Anthroipoidea. Folia Primatol. 19:329-337.

Gingerich, P. D. 1975. Dentition of *Adapis parisiensis* and the evolution of lemuriform primates. Pages 65-80 *in* Lemur biology (I. Tattersall, and R. W. Sussman, eds.). Plenum Press, New York.

Gingerich, P. D., and M. D. Uhen. 1994. Time of Origin of Primates. J. Hum. Evol. 27:443-445.

Giribet, G., G. D. Edgecombe, W. C. Wheeler, and C. Babbit. 2002. Phylogeny and Systematic Position of Opiliones: A Combined Analysis of Chelicerate Relationships Using Morphological and Molecular Data. Cladistics 18:5-70.

Gittleman, J. L., and A. Purvis. 1998. Body size and species-richness in carnivores and primates. Proc. R. Soc. Lond. B Biol. Sci. 265:113-119.

Goddard, W., E. Kubicka, G. Kubicki, and F. R. Mcmorris. 1994. The agreement metric for labeled binary trees. Math. Biosci. 123:215-226.

Goodman, M., W. J. Bailey, K. Hayasaka, M. J. Stanhope, M. J. Slightom, and J. Czelusniak. 1994. Molecular evidence on primate phylogeny from DNA sequences. Am. J. Phys. Anthropol. 94:3-24.

Goodman, M., C. A. Porter, J. Czelusniak, S. L. Page, H. Schneider, J. Shoshani, G. F. Gunnell, and C. Groves. 1998. Toward a phylogenetic classification of primates based on DNA evidence complemented by fossil evidence. Mol. Phylogenet. Evol. 9:585-598.

Groves, C. 1993. Order Primates *in* Mammal Species of the World (D. E. Wilson, and D. M. Reeder, eds.). Smithsonian Institution Press, Washington, DC.

Huelsenbeck, J. P. 1995. Performance of phylogenetic methods in simulation. Syst. Biol. 44:17-48.

Huelsenbeck, J. P., and K. M. Lander. 2003. Frequent inconsistency of parsimony under a simple model of cladogenesis. Syst. Biol. 52:641-648.

Huelsenbeck, J. P., and F. Ronquist. 2003. MrBayes, version 3.0B4.

Jaworski, C. J. 1995. A Reassessment of Mammalian αA-Crystallin Sequences Using DNA Sequencing: Implications for Anthropoid Affinities of Tarsier. J. Mol. Evol. 41:901-908.

Jones, K. E., A. Purvis, A. Maclarnon, O. R. P. Bininda-Emonds, and N. B. Simmons. 2002. A phylogenetic supertree of the bats (Mammalia: Chiroptera). Biol. Rev. 77.

Kumar, S., and S. B. Hedges. 1998. A molecular timescale for vertebrate evolution. Nature 392:917-920.

Liu, F.-G. R., M. M. Miyamoto, N. P. Freire, P. Q. Ong, M. R. Tennant, T. S. Young, and K. F. Gugel. 2001. Molecular and Morphological Supertrees for Eutherian (Placental) Mammals. Science 291:1786-1789.

Maddison, D. R., D. L. Swofford, and W. P. Maddison. 1997. NEXUS: An extensible file format for systematic information. Syst. Biol. 46:590-621.

Maddison, W. P. 1990. A method for testing the correlated evolution of two binary characters: Are gains or losses concentrated on certain branches of a phylogenetic tree? Evolution 44:539-557.

Maddison, W. P. 1997. Gene trees in species trees. Syst. Biol. 46:523-536.

Maddison, W. P., and D. R. Maddison. 2001. Mesquite: a modular system for evolutionary analysis.

Magallon, S., and M. J. Sanderson. 2001. Absolute diversification rates in Angiosperm clades. Evolution 55:1762-1780.

Malia, M. J., R. M. Adkins, and M. W. Allard. 2002. Molecular support for Afrotheria and the polyphyly of Lipotyphla based on analyses of the growth hormone receptor gene. Mol. Phylogenet. Evol. 24:91-101.

Martin, R. D. 1990. Primate Origins and Evolution. Chapman & Hall, London.

Moore, B. R., K. M. A. Chan, and M. J. Donoghue. 2004. Detecting diversification rate variation in supertrees *in* Phylogenetic supertrees: Combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Publishers, Dordrecht.

Mooers, A.O., L. J. Harmon, D. H. J. Wong, and S.B. Heard. In press. Some models of tree shape. *In* New Mathematical Models for Evolution (O. Gascuel and M. Steel, eds.), Oxford University Press, Oxford.

Nee, S., R. M. May and P. H. Harvey. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311

Nei, M., and G. V. Glazko. 2002. Estimation of divergence times for a few mammalian and several primate species. J. Hered. 93:157-164.

Nixon, K. 1999. The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis. Cladistics 15:407-414.

Norman, J. E., and M. V. Ashley. 2000. Phylogenetics of Perissodactyla and tests of the molecular clock. J. Mol. Evol. 50:11-21.

Nowak, R. M. 1991. Walker's mammals of the world, 4th edition. Johns Hopkins University Press, Baltimore.

Nunn, C. L. 1999. The evolution of exaggerated sexual swellings in primates and the graded-signal hypothesis. Anim. Behav. 58:229-246.

Page, R. D. M. 2002. Modified mincut supertrees. Pages 537-551 *in* WABI2002, Rome.

Page, R. D. M. 2004. Phyloinformatics: Towards a Phylogenetic Database *in* Data Mining in Bioinformatics (J. T. L. Wang, et al., eds.). Springer-Verlag, Heidelberg.

Paradis, E. 1998. Detecting shifts in diversification rates without fossils. American Naturalist 152:176-187.

Paradis, E., K. Strimmer, J. Claude, Y. Noel, and B. Bolker. 2003. Analyses of Phylogenetics and Evolution: the *ape* Package, version 1.0.

Pastorini, J., M. R. J. Forstner, and R. D. Martin. 2000. Relationships among Brown Lemurs (*Eulemur fulvus*) Based on Mitochondrial DNA Sequences. Mol. Phylogenet. Evol. 16:418-429.

Penny, D., and M. P. Hendy. 1985. The use of tree comparison metrics. Syst. Zool. 34:75-82.

Penny, D., M. A. Steel, P. J. Waddell, and M. P. Hendy. 1995. Improved Analyses of Human mtDNA Sequences Support a Recent African Origin for *Homo sapiens*. Mol. Biol. Evol. 12:863-882.

Porter, C. A., J. Czelusniak, H. Schneider, M. P. C. Schneider, I. Sampaio, and M. Goodman. 1997a. Sequences of the primate ε-globin gene: implications for systematics of the marmosets and other New World primates. Gene 205:59-71.

Porter, C. A., S. L. Page, J. Czelusniak, H. Schneider, M. P. C. Schneider, I. Sampaio, and M. Goodman. 1997b. Phylogeny and evolution of selected primates as determined by sequences of the ε-globin locus and 5' flanking regions. Int. J. Primatol. 18:261-295.

Posada, D., and K. A. Crandall. 1998. Modeltest: testing the model of DNA substitution. Bioinformatics 14:817-818.

Purvis, A. 1995. A composite estimate of primate phylogeny. Philos. Trans. R. Soc. Lond. B Biol. Sci. 348:405-421.

Purvis, A., S. Nee, and P. H. Harvey. 1995. Macroevolutionary Inferences from Primate Phylogeny. Proc. R. Soc. Lond. B Biol. Sci. 260:329-333.

Purvis, A. and A. J. Webster. 1999. Phylogenetically independent comparisons and primate phylogeny. *In* Comparative primate socioecology (P. C. Lee ed.), pp. 44-70. Cambridge, Cambridge University Press.

Pybus, O. G., M. A. Charleston, S. Gupta, A. Rambaut, E. C. Holmes, and P. H. Harvey. 2001. The epidemic behaviour of the Hepatitis *C* virus. Science 292:2323-2325.

Pybus, O. G., and P. H. Harvey. 2000. Testing macro-evolutionary models using incomplete molecular phylogenies. Proc. R. Soc. Lond. B Biol. Sci. 267:2267-2272.

Pybus, O. G., A. Rambaut, and P. H. Harvey. 2000. An integrated framework for the inference of viral population history from reconstructed genealogies. Genetics 155:1429-1437.

Quicke, D. L. J., J. Taylor, and A. Purvis. 2001. Changing the landscape: A new strategy for estimating large phylogenies. Syst. Biol. 50:60-66.

Ragan, M. A. 1992. Phylogenetic inference based on matrix representation of trees. Mol. Phylogenet. Evol. 1:53-58.

Rasoloarison, R. M., S. M. Goodman, and J. U. Ganzhorn. 2000. Taxonomic revision of mouse lemurs (*Microcebus*) in the western portions of Madagascar. Int. J. Primatol. 21:963-1019.

Rodrigo, A. G. 1992. Two optimality criteria for selecting subsets of most parsimonious trees. Syst. Biol. 41:33-40

Rogers, J. 1993. The phylogenetic relationships among *Homo*, *Pan* and *Gorilla*: a population genetics perspective. J. Hum. Evol. 25:201-215.

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

Ronquist, F., J. P. Huelsenbeck, and T. Britton. 2004. Bayesian supertrees *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht.

Ruber, L., and R. Zardoya. 2005. Rapid cladogenesis in marine fish revisited. Evolution 59:1119-1127.

Salamin, N., T. R. Hodkinson, and V. Savolainen. 2002. Building Supertrees: an Empirical Assessment using the Grass Family (Poaceae). Syst. Biol. 51:136-150.

Sanderson, M. J. 1997. A nonparametric approach to estimating divergence times in the absence of rate constancy. Mol. Biol. Evol. 14:1218-1231.

Sanderson, M. J. 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. Mol. Biol. Evol. 19:101-109.

Schwartz, J. H. 1986. Primate systematics and a classification of the order. Pages 1-41 *in* Systematics, Evolution and Anatomy (D. R. Swindler, and J. Erwin, eds.). A. R. Liss, New York.

Semple, C., P. Daniel, W. Hordijk, R. D. M. Page, and M. A. Steel. 2004. Supertree algorithms for ancestral divergence dates and nested taxa. Bioinformatics 20:1-6.

Semple, C., and M. A. Steel. 2000. A Supertree Method for Rooted Trees. Discrete Appl. Math. 105:147-158.

Shoshani, J., C. P. Groves, E. L. Simons, and G. F. Gunnell. 1996. Primate Phylogeny: Morphological vs Molecular Results. Mol. Phylogenet. Evol. 5:102-154.

Sikes, D. S., and P. O. Lewis. 2001. PAUPRat: PAUP* implementation of the parsimony ratchet., version 1 (beta). Department of Ecology and Evolutionary Biology, University of Connecticut, Storrs.

Simmons, M. P., H. Ochoterena, and J. V. Freudenstein. 2002. Amino acid vs. nucleotide characters: challenging preconceived notions. Mol. Phylogenet. Evol. 24:78-90.

Simpson, G. G. 1945. The principles of classification and a classification of the mammals. Bull. Amer. Mus. Natur. Hist. 85:1-350.

Springer, M. S., and W. W. De Jong. 2001. Which mammalian supertree to bark up? Science 291:1709-1711.

Stauffer, R. L., A. Walker, O. A. Ryder, M. Lyons-Weiler, and S. Blair Hedges. 2001. Human and ape molecular clocks and constraints on paleontological hypotheses. J. Hered. 92:469-474.

Strimmer, K., and O. G. Pybus. 2001. Exploring the demographic history of DNA sequences using the generalised skyline plot. Mol. Biol. Evol. 18:2298-2305.

Suzuki, H., Y. Kawamoto, O. Takenaka, I. Munchika, H. Hori, and S. Sakurai. 1994. Phylogenetic relationships among *Homo sapiens* and related species: based on restriction site variation in rDNA spacers. Biochem. Genet. 32:257-269.

Swofford, D. L. 2003. PAUP*: phylogenetic analysis using parsimony, version 4.0b10.

Tattersall, I., and R. W. Sussman. 1998. 'Little Brown Lemurs' of Northern Madagascar. Folia Primatol. 69:379-388.

Thierry, B., A. N. Iwaniuk, and S. M. Pellis. 2000. The Influence of Phylogeny on the Social Behaviour of Macaques (Primates: Cercopithecidae, genus *Macaca*). Ethology 106:713-728.

Thompson, J. D., D. G. Higgins, and T. J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucl. Acids Res. 22:4673-4680.

Thorley, J. L., and R. D. M. Page. 2000. RadCon: Phylogenetic tree comparison and consensus. Bioinformatics 16:486-487.

Vos, R. A., and A. Ø. Mooers. 2004. Reconstructing divergence times for supertrees: a molecular approach *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht.

Wyner, Y., R. Absher, G. Amato, E. Sterling, R. Stumpf, Y. Rumpler, and R. Desalle. 1999. Species concepts and the determination of historic gene flow patterns in the *Eulemur fulvus* (Brown Lemur) complex. Biol. J. Linn. Soc. 66:39-56.

Yang, Z. H., N. Goldman, and A. Friday. 1995. Maximum likelihood trees from DNA sequences: a peculiar statistical estimation problem. Syst. Biol. 44:384-399.

Yoder, A. D. 1997. Back to the future: A synthesis of strepsirrhine systematics. Evol. Anthropol. 6:11-22.

Yoder, A. D., M. Cartmill, M. Ruvolo, K. Smith, and R. Vilgalys. 1996. Ancient single origin for Malagasy primates. Proc. Nat. Acad. Sci. USA 93:5122-5126.

Yoder, A. D., R. M. Rasoloarison, S. M. Goodman, J. A. Irwin, S. Atsalis, M. J. Ravosa, and J. U. Ganzhorn. 2000. Remarkable species diversity in Malagasy mouse lemurs (primates, *Microcebus*). Proc. Nat. Acad. Sci. USA 97:11325-11330.

Zhang, J. Z. 1999. Performance of likelihood ratio tests of evolutionary hypotheses under inadequate substitution models. Mol. Biol. Evol. 16:868-875.

REFERENCES (SOURCE TREES)

Ackermann, R. R., and J. M. Cheverud. 2000. Phenotypic Covariance Structure in Tamarins (Genus *Saguinus*): A Comparison of Variation Patterns Using Matrix Correlation and Common Principal Component Analysis. Am. J. Phys. Anthropol. 111:489-501.

Adachi, J., and M. Hasegawa. 1995. Improved dating of the human-chimpanzee separation in the mitochondrial DNA tree: Heterogeneity among amino acid sites. J. Mol. Evol. 40:622-628.

Adkins, R. M., and R. L. Honeycutt. 1994. Evolution of the Primate Cytochrome *c* Oxidase Subunit II Gene. J. Mol. Evol. 38:215-231.

Andrews, T. D., L. S. Jermiin, and S. Easteal. 1998. Accelerated Evolution of Cytochrome *b* in Simian Primates: Adaptive Evolution in Concert with Other Mitochondrial Proteins? J. Mol. Evol. 47:249-257.

Apoil, P. A., and A. Blancher. 1999. Sequences and evolution of mammalian RH gene transcripts and proteins. Immunogenetics 49:15-25.

Arnason, U., A. Gullberg, A. S. Burguete, and A. Janke. 2000. Molecular estimates of primate divergences and new hypotheses for primate dispersal and the origin of modern humans. Hereditas 133:217-228.

Arnason, U., A. Gullberg, and A. Janke. 1998. Molecular timing of primate divergences as estimated by two nonprimate calibration points. J. Mol. Evol. 47:718-727.

Arnason, U., A. Gullberg, A. Janke, and X. Xu. 1996. Pattern and timing of evolutionary divergences among hominoids based on analyses of complete mtDNAs. J. Mol. Evol. 43:650-661.

Ashley, M. V., and J. L. Vaughn. 1995. Owl Monkeys (*Aotus*) are Highly Divergent in Mitochondrial Cytochrome *c* Oxidase (COII) Sequences. Int. J. Primatol. 16:793-806.

Barriel, V. 1997. *Pan paniscus* and Hominoid Phylogeny: Morphological Data, Molecular Data and 'Total Evidence'. Folia Primatol. 68:50-56.

Barroso, C. M. L., H. Schneider, M. P. C. Schneider, I. Sampaio, M. L. Harada, J. Czelusniak, and M. Goodman. 1997. Update on the phylogenetic systematics of the New World monkeys: Further DNA evidence for placing the pygmy marmoset (*Cebuella*) within the marmoset genus *Callithrix*. Int. J. Primatol. 18:651-674.

Boinski, S., and S. J. Cropp. 1999. Disparate Data Sets Resolve Squirrel Monkey (*Saimiri*) Taxonomy: Implications for Behavioral Ecology and Biomedial Usage. Int. J. Primatol. 20:237-256.

Boubli, J. P., and A. D. Ditchfield. 2000. The time of divergence between the two species of uacari monkeys: Cacajao calvus and Cacajao melanocephalus. Folia Primatol. 71:387-391.

Canavez, F. C., G. Alves, T. G. Fanning, and H. N. Seuanez. 1996. Comparative karyology and evolution of Amazonian *Callithrix* (Platyrrhini, Primates). Chromosoma 104:348-357.

Canavez, F. C., J. J. Ladasky, J. A. P. C. Muniz, H. N. Seuanez, and P. Parham. 1998. β_2-microglobulin in neotropical primates (Platyrrhini). Immunogenetics 48:133-140.

Canavez, F. C., M. A. M. Moreira, J. J. Ladasky, A. Pissinatii, P. Parham, and H. N. Seuanez. 1999. Molecular phylogeny of New World primates (Platyrrhini) based on β_2-microglobulin DNA sequences. Mol. Phylogenet. Evol. 12:74-82.

Canavez, F. C., M. A. M. Moreira, F. Simon, P. Parham, and H. N. Seuanez. 1999. Phylogenetic Relationships of the Callitrichinae (Platyrrhini, Primates) Based on β_2-Microglobulin DNA Sequences. Am. J. Primatol. 48:225-236.

Cao, Y., A. Janke, P. J. Waddell, M. Westerman, O. Takenaka, S. Murata, N. Okada, S. Pääbo, and M. Hasegawa. 1998. Conflict Among Individual Mitochondrial Proteins in Resolving the Phylogeny of Eutherian Orders. J. Mol. Evol. 47:307-322.

Casane, D., S. Boissinot, B. H.-J. Chang, L. C. Shimmin, and W.-H. Li. 1997. Mutation Pattern Variation Among Regions of the Primate Genome. J. Mol. Evol. 45:216-226.

Chaves, R., I. Sampaio, M. P. C. Schneider, H. Schneider, S. L. Page, and M. Goodman. 1999. The place of *Callimico goeldii* in the callitrichine phylogenetic tree: evidence from von Willebrand factor (vWF) intron II sequences. Mol. Phylogenet. Evol. 13:392-404.

Chiu, C.-H., H. Schneider, M. P. C. Schneider, I. Sampaio, C. M. Meireles, J. L. Slightom, D. L. Gumucio, and M. Goodman. 1996. Reduction of two functional γ-globin genes to one: An evolutionary trend in New World monkeys (infraorder Platyrrhini). Proc. Nat. Acad. Sci. USA 93:6510-6515.

Chiu, C.-H., H. Schneider, M. J. Slightom, D. L. Gumucio, and M. Goodman. 1997. Dynamics of regulatory evolution in primate β-globin gene clusters: *cis*-mediated acquisition of simian γ fetal expression patters. Gene 205:47-57.

Choong, C. S., J. A. Kemppainen, and E. M. Wilson. 1998. Evolution of the Primate Androgen Receptor: A Structural Basis for Disease. J. Mol. Evol. 47:334-342.

Collins, A. C., and J. M. Dubach. 2000. Phylogenetic Relationships of Spider Monkeys (*Ateles*) Based on Mitochondrial DNA Variation. Int. J. Primatol. 21:381-420.

Collins, A. C., and J. M. Dubach. 2001. Nuclear DNA Variation in Spider Monkeys (*Ateles*). Mol. Phylogenet. Evol. 19:67-75.

Collura, R. V., and C.-B. Stewart. 1995. Insertions and duplications of mtDNA in the nuclear genomes of Old World monkeys and hominoids. Nature 378:485-489.

Cropp, S. J., and S. Boinski. 2000. The Central American Squirrel Monkey (*Saimiri oerstedii*): Introduced Hybrid or Endemic Species? Mol. Phylogenet. Evol. 16:350-365.

Czelusniak, J., and M. Goodman. 1995. Hominoid phylogeny estimated by model selection using goodness of fit significance tests. Mol. Phylogenet. Evol. 4:283-290.

Delpero, M., S. Crovella, P. Cervella, G. Ardito, and Y. Rumpler. 1995. Phylogenetic Relationships Among Malagasy Lemurs as Revealed by Mitochondrial DNA Sequence Analysis. Primates 36:431-440.

Delpero, M., J. C. Masters, D. Zuccon, P. Cervella, S. Crovella, and G. Ardito. 2000. Mitochondrial Sequences as Indicators of Generic Classification in Bush Babies. Int. J. Primatol. 21:889-904.

Delson, E. 1994. The diversity of living Colobines *in* Colobine monkeys: their ecology, behaviour, and evolution (A. G. Davies, and J. F. Oates, eds.). Cambridge University Press, Cambridge.

Fracasso, C., and T. Patarnello. 1998. Evolution of the Dystrophin Muscular Promoter and the 5' Flanking Region in Primates. J. Mol. Evol. 46:168-179.

Funkhouser, W., B. F. Koop, P. Charmley, D. Martindale, J. L. Slightom, and L. Hood. 1997. Evolution and Selection of Primate T Cell Antigen Receptor BV8 Gene Subfamily. Mol. Phylogenet. Evol. 8:51-64.

Gebo, D. L., and E. J. Sargis. 1994. Terrestrial Adaptations in the Postcranial Skeletons of Guenons. Am. J. Phys. Anthropol. 93:341-371.

Geissmann, T. 1993. Evolution of Communication in Gibbons (Hylobatidae). Pages 162-173 *in* Philisophischen Fakultat II der Universitat Zurich Universitat Zurich, Zurich.

Goldberg, T. L., and M. Ruvolo. 1997. Molecular phylogenetics and historical biogeography of east African chimpanzees. Biol. J. Linn. Soc. 61:301-324.

Goodman, M., W. J. Bailey, K. Hayasaka, M. J. Stanhope, M. J. Slightom, and J. Czelusniak. 1994. Molecular evidence on primate phylogeny from DNA sequences. Am. J. Phys. Anthropol. 94:3-24.

Goodman, M., C. A. Porter, J. Czelusniak, S. L. Page, H. Schneider, J. Shoshani, G. F. Gunnell, and C. Groves. 1998. Toward a phylogenetic classification of primates based on DNA evidence complemented by fossil evidence. Mol. Phylogenet. Evol. 9:585-598.

Grossman, L. I., T. R. Schmidt, D. E. Wildman, and M. Goodman. 2001. Molecular evolution of aerobic energy metabolism in primates. Mol. Phylogenet. Evol. 18:26-36.

Groves, C. 1998. Systematics of Tarsiers and Lorises. Primates 39:13-27.

Groves, C., and J. W. H. Trueman. 1995. Lemurid systematics revisited. J. Hum. Evol. 28.

Hall, L. M., D. S. Jones, and B. A. Wood. 1998. Evolution of the Gibbon Subgenera Inferred from Cytochrome *b* DNA Sequence Data. Mol. Phylogenet. Evol. 10:281-286.

Hamdi, H., H. Nishio, R. Zielinski, and A. Dugaiczyk. 1999. Origin and phylogenetic distribution of Alu DNA repeats: Irreversible events in the evolution of primates. J. Mol. Biol. 289:861-871.

Hamrick, M. W. 1999. Pattern and process in the evolution of primate nails and claws. J. Hum. Evol. 37:293-297.

Harada, M. L., H. Schneider, M. P. C. Schneider, I. Sampaio, J. Czelusniak, and M. Goodman. 1995. DNA evidence on the phylogenetic systematics of New World monkeys: Support for the sister-grouping of *Cebus* and *Saimiri* from two unlinked nuclear genes. Mol. Phylogenet. Evol. 4:331-349.

Harris, E. E. 2000. Molecular systematics of Old World monkey tribe Papionini: analysis of the total available genetic sequences. J. Hum. Evol. 38:235-256.

Hayasaka, K., K. Fujii, and S. Horai. 1996. Molecular phylogeny of macaques: Implications of nucleotide sequences from an 896-base pair region of mitochondrial DNA. Mol. Biol. Evol. 13:1044-1053.

Hayashi, S., K. Hayasaka, O. Takenaka, and S. Horai. 1995. Molecular Phylogeny of Gibbons Inferred from Mitochondrial DNA Sequences: Preliminary Report. J. Mol. Evol. 41:359-365.

Horai, S., K. Hayasaka, R. Kondo, K. Tsugane, and N. Takahata. 1995. Recent African origin of modern humans revealed by complete sequences of hominoid mitochondrial DNAs. Proc. Nat. Acad. Sci. USA 92:532-535.

Horovitz, I., and A. Meyer. 1995. Systematics of the New World monkeys (Platyrrhini, Primates) based on 16s mitochondrial DNA sequences: A comparative analysis of different weighting methods in cladistic analysis. Mol. Phylogenet. Evol. 4:448-456.

Horovitz, I., R. Zardoya, and A. Meyer. 1998. Platyrrhine Systematics: A Simultaneous Analysis of Molecular and Morphological Data. Am. J. Phys. Anthropol. 106:261-281.

Huang, C.-H., Z. Liu, P.-A. Apoil, and A. Blancher. 2000. Sequence, Organization, and Evolution of Rh50 Glycoprotein Genes in Nonhuman Primates. J. Mol. Evol. 51:76-87.

Jablonski, N. G. 1995. The Phyletic Position and Systematics of the Douc Langurs of Southeast Asia. Am. J. Primatol. 35:185-205.

Jacobs, S. C., A. Larson, and J. M. Cheverud. 1995. Phylogenetic relationships and orthogenetic evolution of coat color among tamarins (genus *Saguinus*). Syst. Biol. 44:512-532.

Jaworski, C. J. 1995. A Reassessment of Mammalian αA-Crystallin Sequences Using DNA Sequencing: Implications for Anthropoid Affinities of Tarsier. J. Mol. Evol. 41:901-908.

Kermarrec, N., F. Roubinet, P. A. Apoil, and A. Blancher. 1999. Comparison of allele O sequences of the human and non-human primate ABO system. Immunogenetics 49:517-526.

Kim, H.-S., and O. Takenaka. 1996. A comparison of TSPY genes from Y-chromosomal DNA of the great apes and humans: sequence, evolution, and phylogeny. Am. J. Phys. Anthropol. 100:301-309.

Kitano, T., and N. Saitou. 1999. Evolution of Rh Blood Group Genes Have Experienced Gene Conversions and Positive Selection. J. Mol. Evol. 49:615-626.

Klonisch, T., C. Froehlich, F. Tetens, B. Fischer, and S. Hombach-Klonisch. 2001. Molecular Remodeling of Members of the Relaxin Family During Primate Evolution. Mol. Biol. Evol. 18:393-403.

Kobayashi, S. 1995. A Phylogenetic Study of Titi Monkeys, Genus *Callicebus*, Based on Cranial Measurements: I. Phyletic Groups of *Callicebus*. Primates 36:101-120.

Kuyl, A. C. V. D., J. T. Dekker, and J. Goudsmit. 2000. Primate Genus *Miopithecus*: Evidence for the Existence of Species and Subspecies of Dwarf Guenons Based on Cellular and Endogenous Viral Sequences. Mol. Phylogenet. Evol. 14.

Kuyl, A. C. V. D., C. L. Kuiken, J. T. Dekker, and J. Goudsmit. 1995. Phylogeny of African monkeys based upon mitochondrial 12s rRNA sequences. J. Mol. Evol. 40:173-180.

Liao, D., T. Pavelitz, and A. M. Weiner. 1998. Characterization of a Novel Class of Interspersed LTR Elements in Primate Genomes: Structure, Genomic Distribution, and Evolution. J. Mol. Evol. 46:649-660.

Macedonia, J. M., and K. F. Stanger. 1994. Phylogeny of the Lemuridae Revisited: Evidence from Communication Signals. Folia Primatol. 63:1-43.

Martinez, J., L. J. Dugaiczyk, R. Zielinski, and A. Dugaiczyk. 2001. Human genetic disorders, a phylogenetic perspective. J. Mol. Biol. 308:587-596.

Masters, J. C., R. J. Rayner, H. Ludewick, E. Zimmermann, N. Molez-Verriere, F. Vincent, and L. T. Nash. 1994. Phylogenetic Relationships Among the Galaginae as Indicated by Erythrocytic Allozymes. Primates 35:177-190.

Medeiros, M. A., R. M. S. Barros, J. C. Pieczarka, C. Y. Nagamachi, M. Ponsa, M. Garcia, and J. Egozcue. 1997. Radiation and Speciation of Spider Monkeys, Genus *Ateles*, From the Cytogenetic Viewpoint. Am. J. Primatol. 42:167-178.

Meireles, C. M., J. Czelusniak, M. P. C. Schneider, J. A. P. C. Muniz, M. C. Brigdo, H. S. Ferrera, and M. Goodman. 1999. Molecular phylogeny of ateline New World monkeys (Platyrrhini, atelinae) based on γ-globin sequences: Evidence that *Brachyteles* is the sister group of *Lagothrix*. Mol. Phylogenet. Evol. 12:10-30.

Messier, W., and C.-B. Stewart. 1997. Episodic adaptive evolution of primate lysozymes. Nature 385:151-154.

Mohammad-Ali, K., M.-E. Eladari, and F. Galibert. 1995. Gorilla and Orangutan c-myc Nucleotide Sequences: Inference on Hominoid Phylogeny. J. Mol. Evol. 41:262-276.

Morales, J. C., and D. J. Melnick. 1998. Phylogenetic relationships of the macaques (Cercopithecidae: *Macaca*) as revealed by high resolution restriction site mapping of mitochondrial ribosomal genes. J. Hum. Evol. 34:1-23.

Moreira, M. A. M., and H. N. Seuanez. 1999. Mitochondrial Pseudogenes and Phyletic Relationships of *Cebuella* and *Callithrix* (Platyrrhini, Primates). Primates 40:353-364.

Nagamachi, C. Y., J. C. Pieczarka, J. A. P. C. Muniz, R. M. S. Barros, and M. S. Mattevi. 1999. Proposed Chromosomal Phylogeny for the South American Primates of the Callitrichidae Family (Platyrrhini). Am. J. Primatol. 49:133-152.

Natori, M. 1994. Craniometrical Variations Among Eastern Brazilian Marmosets and Their Systematic Relationships. Primates 35:167-176.

Nikaido, M., M. L. Harada, Y. Cao, M. Hasegawa, and N. Okada. 2000. Monophyletic Origin of the Order Chiroptera and Its Phylogenetic Position Among Mammalia, as Inferred from the Complete Sequence of the Mitochondrial DNA of a Japanese Megabat, the Ryukyu Flying Fox (*Pteropus dasymallus*). J. Mol. Evol. 51:318-318.

O'huigin, C., A. Sato, and J. Klein. 1997. Evidence for convergent evolution of A and B blood group antigens in primates. Hum. Genet. 101:141-148.

Page, S. L., C.-H. Chiu, and M. Goodman. 1999. Molecular phylogeny of Old World monkeys (Cercopithecidae) as inferred from γ-globin DNA sequences. Mol. Phylogenet. Evol. 13:348-359.

Page, S. L., and M. Goodman. 2001. Catarrhine Phylogeny: Noncoding DNA Evidence for a Diphyletic Origin of the Mangabeys and for a Human-Chimpanzee Clade. Mol. Phylogenet. Evol. 18:14-25.

Pastorini, J., M. R. J. Forstner, and R. D. Martin. 2001. Phylogenetic history of sifakas (Propithecus: lemuriformes) derived from mtDNA sequences. Am. J. Primatol. 53:1-17.

Pastorini, J., M. R. J. Forstner, R. D. Martin, and D. J. Melnick. 1998. A Reexamination of the Phylogenetic Position of *Callimico* (Primates) Incorporating New Mitochondrial DNA Sequence data. J. Mol. Evol. 47:32-41.

Pastorini, J., R. D. Martin, P. Ehresmann, E. Zimmermann, and M. R. J. Forstner. 2001. Molecular phylogeny of the lemur family cheirogaleidae (primates) based on mitochondrial dna sequences. Mol. Phylogenet. Evol. 19:45-56.

Porter, C. A., I. Sampaio, H. Schneider, M. P. C. Schneider, J. Czelusniak, and M. Goodman. 1995. Evidence on primate phylogeny from ε-globin gene sequences and flanking regions. J. Mol. Evol. 40:30-55.

Porter, C. A., J. Czelusniak, H. Schneider, M. P. C. Schneider, I. Sampaio, and M. Goodman. 1997a. Sequences of the primate ε-globin gene: implications for systematics of the marmosets and other New World primates. Gene 205:59-71.

Porter, C. A., S. L. Page, J. Czelusniak, H. Schneider, M. P. C. Schneider, I. Sampaio, and M. Goodman. 1997b. Phylogeny and evolution of selected primates as determined by sequences of the ε-globin locus and 5' flanking regions. Int. J. Primatol. 18:261-295.

Porter, C. A., J. Czelusniak, H. Schneider, M. P. C. Schneider, I. Sampaio, and M. Goodman. 1999. Sequences From the 5' Flanking Region of the ε-Globin Gene Support the Relationship of *Callicebus* With the Pitheciins. Am. J. Primatol. 48:69-75.

Rannala, B., and Z. Yang. 1996. Probability Distribution of Molecular Evolutionary Trees: A New Method of Phylogenetic Inference. J. Mol. Evol. 43:304-311.

Rasmussen, D. T., and K. A. Nekaris. 1998. Evolutionary History of Lorisiform Primates. Folia Primatol. 69:250-285.

Razafindraibe, H., D. Montagnon, B. I. Ravoarimanana, and Y. Rumpler. 2000. Interspecific Nucleotide Sequence Differences in the Cytochrome B Gene of Indriidae (Primates, Strepsirhini). Primates 41:189-197.

Razafindraibe, H., D. Montagnon, and Y. Rumpler. 1997. Phylogenetic relationships among Indriidae (Primates, Strepsirhini) inferred from highly repeated DNA band patterns. C. R. Acad. Sci. Paris, Sciences de la vie 320:469-475.

Roos, C., and T. Geissmann. 2001. Molecular Phylogeny of the Major Hylobatid Divisions. Mol. Phylogenet. Evol. 19:486-494.

Rosenblum, L. L., J. Supriatna, M. N. Hasan, and D. J. Melnick. 1997. High Mitochondrial DNA Diversity with Little Structure Within and Among Leaf Monkey Populations (*Trachypithecus cristatus* and *Trachypithecus auratus*). Int. J. Primatol. 18:1005-10028.

Rumpler, Y., S. Crovella, and D. Montagnon. 1994. Systematic Relationships among Cheirogaleidae (Primates, Strepsirhini) Determined from Analysis of Highly Repeated DNA. Folia Primatol. 63:149-155.

Ruvolo, M. 1994. Molecular Evolutionary Processes and Conflicting Gene Trees: The Hominoid Case. Am. J. Phys. Anthropol. 94:89-113.

Samollow, P. B., L. M. Cherry, S. M. Witte, and J. Rogers. 1996. Interspecific Variation at the Y-linked RPS4Y Locus in Hominoids: Implications for Phylogeny. Am. J. Phys. Anthropol. 101:333-343.

Sampaio, I., M. P. C. Schneider, and H. Schneider. 1996. Taxonomy of the *Allouatta seniculus* Group: Biochemical and Chromosome Data. Primates 37:65-73.

Schmidt, T. R., M. Goodman, and L. I. Grossman. 1999. Molecular evolution of the COX7A gene family in primates. Mol. Biol. Evol. 16:619-626.

Schmitz, J., M. Ohme, and H. Zischler. 2001. SINE insertions in cladistic analyses and the phylogenetic affiliations of *Tarsius bancanus* to other primates. Genetics 157:777-784.

Schneider, H., I. Sampaio, M. L. Harada, C. M. L. Barroso, M. P. C. Schneider, J. Czelusniak, and M. Goodman. 1996. Molecular Phylogeny of the New World Monkeys (Platyrrhini, Primates) Based on Two Unlinked Nuclear Genes: IRBP Intron 1 and ε-Globin Sequences. Am. J. Phys. Anthropol. 100:153-179.

Shimmin, L. C., P. Mai, and W.-H. Li. 1997. Sequences and Evolution of Human and Squirrel Monkey Blue Opsin Genes. J. Mol. Evol. 44:378-382.

Soligo, C., and A. E. Müller. 1999. Nails and claws in primate evolution. J. Hum. Evol. 36:97-114.

Spek, C. A., R. M. Bertina, and P. H. Reitsma. 1998. Identification of Evolutionarily Invariant Sequences in the Protein C Gene Promoter. J. Mol. Evol. 47:663-669.

Stanhope, M. J., M. R. Smith, V. G. Waddell, C. A. Porter, M. S. Shivji, and M. Goodman. 1996. Mammalian Evolution and the Interphotoreceptor Retinoid Binding Protein (IRBP) Gene: Convincing Evidence for Several Superordinal Clades. J. Mol. Evol. 43:83-92.

Stanhope, M. J., V. G. Waddell, O. Madsen, W. W. De Jong, S. B. Hedges, G. C. Cleven, D. Kao, and M. S. Springer. 1998. Molecular evidence for multiple origins of Insectivora and for a new order of endemic African insectivore mammals. Proc. Nat. Acad. Sci. USA 95:9967-9972.

Suchard, M. A., R. E. Weiss, and J. S. Sinsheimer. 2001. Bayesian Selection of Continuous-Time Markov Chain Evolutionary Models. Mol. Biol. Evol. 18:1001-1013.

Suzuki, H., Y. Kawamoto, O. Takenaka, I. Munchika, H. Hori, and S. Sakurai. 1994. Phylogenetic relationships among *Homo sapiens* and related species: based on restriction site variation in rDNA spacers. Biochem. Genet. 32:257-269.

Tagliaro, C. H., H. Schneider, M. P. C. Schneider, I. Sampaio, and M. J. Stanhope. 1997. Marmoset phylogenetics, conservation perspectives, and evolution of the mtDNA control region. Mol. Biol. Evol. 14:674-684.

Tattersall, I., and R. W. Sussman. 1998. 'Little Brown Lemurs' of Northern Madagascar. Folia Primatol. 69:379-388.

Tosi, A. J., J. C. Morales, and D. J. Melnick. 2000. Comparison of Y Chromosome and mtDNA Phylogenies Leads to Unique Inferences of Macaque Evolutionary History. Mol. Phylogenet. Evol. 17:133-144.

Trtkova, K., W. E. Mayer, C. O'huigin, and J. Klein. 1995. *Mhc-DRB* Genes and the Origin of New World Monkeys. Mol. Phylogenet. Evol. 4:408-419.

Von Dornum, M., and M. Ruvolo. 1999. Phylogenetic relationships of the New World monkeys (Primates, Platyrrhini) based on nuclear G6PD sequences. Mol. Phylogenet. Evol. 11:459-476.

Wang, W., M. R. J. Forstner, Y.-P. Zhang, Z.-M. Liu, Y. Wei, H.-Q. Huang, H.-G. Hu, Y.-X. Xie, D.-H. Wu, and D. J. Melnick. 1997. A phylogeny of Chinese leaf monkeys using mitochondrial ND3-ND4 gene sequences. Int. J. Primatol. 18:305-320.

Wang, W., B. Su, H. Lan, Y.-P. Zhang, S.-Y. Lin, A.-H. Liu, R.-Q. Liu, W.-Z. Ji, H.-G. Hu, Y.-X. Xie, and D.-H. Wu. 1995. Phylogenetic Relationships among Two Species of Golden Monkey and Three Species of Leaf Monkey Inferred from rDNA Variation. Folia Primatol. 65:138-143.

Warter, S., M. Hauwy, and Y. Rumpler. 2000. Chromosome Painting Technique Contributes to Constructions of Evolutionary Trees of Lemurs. Int. J. Primatol. 21:905-913.

Weinreich, D. M. 2001. The Rates of Molecular Evolution in Rodent and Primate Mitochondrial DNA. J. Mol. Evol. 52:40-50.

Westhoff, C. M., and D. E. Wylie. 1996. Investigation of the RH Locus in Gorillas and Chimpanzees. J. Mol. Evol. 42:658-668.

Wilson, D. E., and D. M. Reeder (eds) 1993. Mammal Species of the World. Smithsonian Institution Press, Washington, DC.

Wu, W., M. Goodman, M. I. Lomax, and L. I. Grossman. 1997. Molecular Evolution of Cytochrome *c* Oxidase Subunit IV: Evidence for Positive Selection in Simian Primates. J. Mol. Evol. 44:477-491.

Wu, W., T. R. Schmidt, M. Goodman, and L. I. Grossman. 2000. Molecular Evolution of Cytochrome *c* Oxidase Subunit I in Primates: Is There Coevolution between Mitochondrial and Nuclear Genomes? Mol. Phylogenet. Evol. 17:294-304.

Wyner, Y., R. Absher, G. Amato, E. Sterling, R. Stumpf, Y. Rumpler, and R. Desalle. 1999. Species concepts and the determination of historic gene flow patterns in the *Eulemur fulvus* (Brown Lemur) complex. Biol. J. Linn. Soc. 66:39-56.

Wyner, Y., R. Desalle, and R. Absher. 2000. Phylogeny and Character Behavior in the Family Lemuridae. Mol. Phylogenet. Evol. 15:124-134.

Yang, Z., and A. D. Yoder. 1999. Estimation of the Transition/Transversion Rate Bias and Species Sampling. J. Mol. Evol. 48:274-283.

Yoder, A. D. 1994. Relative Position of the Cheirogaleidae in Strepsirhine Phylogeny: A Comparison of Morphological and Molecular Methods and Results. Am. J. Phys. Anthropol. 94:25-46.

Yoder, A. D. 1996. Strepsirrhine phylogeny: Congruence of results from a mitochondrial and a nuclear gene. Am. J. Phys. Anthropol. 22:250.

Yoder, A. D., M. Cartmill, M. Ruvolo, K. Smith, and R. Vilgalys. 1996. Ancient single origin for Malagasy primates. Proc. Nat. Acad. Sci. USA 93:5122-5126.

Yoder, A. D., R. M. Rasoloarison, S. M. Goodman, J. A. Irwin, S. Atsalis, M. J. Ravosa, and J. U. Ganzhorn. 2000. Remarkable species diversity in Malagasy mouse lemurs (primates, *Microcebus*). Proc. Nat. Acad. Sci. USA 97:11325-11330.

Yoder, A. D., and Z. Yang. 2000. Estimation of primate speciation dates using local molecular clocks. Mol. Biol. Evol. 17:1081-1090.

Zelus, D., M. R. Robinson-Rechavi, M. Delacre, C. Auriault, and V. Laudet. 2000. Fast Evolution of Interleukin-2 in Mammals and Positive Selection in Ruminants. J. Mol. Evol. 51:234-244.

Zhang, Y.-P., and O. A. Ryder. 1998. Mitochondrial Cytochrome b Gene Sequences of Old World Monkeys: With Special Reference on Evolution of Asian Colobines. Primates 39:39-49.

Zhou, Y.-H., D. Hewett-Emmet, J. P. Ward, and W.-H. Li. 1997. Unexpected Conservation of the X-Linked Color Vision Gene in Nocturnal Prosimians: Evidence from Two Bush Babies. J. Mol. Evol. 45:610-618.

Zietkiewicz, E., C. Richer, and D. Labuda. 1999. Phylogenetic affinities of tarsier in the context of primate Alu repeats. Mol. Phylogenet. Evol. 11:77-83.

Zietkiewicz, E., C. Richer, D. Sinnett, and D. Labuda. 1998. Monophyletic Origin of Alu Elements in Primates. J. Mol. Evol. 47:172-182.

TABLE V-1 PUBLISHED ESTIMATES OF DIVERGENCE DATES

1. Apes-Old World monkeys; 2. *Homo-Pan*; 3. (*Homo*, *Pan*)-*Gorilla*; 4. ((*Homo*,*Pan*),*Gorilla*)-*Pongo*; 5. Great apes-Gibbons; 6. Old World Monkeys-New World Monkeys; 7. Root; 8. Lemurs-Lorisiforms; 9. Colobinae-Cercopithecinae. All ages are in millions of years ago. Adapted from Vos & Mooers (2004).

Source	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Nei and Glazko (2002)	23	6	7			33			
Stauffer *et al.* (2001)	23	5.4	6.4	11	15				
Gingerich and Uhen (1994)							63		
Yoder (1997)								54	
Arnason *et al.* (1998)	50					60	80		30
Porter *et al.* (1997)	25								
Goodman *et al.* (1998)						38			
Adachi and Hasegawa (1995)		4		16					
Easteal and Herbert (1997)				8.5					
Kumar and Hedges (1998)	23.3	5.5	6.7	8.2	14.6	47.6			
Arnason *et al.* (1996b)		6.1							
Adachi and Hasegawa (1996)		4.3							
Arnason *et al.* (2000)		13	16	30	35	70			
Arnason *et al.* (1996a)		10.4	14.2	19.2	32.4				
Purvis (1995)	27.5	7.0	8.3	14.5	18.2	40.5	57.5	45.1	14.4
Median of published studies	24.1	6	7.6	14.5	18.2	44.0	63	49.5	22.2

TABLE V-2 LOCI USED TO INFER DIVERGENCE DATES

See text for details.

Locus	Number of sequences	Pruned
alpha13galacto	18	0
ATP7A	7	0
BRCA1	7	0
calmodulin	6	0
CCR5	59	3
CD4	17	0
COII	44	5
CXCR4	38	3
cytba	17	63
cytbb	8	47
DRD4	14	0
fut1	26	6
G6PD	22	0
gamma1globin	13	0
IL10	9	0
IL16	7	0
IL2	10	1
IL3	4	0
IL4	8	0
interferongamma	13	0
IRBP.intron1	23	12
IRBP.partial	23	0
LZM	16	1
ND1	12	0
ND2	12	1
ND3	34	0
ND4b	32	1
ND4L	14	26
ND5	26	1
ND6	12	0
NRAMP1	14	0
PLCB4	7	0
PNOC	7	0
SRY	56	1
trnaala	10	0
trnaarg	35	0
trnaasn	10	0
trnaasp	10	0
trnacys	10	0
trnagln	9	0
trnaglu	10	0
trnagly	29	0
trnaile	10	0
trnalys	10	0
trnaphe	6	0

Locus	Number of sequences	Pruned
trnapro	11	0
trnathr	12	0
trnatrp	10	0
trnatyr	10	0
trnaval	26	0
TSPY	38	0
vwf	17	0
ZFXa	9	0
ZFXb	13	0
ZFY	10	3

TABLE V-3 DIVERGENCE DATES AND SUPPORT FOR NODES

Ages labeled "a" are interpolated from a pure birth model, and those labeled "b" are estimated from molecular data.

Node	Age	Stderr	Clade size	Bremer support	rQS index	Matching sources	Mismatching sources	Equivocal sources
1	3.314008a	n/a	2	4	0.013	5	0	369
2	3.272367a	n/a	2	6	0.008	6	3	365
3	7.881723a	n/a	4	1	0.003	6	5	363
4	1.377923a	n/a	2	1	0.003	1	0	373
5	3.033345a	n/a	3	1	0.016	7	1	366
6	5.071805a	n/a	4	0	0.013	7	2	365
7	1.960615a	n/a	2	0	0	2	2	370
8	4.383235a	n/a	3	0	0	3	3	368
9	7.676796a	n/a	7	0	-0.019	1	8	365
10	9.818911a	n/a	8	0	-0.021	1	9	364
11	12.68218a	1.235	12	5	0.051	19	0	355
12	1.43E-14b	0.000	2	1	0.003	1	0	373
13	7.682266a	n/a	3	7	0.013	5	0	369
14	15.65228a	3.930	15	7	-0.011	23	27	324
15	1.43E-14b	0.000	2	1	0	1	1	372
16	1.43E-14b	0.000	2	1	0	1	1	372
17	0.511943b	0.338	4	2	0.027	11	1	362
18	3.429579a	n/a	2	1	0.003	1	0	373
19	11.20397b	1.729	6	2	0.029	12	1	361
20	2.071066b	0.366	2	9	0.019	7	0	367
21	3.713118b	0.657	3	3	0.045	19	2	353
22	11.3592a	1.758	9	35	0.112	44	2	328
23	18.37584b	2.947	10	3	0.045	33	16	325
24	18.99167a	5.096	25	14	0.16	62	2	310
25	1.231955a	n/a	2	0	0	0	0	374
26	2.62541a	0.070	3	0	0	1	1	372
27	4.155122a	n/a	4	0	0.003	1	0	373
28	6.283654a	n/a	5	0	0.003	1	0	373
29	8.675736a	1.124	6	0	0.019	8	1	365
30	4.522607a	n/a	2	1	0.003	1	0	373
31	11.75498a	1.124	8	0	0.016	7	1	366
32	15.78918a	0.400	9	8	0.029	11	0	363
33	22.03249b	3.357	34	9	0.07	51	25	298
34	1.758327a	n/a	2	0	0	1	1	372
35	3.946727a	n/a	3	0	0.013	9	4	361
36	6.829616a	n/a	4	0	0.013	9	4	361
37	8.429973b	3.031	5	6	0.07	26	0	348
38	0.138835b	0.138	2	6	0.013	5	0	369
39	3.9671a	n/a	2	3	0.008	4	1	369
40	7.267568b	n/a	4	10	0.04	15	0	359
41	23.1875b	3.104	9	25	0.086	43	11	320
42	25.68486a	5.942	43	11	0.187	91	21	262
43	2.913174a	n/a	2	2	0.003	1	0	373
44	9.031937b	0.756	3	2	0.045	24	7	343
45	1.209202b	0.317	2	5	0.005	3	1	370
46	2.046641b	0.139	3	1	0.019	9	2	363

Node	Age	Stderr	Clade size	Bremer support	rQS index	Matching sources	Mismatching sources	Equivocal sources
47	2.890899a	n/a	2	2	0	1	1	372
48	4.700399b	0.319	6	5	0.043	16	0	358
49	11.29623b	1.060	9	11	0.102	40	2	332
50	1.611309a	n/a	2	0	0.029	11	0	363
51	3.417882a	n/a	3	0	0.029	11	0	363
52	5.710881a	n/a	4	0	0.029	11	0	363
53	6.920234b	1.338	5	0	0.037	14	0	360
54	12.73689a	0.267	6	4	0.043	16	0	358
55	20.68827b	2.723	15	27	0.12	46	1	327
56	2.327182a	n/a	2	2	0.005	2	0	372
57	2.357248a	n/a	2	1	0.003	1	0	373
58	5.330026a	n/a	4	2	0.008	3	0	371
59	1.991047a	n/a	2	2	0.003	1	0	373
60	4.727224a	n/a	3	0	0	1	1	372
61	7.993336a	n/a	7	0	0	1	1	372
62	10.29525a	n/a	8	0	0	1	1	372
63	3.080813a	n/a	3	0	0	1	1	372
64	7.327279a	n/a	4	0	0	1	1	372
65	13.3041a	n/a	12	1	0.003	2	1	371
66	16.47941a	n/a	13	6	0.051	19	0	355
67	1.764904a	n/a	2	0	0.003	1	0	373
68	3.780603a	n/a	3	0	0.003	1	0	373
69	6.551048a	n/a	4	1	0.005	2	0	372
70	10.24107a	n/a	5	2	0.011	4	0	370
71	3.709371a	n/a	2	3	0.005	2	0	372
72	3.679223a	n/a	2	1	0.003	1	0	373
73	8.124424b	n/a	4	8	0.056	22	1	351
74	11.8862b	n/a	9	20	0.094	35	0	339
75	20.96835a	3.493	22	10	0.061	33	10	331
76	25.31997a	6.891	37	12	0.048	39	21	314
77	30.15532b	4.441	80	45	0.369	138	0	236
78	0.792181b	0.085	2	1	0.003	2	1	371
79	1.804391a	n/a	3	8	0.016	6	0	368
80	4.016289a	0.941	4	4	-0.016	1	7	366
81	3.105567b	0.569	2	1	0	3	3	368
82	3.520138b	0.349	3	3	-0.003	3	4	367
83	0.888868a	n/a	2	2	-0.013	0	5	369
84	3.548143b	0.681	5	3	0.011	6	2	366
85	3.548143b	0.681	6	8	0.019	8	1	365
86	4.016289a	1.950	7	3	0.008	7	4	363
87	5.878584b	0.542	11	7	0.051	19	0	355
88	1.961341b	0.144	2	47	0.11	44	3	327
89	5.069091b	0.415	3	49	0.139	65	13	296
90	6.413876b	0.321	4	79	0.238	92	3	279
91	15.84305b	0.960	5	9	0.259	104	7	263
92	19.63822b	1.300	16	16	0.369	138	0	236
93	5.179923b	1.068	2	1	-0.008	5	8	361
94	0.96245a	1.106	2	6	0.013	5	0	369
95	0.760628a	n/a	2	1	0.003	1	0	373
96	0.490956a	1.106	2	0	0.008	3	0	371
97	1.040339a	n/a	3	1	0.011	4	0	370
98	1.704561a	n/a	5	2	0.011	5	1	368

Node	Age	Stderr	Clade size	Bremer support	rQS index	Matching sources	Mismatching sources	Equivocal sources
99	2.332285a	0.716	7	1	0.008	4	1	369
100	1.23689a	n/a	2	1	0.003	1	0	373
101	2.956938a	1.216	9	1	-0.011	1	5	368
102	0.769316a	n/a	2	3	0.008	3	0	371
103	0.789489a	n/a	2	4	0.011	4	0	370
104	1.24921b	0.296	4	5	0.008	3	0	371
105	2.564199a	0.790	5	1	-0.008	1	4	369
106	3.637249a	0.840	14	0	-0.003	2	3	369
107	4.22004a	n/a	15	2	-0.003	2	3	369
108	8.59E-05b	0.000	2	1	0.003	2	1	371
109	0.839612b	0.091	3	5	0.003	2	1	371
110	4.920964a	0.938	18	2	0.008	5	2	367
111	5.724032a	0.888	20	1	0.019	14	7	353
112	6.751236a	1.021	21	4	0.024	14	5	355
113	8.292042b	1.498	22	2	0.045	20	3	351
114	0.27622b	0.179	2	4	0.003	3	2	369
115	0.908084b	0.154	3	5	0.016	8	2	364
116	2.221603b	0.443	4	5	0.016	9	3	362
117	1.047513b	0.128	2	0	-0.003	1	2	371
118	0.70926a	0.324	2	0	-0.005	1	3	370
119	1.709499a	0.499	4	2	0	2	2	370
120	2.800668b	0.486	5	4	0.008	5	2	367
121	4.433933b	0.342	9	4	0.016	8	2	364
122	2.250415b	0.919	2	2	0	2	2	370
123	1.645312b	0.320	2	1	0.003	1	0	373
124	0.13504b	0.069	2	1	0.003	1	0	373
125	2.654026b	0.458	4	5	0.011	4	0	370
126	3.582641b	0.564	6	5	0.008	5	2	367
127	6.259739a	0.491	15	1	0.035	16	3	355
128	7.483745a	5.047	16	8	0.056	22	1	351
129	1.547503b	0.176	2	0	0.008	9	6	359
130	2.233375b	0.299	3	7	0.027	15	5	354
131	2.313178b	0.471	2	0	0	3	3	368
132	1.958261b	0.245	2	8	0.032	12	0	362
133	5.443154b	0.969	4	13	0.035	14	1	359
134	6.714332b	0.914	7	11	0.032	17	5	352
135	9.23115a	0.564	23	7	0.118	46	2	326
136	11.25912b	1.145	45	6	0.176	67	1	306
137	0.800086b	0.086	2	2	0.005	2	0	372
138	3.486339a	1.296	3	1	0.008	3	0	371
139	6.030347a	n/a	4	2	0.011	4	0	370
140	2.237741a	n/a	2	1	0.003	1	0	373
141	5.256468a	n/a	3	2	0.005	2	0	372
142	11.95993b	2.295	7	3	0.013	5	0	369
143	3.48278a	n/a	2	4	0.011	4	0	370
144	1.035514a	n/a	2	1	0.003	1	0	373
145	2.604488b	0.192	3	1	0.008	3	0	371
146	3.873409b	0.510	4	5	0.016	6	0	368
147	8.001457b	0.708	5	2	0.011	6	2	366
148	1.595359a	n/a	4	1	0.003	1	0	373
149	3.804516a	n/a	5	0	0.003	2	1	371
150	6.932486a	0.869	2	0	0	1	1	372

Node	Age	Stderr	Clade size	Bremer support	rQS index	Matching sources	Mismatching sources	Equivocal sources
151	6.932486a	0.577	3	0	0	3	3	368
152	0.696042a	n/a	2	0	0.003	1	0	373
153	0.801338b	0.124	4	0	0.003	1	0	373
154	2.900798b	0.243	5	0	0.019	7	0	367
155	3.735895a	n/a	6	0	0.021	8	0	366
156	5.132287a	0.000	7	0	0.021	8	0	366
157	6.932486a	0.803	15	1	0.024	10	1	363
158	8.877896a	0.922	20	2	0.037	15	1	358
159	8.896662b	2.159	22	6	0.056	21	0	353
160	13.60246b	1.113	29	10	0.072	27	0	347
161	18.33012b	1.085	74	27	0.243	92	1	281
162	32.41041b	1.507	90	18	0.529	200	2	172
163	51.23347b	3.875	170	11	0.794	297	0	77
164	16.07951b	1.456	2	1	0.013	5	0	369
165	15.98969a	n/a	2	1	0.003	1	0	373
166	38.21588a	n/a	4	3	0.016	6	0	368
167	62.609b	3.352	174	9	0.781	297	5	72
168	9.407427b	0.464	2	2	0.005	2	0	372
169	2.729257a	0.264	2	1	0.005	2	0	372
170	4.093885a	0.674	3	2	0.013	7	2	365
171	7.061984a	0.748	4	2	0.013	7	2	365
172	11.46329a	0.860	6	1	0.043	16	0	358
173	17.02967a	n/a	7	7	0.048	18	0	356
174	1.534145b	0.417	2	1	0.003	2	1	371
175	5.872105b	0.230	3	3	0.005	4	2	368
176	7.439765a	n/a	4	3	0.003	6	5	363
177	13.61071b	0.562	5	13	0.027	13	3	358
178	3.677795b	0.156	2	5	-0.019	5	12	357
179	3.754391b	0.185	3	3	-0.035	2	15	357
180	3.066328b	0.151	2	5	-0.016	4	10	360
181	6.369492a	0.156	5	25	0.048	18	0	356
182	11.89144b	1.176	6	3	-0.019	8	15	351
183	5.667141b	0.530	2	1	0	1	1	372
184	5.667141b	0.530	3	4	0.013	6	1	367
185	6.232139b	0.640	4	7	0.016	14	8	352
186	13.05581a	0.443	10	10	0.07	29	3	342
187	1.366414a	n/a	2	1	0.003	1	0	373
188	3.002269a	n/a	3	1	0.003	1	0	373
189	5.022715a	n/a	4	1	0.003	1	0	373
190	7.591075a	n/a	5	1	0.003	1	0	373
191	11.20404a	n/a	6	2	0.008	3	0	371
192	17.1708a	n/a	16	2	0.053	27	7	340
193	22.04095b	0.884	21	6	0.045	28	11	335
194	27.51953b	1.122	28	14	0.115	52	9	313
195	39.47811b	1.720	29	26	0.131	56	7	311
196	2.672337a	n/a	2	3	0.003	1	0	373
197	9.194181b	1.217	3	4	0.021	11	3	360
198	4.698149a	n/a	2	2	0	4	4	366
199	16.34787b	2.164	5	4	0.037	18	4	352
200	2.371579a	n/a	3	2	0.016	7	1	366
201	5.397494a	n/a	4	1	-0.019	0	7	367
202	4.240976b	0.561	2	3	0.008	5	2	367

Node	Age	Stderr	Clade size	Bremer support	rQS index	Matching sources	Mismatching sources	Equivocal sources
203	8.517026b	1.127	6	1	0.005	6	4	364
204	3.574516a	n/a	2	2	-0.005	2	4	368
205	3.6487a	n/a	2	1	0	1	1	372
206	8.517089b	1.127	4	1	-0.013	1	6	367
207	8.517089b	1.127	10	2	0.035	13	0	361
208	20.91996b	1.286	15	18	0.102	42	4	328
209	51.86696b	4.207	44	9	0.249	96	3	275
210	65.09789b	4.324	218	n/a	1	374	0	0

TABLE V-4 COMPARISON BETWEEN PRIMATE SUPERTREE ANALYSES

	Purvis (1995)	**Present study**
Number of taxa	203	218
Resolution	79.21	85.32
Resolution of common taxon subset	78.95	86.84
Matching clades	112/150	112/165

TABLE V-5 RATES OF CLADOGENESIS FOR MAJOR CLADES, ALL NODES

clade	b-d (s.e.)
Apes	0.076 (0.061)
Atelidae	0.10 (0.058)
Cebidae	0.12 (0.013)
Cercopithecinae	0.29 (0.032)
Colobinae	0.19 (0.026)
Galaginae	0.11 (0.021)
Lemurinae	0.086 (0.012)
New world monkeys	0.12 (0.0098)
Old world monkeys	0.24 (0.020)
Pithecidae	0.12 (0.019)
Strepsirhini	0.076 (0.021)

FIGURE V-1 COMPARISON OF EXPECTED DIVERGENCE DATES.

Comparison of expected divergence dates calculated using the natural log of clade sizes (diamonds) and divergence dates calculated using a pure birth model (boxes). The approach based on the natural log of clade sizes results in *increasing* waiting times as cladogenesis proceeds and in differing slopes of lineage-through-time plots depending on the tree shape (solid diamonds: ladder shaped tree; open diamonds: fully balanced trees), whereas the pure birth approach does neither.

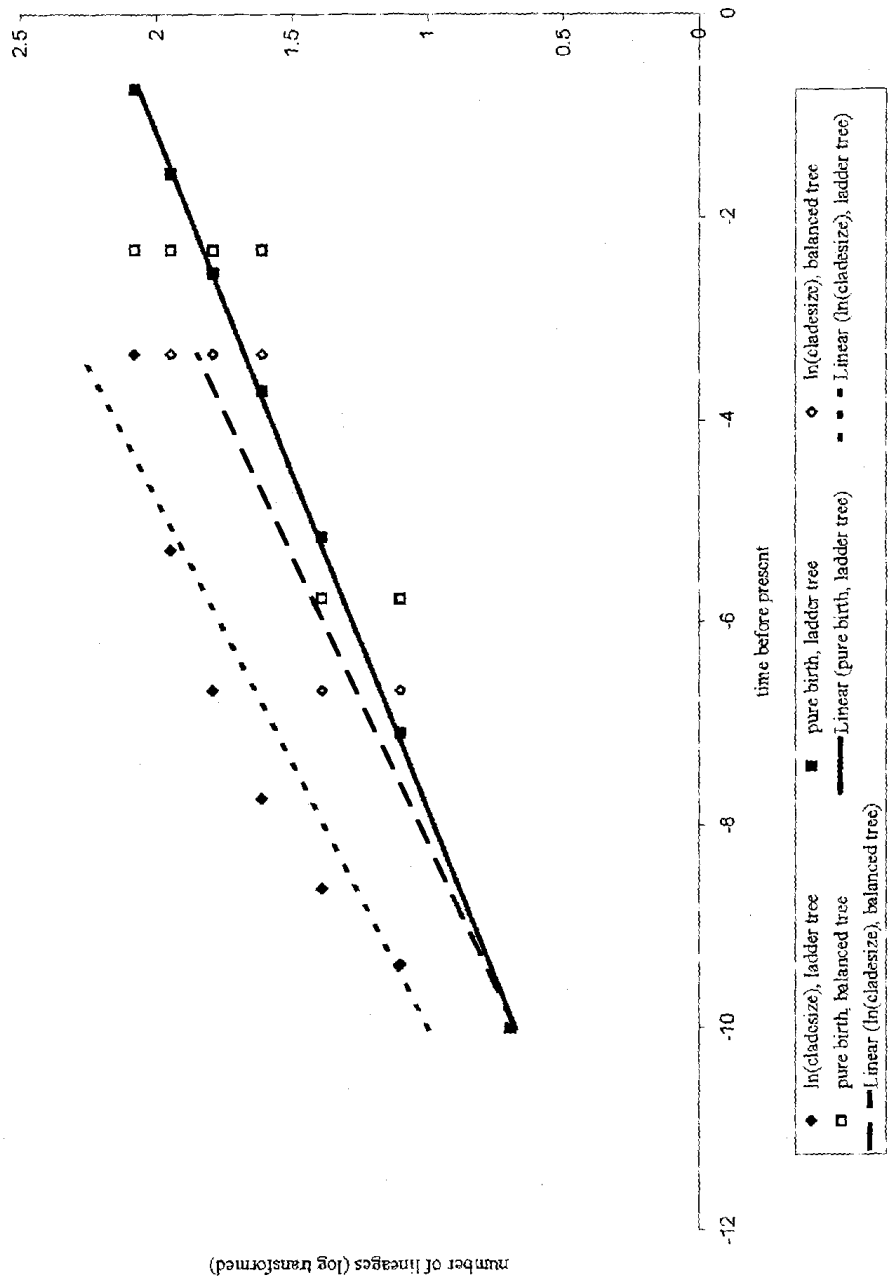


FIGURE V-2 SUPERTREE BACKBONE TOPOLOGY.

The triangular tips are expanded in the subsequent figures. The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. See text for details.

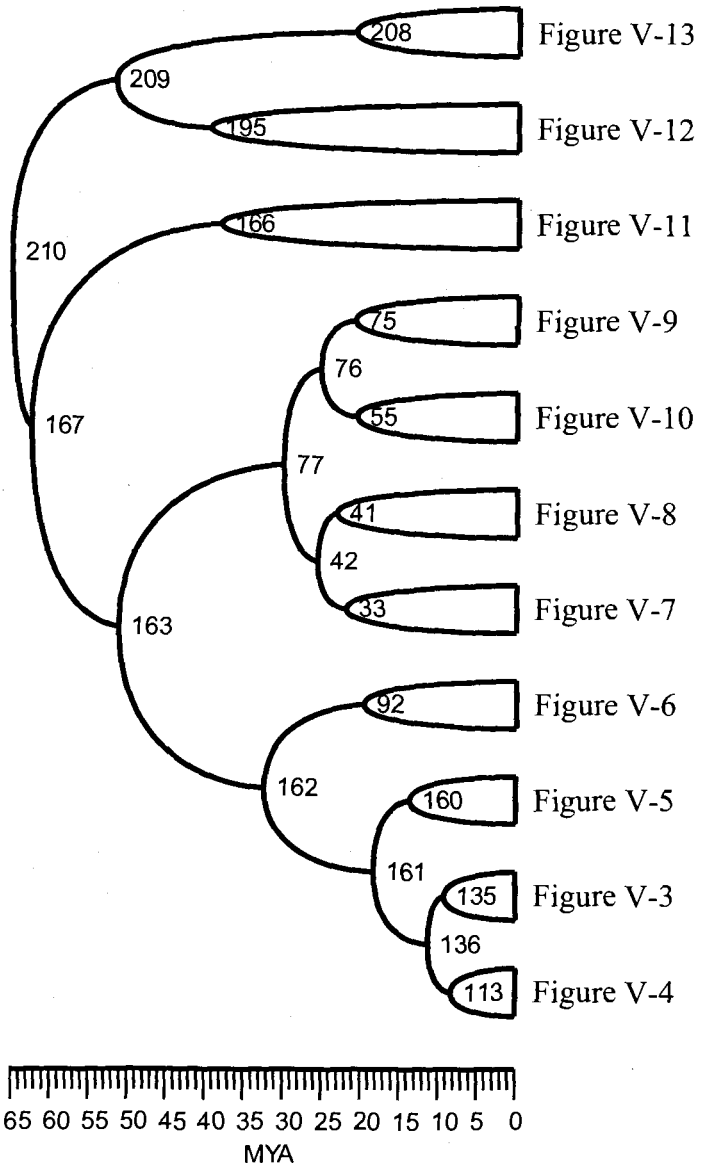


FIGURE V-3 *MACACA, CERCOCEBUS, MANDRILLUS, PAPIO, THEROPITHECUS, LOPHOCEBUS*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. Dotted branches are collapsed in the strict consensus. See text for details.

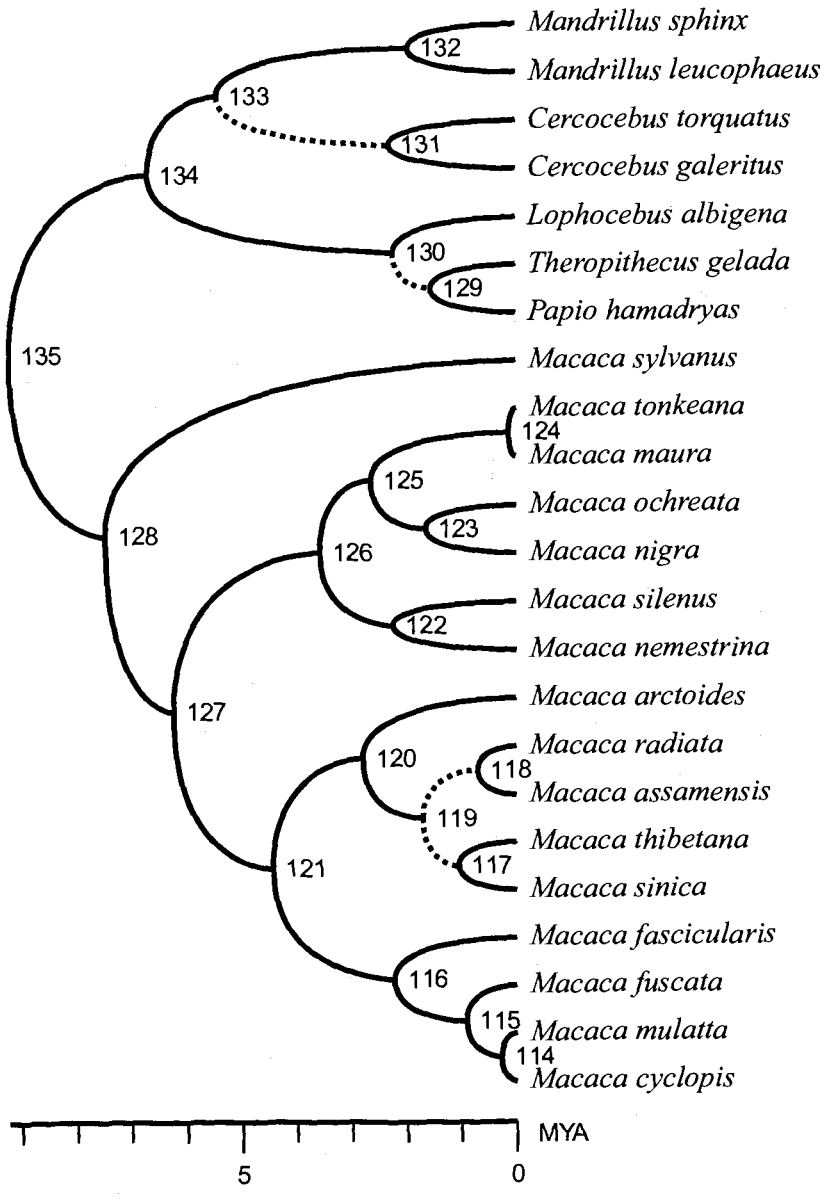


FIGURE V-4 *CERCOPITHECUS, CHLOROCEBUS, ERYTHROCEBUS, MIOPITHECUS, ALLENOPITHECUS*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. Dotted branches are collapsed in the strict consensus. See text for details.

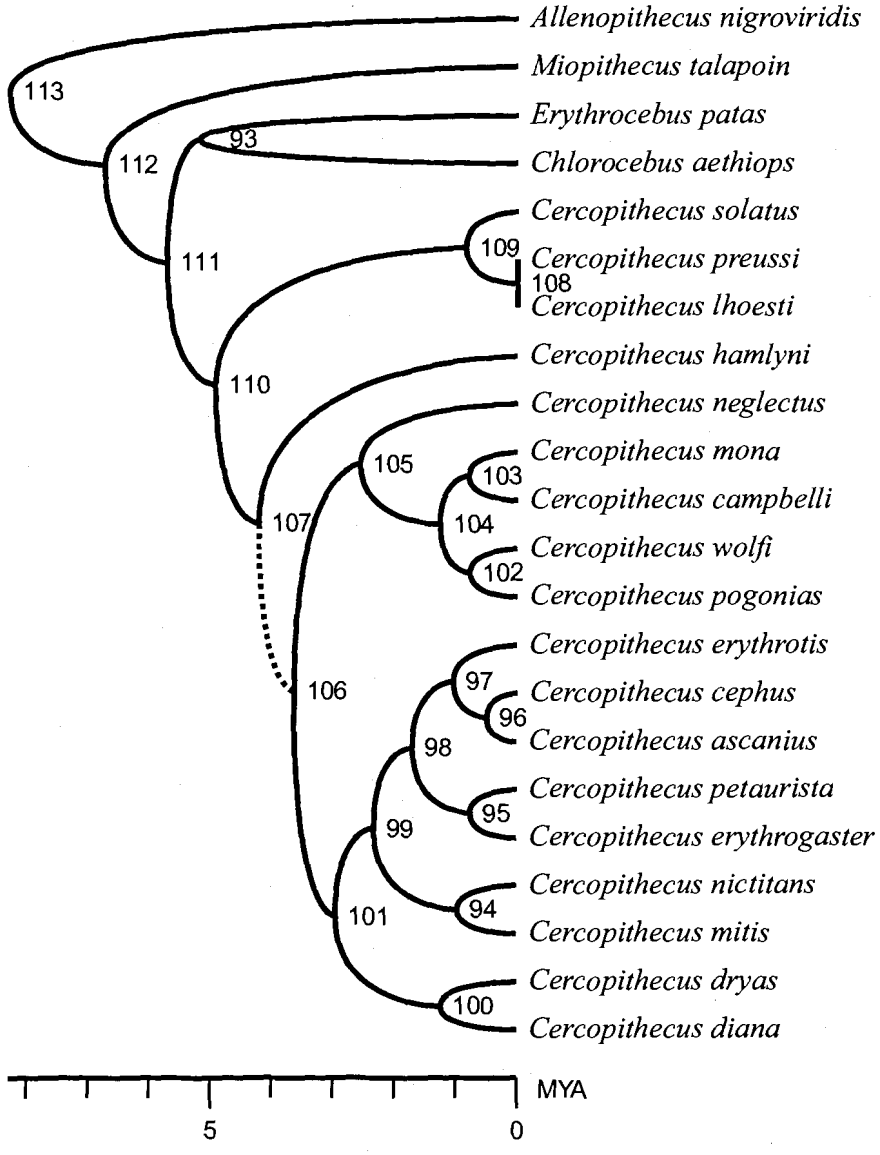


FIGURE V-5 *TRACHYPITHECUS, PRESBYTIS, SEMNOPITHECUS, PYGATHRIX, NASALIS, COLOBUS, PROCOLOBUS*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. Dotted branches are collapsed in the strict consensus. See text for details.

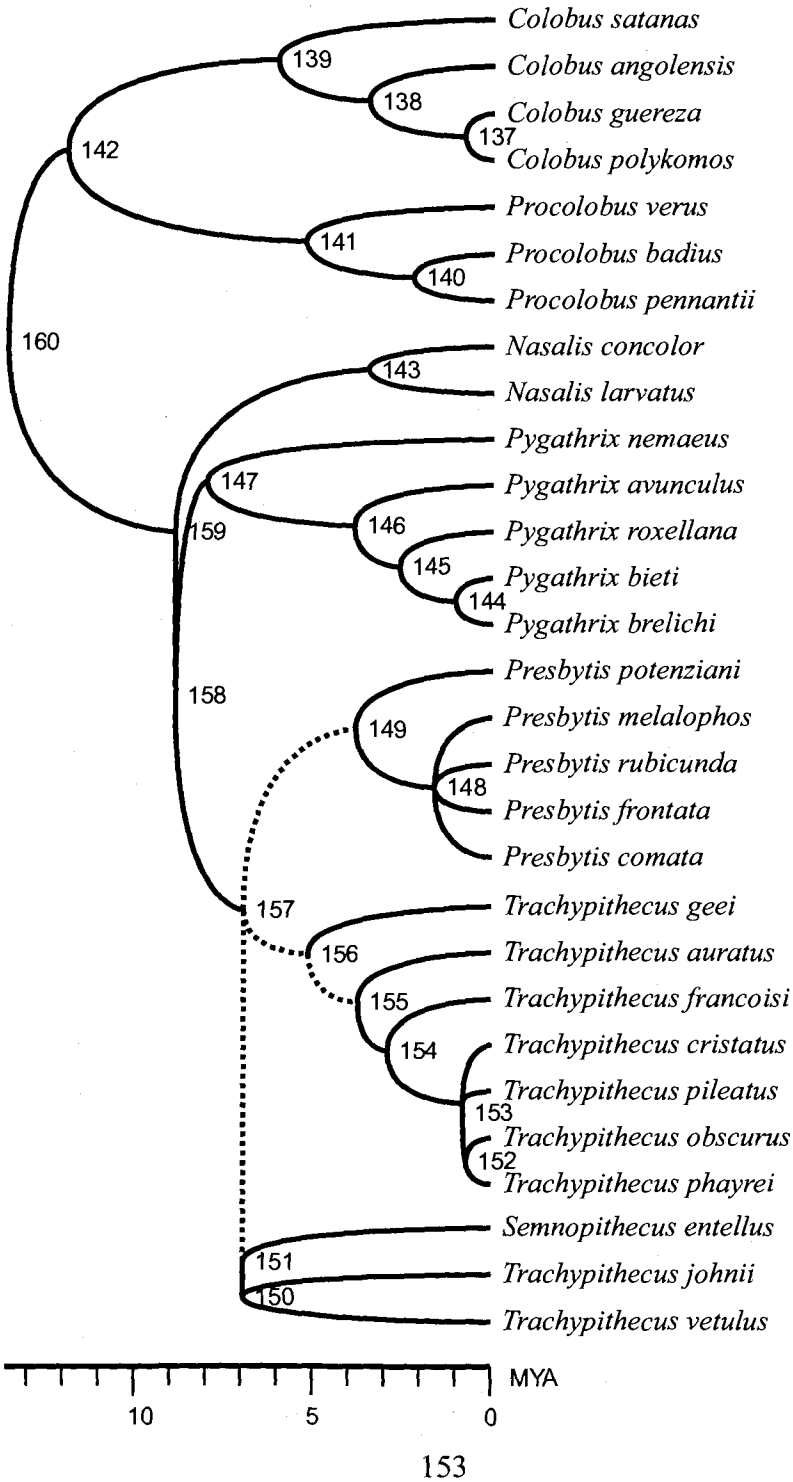


FIGURE V-6 *HYLOBATES, PAN, HOMO, GORILLA. PONGO*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. See text for details.

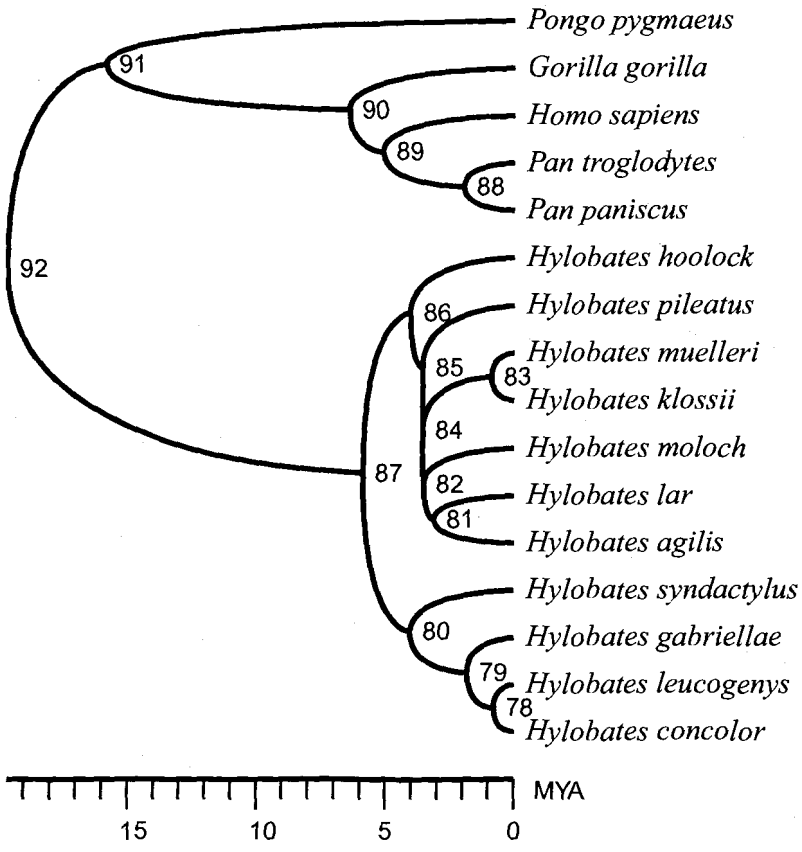


FIGURE V-7 *SAGUINUS, LEONTOPITHECUS, CALLITHRIX, CALLIMICO, AOTUS.*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. Dotted branches are collapsed in the strict consensus. See text for details.

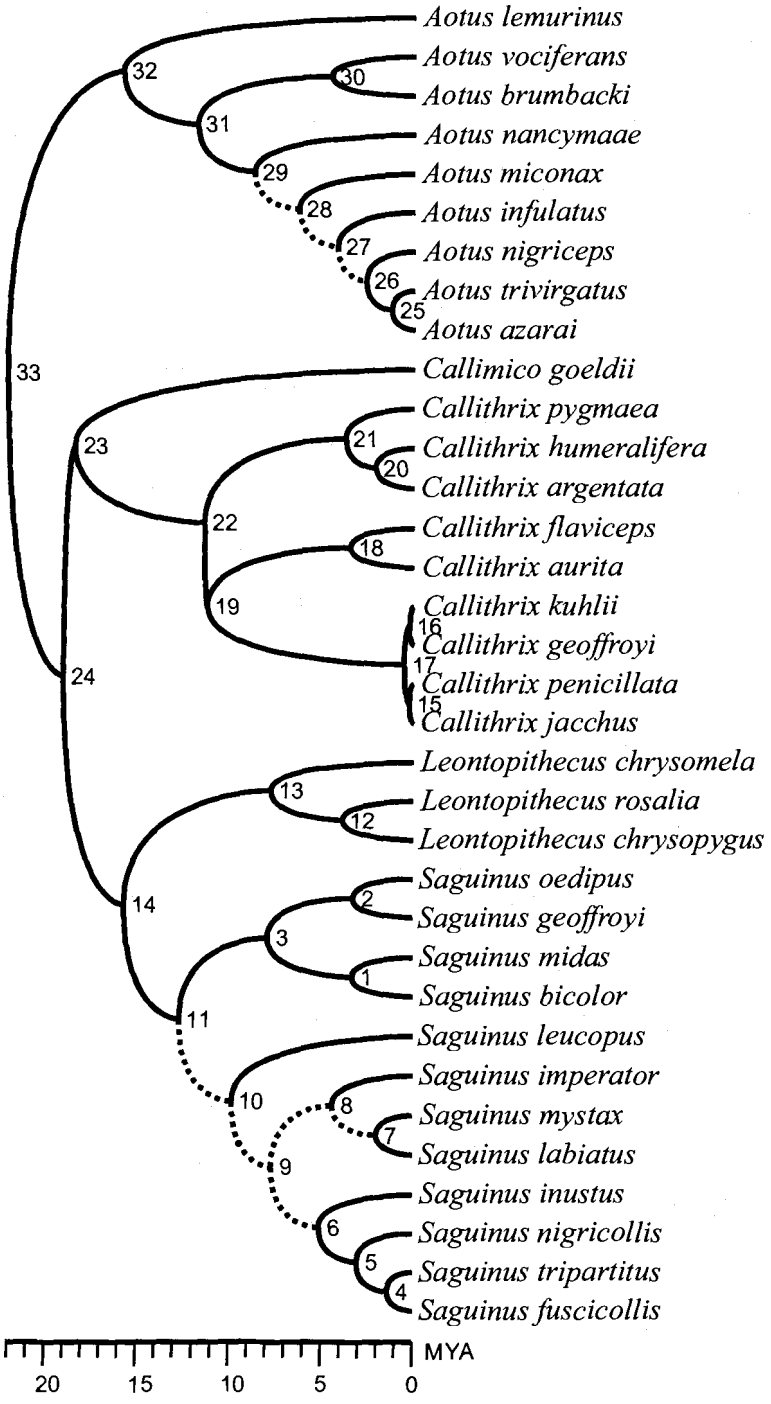


FIGURE V-8 *SAIMIRI, CEBUS*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. Dotted branches are collapsed in the strict consensus. See text for details.

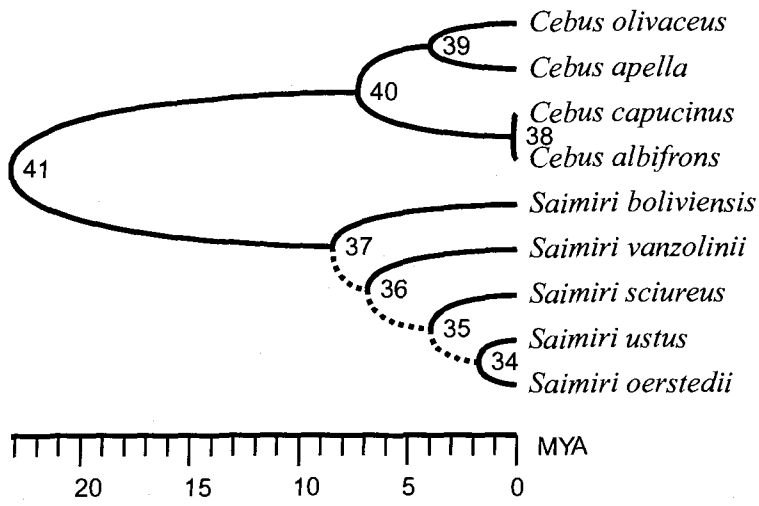


FIGURE V-9 *CALLICEBUS, PITHECIA, CACAJAO, CHIROPOTES*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. Dotted branches are collapsed in the strict consensus. See text for details.

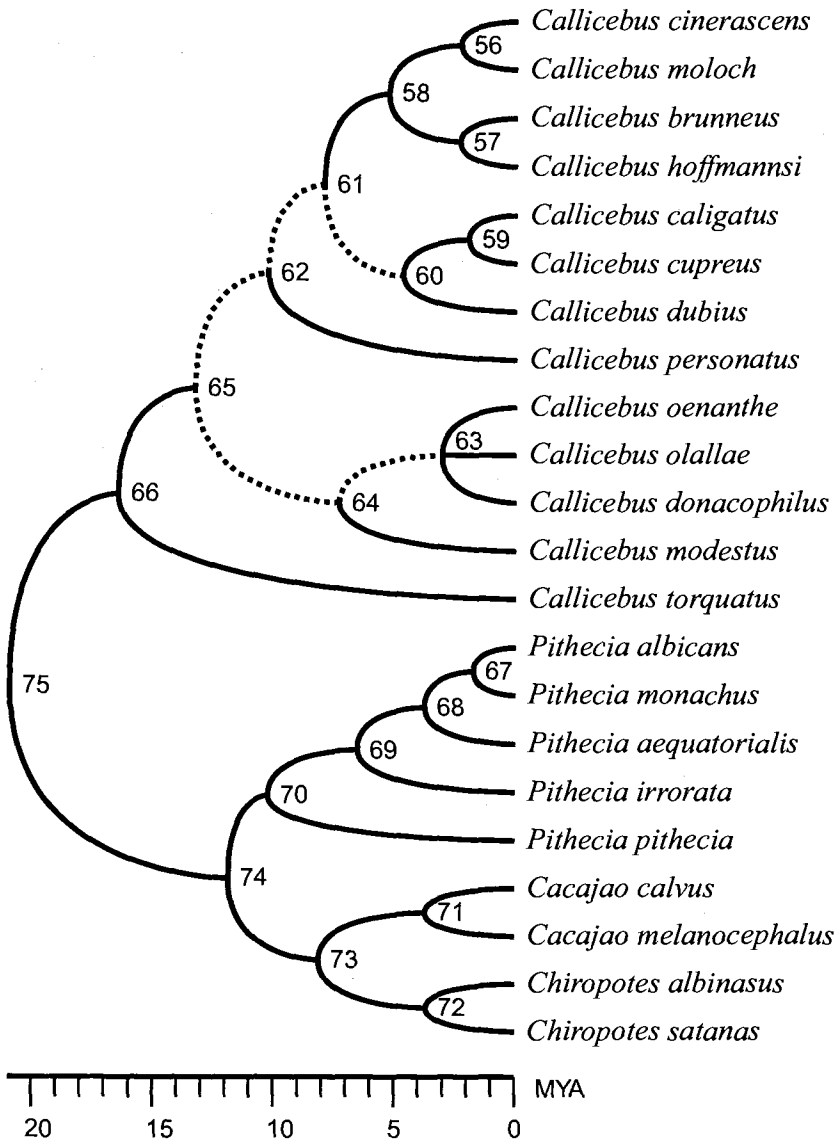


FIGURE V-10 *ATELES, LAGOTHRIX, BRACHYTELES, ALOUATTA*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. See text for details.

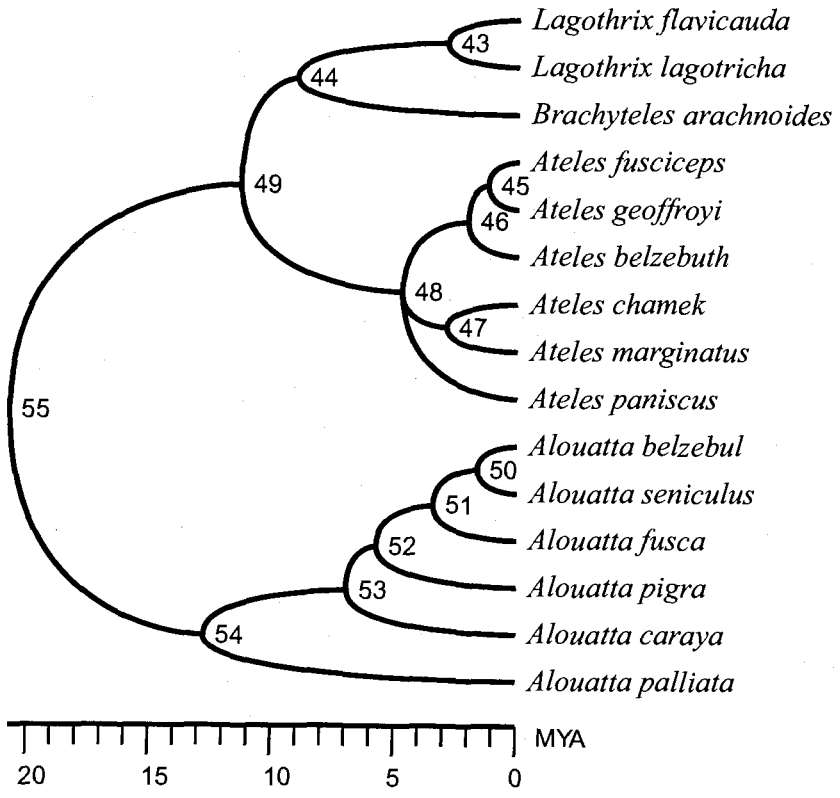


FIGURE V-11 *TARSIUS*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. See text for details.

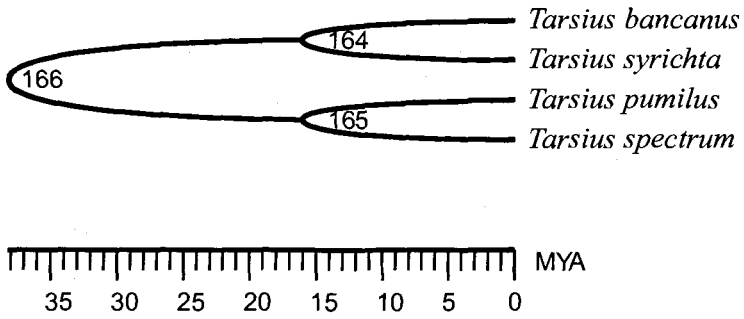


FIGURE V-12 *EULEMUR, VARECIA, HAPALEMUR, LEMUR, LEPILEMUR, PROPITHECUS, INDRI, AVAHI, MICROCEBUS, ALLOCEBUS, CHEIROGALEUS, PHANER, DAUBENTONIA*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. See text for details.

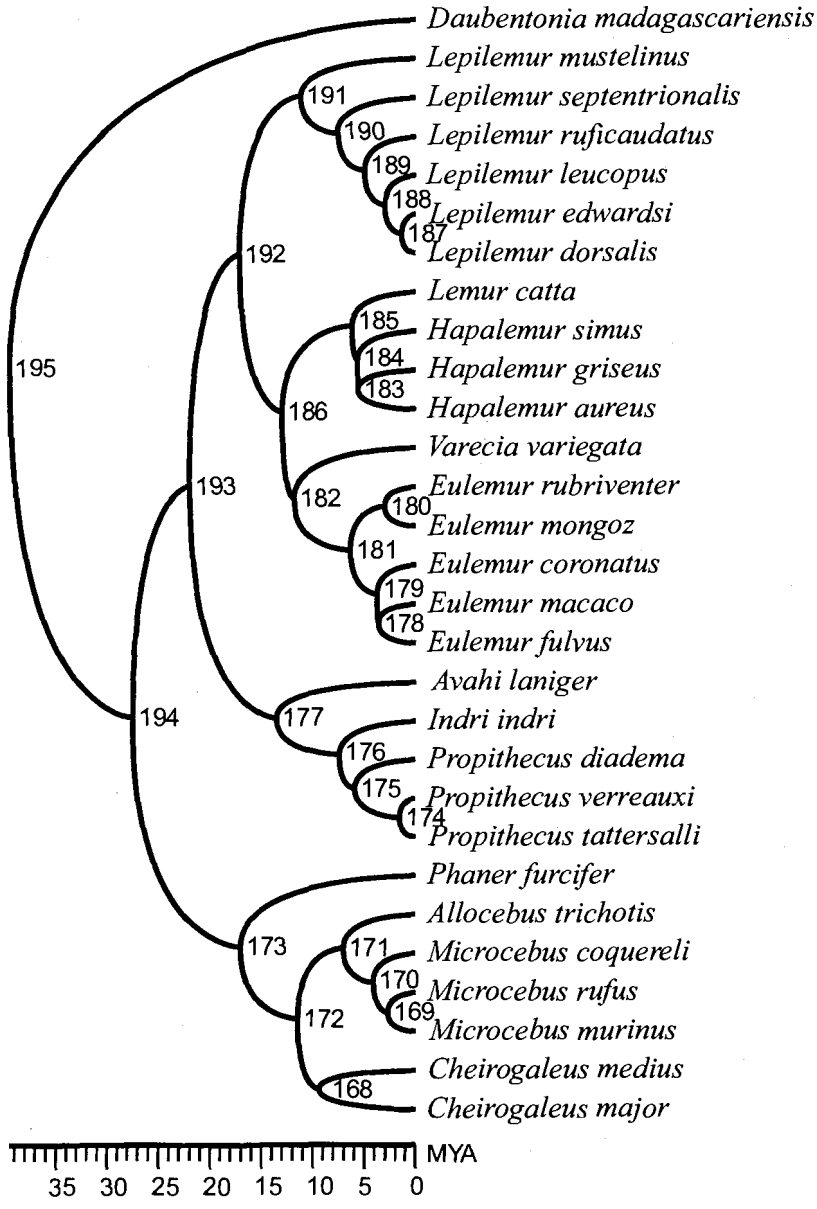


FIGURE V-13 *GALAGO, OTOLEMUR, GALAGOIDES, EUOTICUS, NYCTICEBUS, LORIS, ARCTOCEBUS, PERODICTICUS*

The node labels are unique identifiers over the whole tree, and correspond with the entries in Table V-3. See text for details.

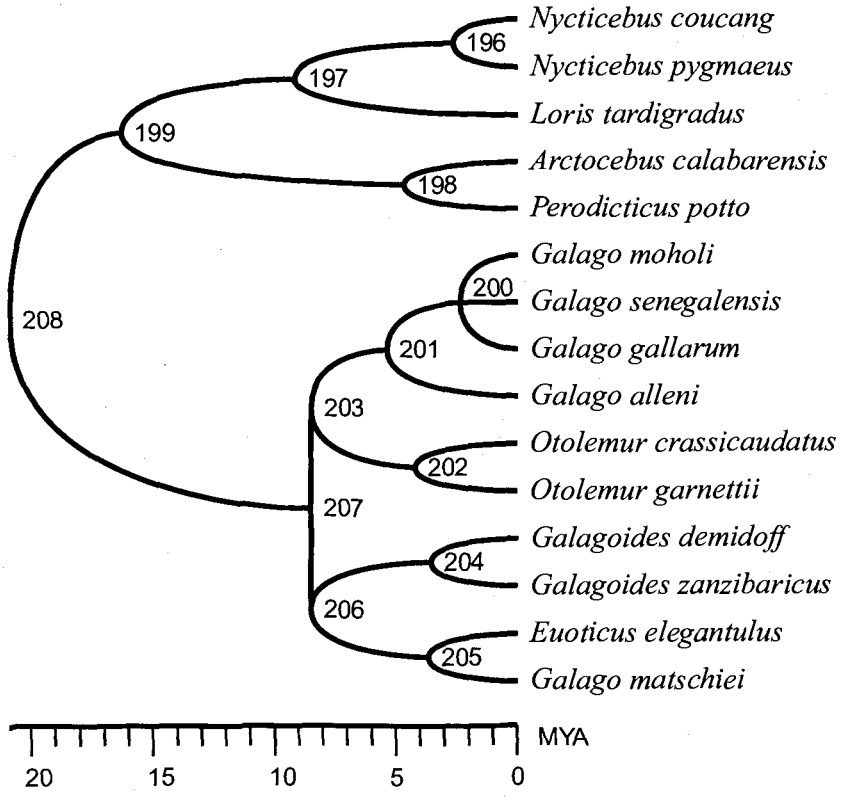


FIGURE V-14 NODE DEPTH DISTRIBUTIONS: MODELED VERSUS MOLECULAR ESTIMATES

Distribution of modeled node depths versus depths estimated from calibrated near-clocklike sequences. Diamond boxes indicate mean and 95% confidence interval. See text for details.

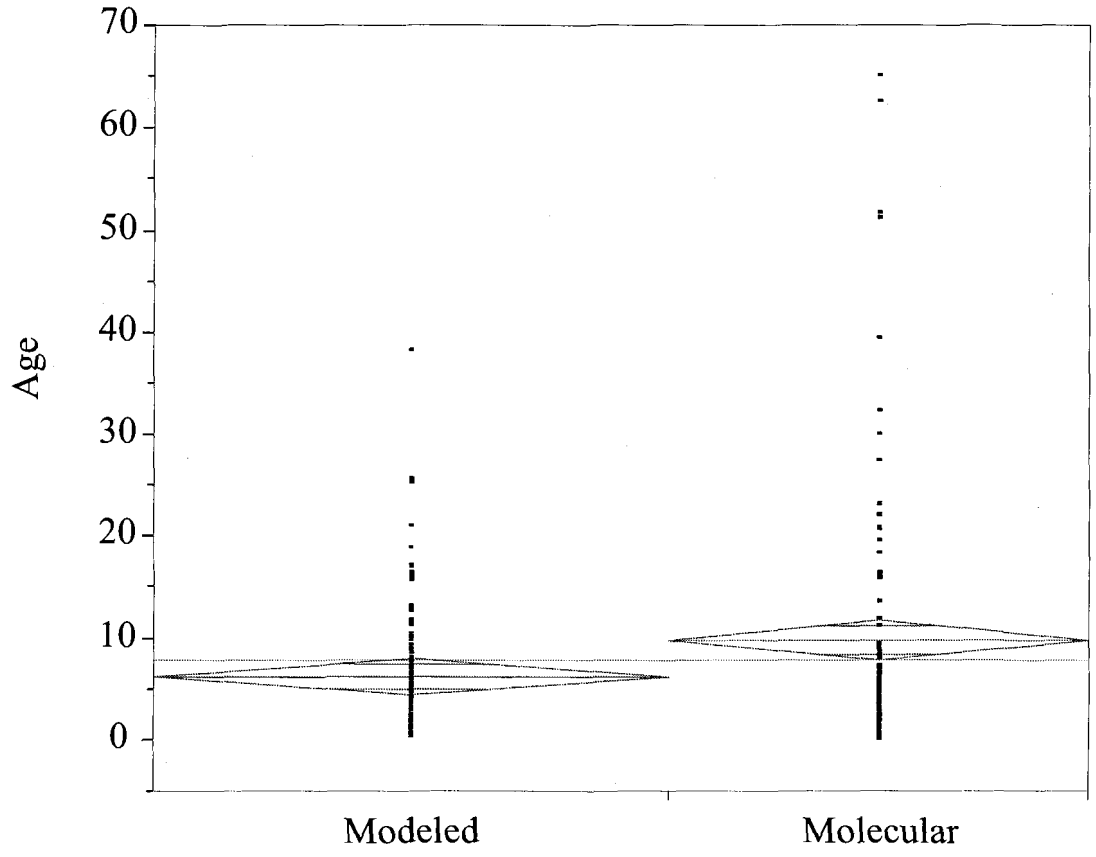
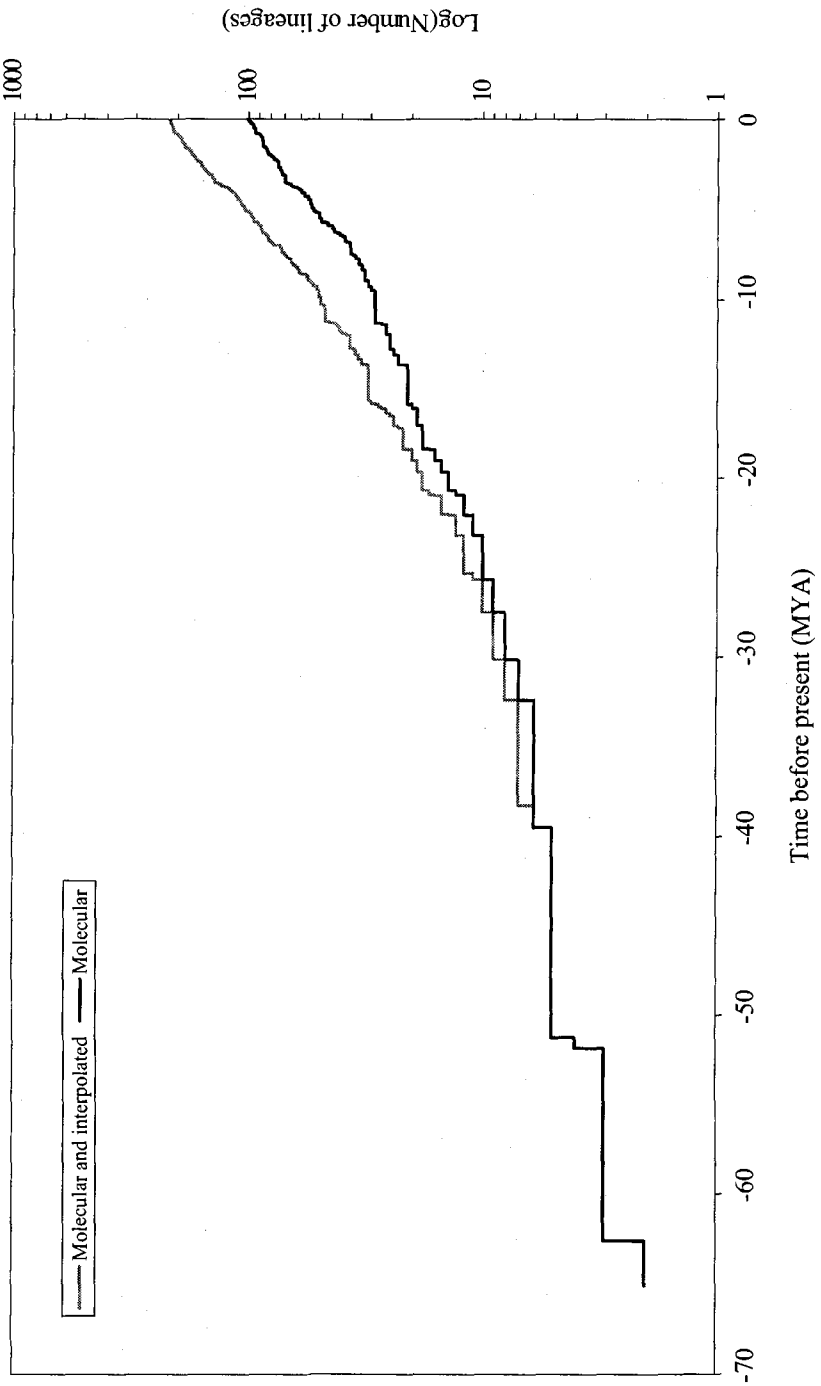


FIGURE V-15 LINEAGE-THROUGH-TIME PLOTS

Lineage through time plots for the order Primates (log transformed number of lineages). The gray curve shows lineage through time growth when both molecular estimates of divergence dates, as well as interpolated (pure birth) based divergence dates are considered. The black curve shows lineage through time curve only considering lineages for which molecular estimates are available. See text for details.



CHAPTER VI - DESIGN PATTERNS IN PHYLOGENETICS:

PRACTICAL TREE DATA STRUCTURES AND OBJECTS FOR SERIALIZATION

Rutger A. Vos

INTRODUCTION

Phylogenetic trees are graph-like structures describing how biological units – species, molecular sequences, higher-level taxa – are related. Phylogenies are of interest in a number of fields of biology, for example to aid in comparative studies (Felsenstein, 1985; Harvey and Pagel, 1991), for classification (Doolittle, 1999), for analyses of tree shape to infer macro-evolutionary processes (Guyer and Slowinski, 1991; Guyer and Slowinski, 1993; Slowinski and Guyer, 1989) and so on.

In a formal, graph theoretical context (see, e.g., Diestel, 2005), phylogenetic trees are connected, undirected, acyclic graphs. These graphs can be binary resolved ("*dyadic*") or unresolved ("*multiway*"). In a phylogenetic context, several meanings can be ascribed to an unresolved tree structure: it can mean that multiple, instantaneous speciation events are postulated, or that there is not enough evidence to resolve the structure further.

Trees are comprised of nodes ("*vertices*" with in-degree 0 or 1). In the case of rooted trees, one of these nodes is the root, i.e. the most recent common ancestor of all other nodes in the tree. Of the remaining nodes, some are internal to the tree; that is, these nodes – hypothetical ancestors – have descendants; and the rest is terminal ("*leafs*"). (Specifically, the number of interior nodes, including the root, is *n-1* for *n* leafs on a binary, rooted tree and *n-2* for a binary, unrooted tree.) The root node has no parents or siblings, only children; internal nodes have parents, siblings and children; terminal nodes have parents and siblings, but no children.

Nodes are connected, one to another, by branches ("*edges*"), which can have lengths ("*edge weights*") quantified as integer values or real numbers, or no value at all.

In a phylogenetic context, the branch length can have a variety of meanings: the number of inferred changes under a parsimony-based optimality criterion, the time between the parent split and the child split; a support value such as the bootstrap value of a node (Felsenstein, 1985), or its posterior probability (Hastings, 1970; Metropolis et al., 1953), and so on. Conceptually, phylogenetic trees are unordered; that is, there is no biological significance ascribed to the left-to-right order in which child nodes are connected to their parents. (See Figure VI-1)

Phylogenetic trees are not what computer scientists refer to as 'search trees' – their balance, i.e. the degree to which the number of descendants of a left subtree approaches that of a right subtree, for all splits in a tree (the symmetry of the tree), is a result of evolutionary processes and not open to modification for the purpose of improved traversal efficiency, unlike, for example, B-trees (Bayer, 1971; Bayer and McCreight, 1972).

In recent years, many researchers have developed software for phylogenetic analysis (for an attempt at a comprehensive list of programs, see: http://evolution.genetics.washington.edu/phylip/software.html). These programs are written in a number of programming languages, such as C/C++, Java, and dynamic languages such as Python and Perl. Although many programs were written to address only a small subset of the phylogenetics problem space – such as certain kinds of calculations – the authors often face the same implementation problem: the choice of a suitable internal representation of phylogenetic tree shape. The scope of this paper is to review possible solutions to this problem, with particular reference to their suitability for

transferring large trees across programming language boundaries and to persistent storage media.

DATA STRUCTURES

Programming languages discern different types of variables, such as integers, real numbers and single characters. These *primitive* types can be organized into *complex* types which can be used to represent tree shape.

INTEGER ARRAYS

An array is a collection of variables. Some programming languages restrict arrays to a single variable type, while other programming languages place no such constraints. Arrays can be used to represent tree structures. For example, an array of integers can be constructed such that every array element represents a node. The array is constructed so that all but one of these elements dereference into the array index of their respective parents. To represent additional data, such as the branch length, multiple parallel arrays are used so that the element's index of any given node in the parent-offspring array also dereferences to that node's branch length in the branch length array. This approach can be extended to incorporate, for example, an array for the next sister of the element, or other one-to-one or many-to-one relationships. This representation of tree shape is implemented in the Java program Mesquite (Maddison and Maddison, 2001).

Alternatively, multidimensional arrays can be used where the first dimension represent a node, and the second dimension's element various node attributes. This is analogous to the comma-separated file format used in the Discrete (Pagel, 1994), Continuous (Pagel, 1997) and Multistate programs (Pagel, 1999; Pagel, 1999). The latter approach, however, may not be possible for strongly typed programming languages.

By using one-dimensional parent-offspring arrays, the internal representation of the tree becomes implicitly rooted (see Figure VI-2). Using a two-dimensional array,

unrooted trees can be represented. This follows the concepts outlined in *Inferring phylogenies* (Felsenstein, 2003), where multiple elements that form a ring comprising a single node are distinguished. Each of these elements refers to another element in the ring (the "next" element) and an element outside the node, on the opposite side of a branch (the "out" element). In a two-dimensional array, the first dimension can be used for the sub-node elements in the tree, with the second dimension containing a field used for the focal element's "next" neighbor, and a field for the focal element's "out" neighbor.

ASSOCIATIVE ARRAYS

Some programming languages implement a special kind of array, where the indices are not integers, but keys of some other form – usually strings. Arrays of this kind are known as associative arrays (or "dictionaries", "hashtables" or "hashes"). Associative arrays yield the functionality of key – value pairs. This functionality can be put to use to store multiple node properties into a single variable: a node is represented as an associative array with keys that dereference, for example, to its branch length, to a reference to its parent, siblings and children. By implementing references this way, the tree becomes implicitly rooted (see Figure VI-3). Associative arrays are used this way in the Python PIPRES libraries (Mark T. Holder, http://cvs.sdsc.edu/cgi-bin/cvsweb.cgi/phylo/framework/python/PIPRes/) and the Perl Bio::Phylo libraries (Rutger A. Vos, http://search.cpan.org/~rvosa/Bio-Phylo/). Associative arrays can also be used to append an arbitrary number of additional key/value pairs to nodes, as is implemented in the Java PAL libraries (Drummond and Strimmer, 2001).

The ring structure (Felsenstein, 2003) can also be implemented using associative arrays, where every element in a ring is an associative array with "out" and "next" keys

that dereference to other such associative arrays. Trees represented this way are implicitly unrooted (Figure VI-4).

POINTER STRUCTURES

Some programming languages implement a functionality whereby *the location of a variable* can be referenced. These references are known as "pointers" (strictly speaking, C and C++ have true pointers that point to the memory address of a variable, while other languages implement the same functionality – and sometimes refer to them as "pointers" – when usually they are, as in the case of Java, more properly called "references").

By constructing complex variables that contain pointers that point to other such variables, the functionality for representing tree shapes becomes available. Typically, this is done by creating node "structs" – complex variables that group other variables (integers, strings, real numbers, pointers). Some fields of the struct are used to store pointers to, for example, the parent, next sister and first daughter, while other fields are used to store additional data for the node such as its branch length. This is a functionally similar approach to that using associative arrays (and shown in Figure VI-3). MrBayes is an example of this approach. MrBayes is a program for the Bayesian estimation of phylogeny (Huelsenbeck and Ronquist, 2001; Ronquist and Huelsenbeck, 2003), written in C. Nodes and trees are implemented as structs. The **`TreeNode`** struct (see, for example, mb.h for MrBayes v3.1.1-p1, CVS revision 1.1) includes pointers to the "**`left`**" and "**`right`**" nodes, and the ancestor node. **`TreeNode`** structs are subsequently organized in pointer arrays in the **`Tree`** struct, where one node is explicitly defined as the root. The implication of the **`TreeNode`** approach is that trees are bifurcating. In order to facilitate unresolved trees a separate **`PolyNode`** struct is provided, which contains

pointers to the "**left**" node, the "**sib**" (next sister) node, and the ancestor. The **PolyNode**s are subsequently organized in the **PolyTree** struct, which again defines a node explicitly as root.

Unrooted trees can be implemented using pointer structures by implementing the struct's pointer fields to point to the "next" and "out" struct (Figure VI-4). The canonical example of this technique is Phylip, a collection of software packages written in C (Felsenstein, 1989). The node struct (defined in phylip.h, header version 3.6 from the 3.64 release) is essentially that of the ring structure where structs only point to the "**next**" and "**out**" struct, and a ring of multiple structs jointly form a node.

Internal representations of nodes using structs or associative arrays do not necessitate combining nodes into a container – the tree structure is an emergent property of the interrelationships of the nodes, so technically a tree struct only needs to contain a pointer to the root node in addition to tree metadata fields, but somehow trees are often implemented as arrays of node structs (sometimes a few additional, unconnected "scratch" nodes are added to the array, so that some memory has already been allocated before rearrangement or resolution procedures start).

Taking the described techniques into consideration, a distinction can be made between, on the one hand, approaches where nodes explicitly refer to one another, using pointers or references, and on the other hand approaches where the programmer establishes the convention (perhaps only with herself) that the integer held by an array element *means* the index of a related node. I will refer to the former approach as "true" recursive data structures, and the latter as "recursive-by-convention".

DATABASE DESIGNS

A special problem with regard to the internal representation of phylogenetic trees is posed by databases. The prevailing model in this field is the relational model, based on predicate logic and set theory (Codd, 1970). Under this model, information stored in a database must be "normalized" (i.e. made non-redundant). The construction of phylogenetic databases has been advocated for some time now (Sanderson et al., 1993), but the first version of TreeBASE (http://www.treebase.org/treebase/) – the database intended to aggregate the results of phylogenetic analyses – stored trees as strings. Recent work has gone into developing database designs that take a more "tree-aware" approach (Nakhleh et al., 2003), resulting in a design for the next generation of TreeBASE. Figure VI-5 shows a simplified version of this design. Depending on the database management system chosen, these data structures can be implemented as true recursive structures (perhaps using foreign key constraints and stored procedures) or recursive-by-convention.

Node and Tree Objects

One of the prevailing current paradigms in software engineering is that of "object-oriented" design. Simply put, the idea is that software should model the entities in the problem space it seeks to address as classes that are instantiated as "objects". The objects communicate with one another through their methods, in order to solve, jointly, the problem for which the software was written. Some programming languages were developed from the ground up with object-oriented design in mind, e.g. Java and Python; others – such as C++ and Perl – take, perhaps for historical reasons, an intermediate stance, while yet others have no explicit notion of objects – such as C.

Encapsulation

One of the reasons why object-oriented design is a suitable approach, especially for larger projects, is the possibility of "*encapsulation*", that is, an object can implement certain code behaviors, "*methods*", and hold certain variables, combined – encapsulated – into itself.

The variables the object holds (say, if the object models a node, its branch length) can be shielded from direct access or manipulation by other objects. Instead, the object can make access to these variables available through "*accessors*", methods that return the value of the variable. Likewise, methods might allow the manipulation of these variables, through "*mutators*". An advantage of this approach is that the object itself can decide, behind the scenes, whether or not the change that some other object wishes to make to its state is allowed. For example, a node may disallow negative values for its branch lengths. Hence, checks as to the validity of the proposed branch length are not necessary elsewhere, which simplifies the code. The underlying data structure becomes effectively

decoupled from the rest of the program, what is presented to the outside world is the interface, while what happens behind the interface can be changed without action-at-a-distance side-effects. The implications of the chosen underlying data structure used to represent nodes and trees become somewhat irrelevant: a data structure that implies rooting (e.g. a child → parent array), can be encapsulated such that it pretends to be unrooted – or vice versa. The integer arrays used by Mesquite (Maddison and Maddison, 2001) are encapsulated this way, providing for pretend-unrootedness.

Additional data – say, the tree's name, or score; or cross-references with related entities such as taxa or characters – that cannot conveniently be stored in a single complex data structure, for example because they describe many-to-many relationships, can still be tucked into the object. As well, the object can make available other methods such as implementations of tree traversal, and methods that return properties that turn out to be commonly required in larger analyses, such as the path length from the "*invocant*" (the instance to which the method call is addressed) to the root, the number of nodes from the invocant to the root, a list of all the invocant's ancestors, descendants, and so on.

The Phylogenetic Analysis Library, "PAL" (Drummond and Strimmer, 2001) consists of a Java library for molecular evolution and phylogenetics. It implements node objects (see, for example, class **`pal.tree.SimpleNode`** in pal/tree/SimpleNode.java, CVS revision 1.27). The **`SimpleNode`** object implements accessors and mutators for the parent node and the children. The **`SimpleTree`** object (see, for example, class **`pal.tree.SimpleTree`** in pal/tree/SimpleTree.java, CVS revision 1.23), has a less-rich API than Mesquite: no traversal methods are provided.

The design chosen for **SimpleNode**s and **SimpleTree**s in PAL implies that i) trees are rooted, there is no work-around to pretend unrooted-ness, *ii*) trees can have polytomies; iii) traversal methods for **SimpleNode**s and **SimpleTree**s have to be implemented separately.

PAL also implements nodes and branches specifically for combinatorial optimization: the implementations of **FreeNode** (**pal.treesearch.FreeInternalNode**, and **pal.treesearch.FreeLeafNode**) and the **FreeBranch** object provide a rich interface for traversal over "left" and "right" children – which implies binary rooted trees – branch swaps, and likelihood tests. In this implementation, the nodes are not explicitly aggregated into a tree object; the tree only "exists" as the result of the connections between the **FreeNode**s.

INHERITANCE

In the context of object-oriented design, "inheritance" is the ability to take an existing class and use it as a starting point for another class. This can be used to create special purpose child classes that inherit much of their functionality from a more generalized parent class without code duplication. For example, a special purpose "bayesian tree" class can inherit much of its accessor and mutator methods from a general purpose tree class, and implement additional methods particular to the Bayesian problem space (e.g. **tree.setPosterior**, **tree.getPosterior**).

POLYMORPHISM

Polymorphism is the notion that a parent class provides a "place holder" method that can be overridden by children that inherit from it. Different child objects can be

supplied as arguments to a method that has the parent class in its argument list. The children then sort out for themselves that actually their own implementation of the "place holder" is to be used. Although inheritance and polymorphism are considered essential features of object-oriented design, they are not central to internal representation of tree shape, where concern lies more with "has a" (parent) and "has many" (children) relationships than with "is a" relationships (i.e. the "object – relational model impedance mismatch").

TREE TRAVERSAL

Phylogenetic trees are represented internally in order to do something with them – a calculation, a manipulation of the tree's shape. This implies that the nodes in a tree must be visited, perhaps in various ways. Hence, internal representations of tree shapes must allow for – hopefully efficient – tree traversal algorithms.

Sometimes there is no constraint on the order in which nodes are visited. In that case, it may be most efficient to simply visit the nodes in the order in which they were inserted into an array containing them. However, in many instances nodes must be visited in a more meaningful way, by following some pattern of relatedness. In such cases recursion is used.

RECURSIVE TRAVERSAL

Recursive subroutines are subroutines that call themselves with changing arguments. This type of subroutine can be used to visit the nodes on a tree in various orders. Following are several types of traversals relevant for tree processing (see, for example: http://www.nist.gov/dads/).

Pre-order traversal — Using this type of traversal, all nodes in a tree are processed by processing the root and subsequently processing all subtrees. Example pseudocode:

```
preorder(node)
begin
      print node.name;
      if node.first_daughter is not null, then
            preorder(node.first_daughter);
      if node.last_daughter is not null, then
            preorder(node.last_daughter);
end
```

On the tree of Figure VI-3, this prints out "n3 n2 n1 A B C D".

In-order traversal — First processing the left subtree, then the root, and lastly the right subtree is known as in-order traversal. Example pseudocode:

```
inorder(node)
begin
      if node.first_daughter is not null, then
            inorder(node.first_daughter);
      print node.name;
      if node.last_daughter is not null, then
            inorder(node.last_daughter);
end
```

On the tree of Figure VI-3, this prints out "A n1 B n2 C n3 D".

Post-order traversal — Post-order traversal is a traversal whereby the nodes in a tree are visited recursively by first processing all subtrees, and then the root. Example pseudocode:

```
postorder(node)
begin
      if node.first_daughter is not null, then
            postorder(node.first_daughter);
      if node.last_daughter is not null, then
            postorder(node.last_daughter);
      print node.name;
end
```

On the tree of Figure VI-3, this prints out "A B n1 C n2 D n3".

Level-order traversals — The preceding traversal types all assume the tree is binary, so that all descendants of a node are visited by visiting its first daughter and last daughter. Level-order traversals, where nodes are visited by level, for example by first visiting the root, then its children, then theirs and so on, do not place this constraint. Two common traversals in phylogenetics are breadth-first:

```
breadthfirst(node)
begin
      print node.name;
      if node.next_sister is not null, then
            breadthfirst(node.next_sister);
      if node.first_daughter is not null, then
            breadthfirst(node.first_daughter);
end
```

On the tree of Figure VI-3, this prints out "n3 n2 D n1 C A B".

Depth-first:

```
depthfirst(node)
begin
      print node.name;
      if node.first_daughter is not null, then
            depthfirst(node.first_daughter);
      if node.next_sister is not null, then
            depthfirst(node.next_sister);
end
```

On the tree of Figure VI-3, this prints out "n3 n2 n1 A B C D".

EFFICIENCY

The efficiency of algorithms is often quantified in *O(N) notation* (Knuth, 1976), the worst-case performance change (*O* for "order of growth") as the size (*N*) of the data set that the algorithm operates on increases. Dereferencing an array element is *O(1)*, the no growth curve: apart from implementation details such as memory management, every lookup, irrespective of array size, takes the same amount of time. Scanning an array (for example to look for the highest value), is *O(N)*. Nested scans with inner loops are $O(N^2)$ – one outer loop and one inner loop – and beyond (i.e. $O(N^3)$ for three levels of nesting). In cases where data is ordered specifically for fast lookups, much greater efficiency can be achieved. For example, if data is iteratively partitioned, as in a balanced binary tree, probing the data takes *O(log(N))*. This is analogous to guessing a number below ten, with hints for higher or lower (which shouldn't take more than four guesses in the worst case).

As applied to phylogenetic tree structures, the richness of the structure influences the efficiency of traversal. If a tree is implemented as a parent->offspring array, finding the siblings of any given node is *O(N)*: the first step is to dereference the focal node's array element to obtain its parent's index (*O(1)*), the second step is a scan of the array (*O(N)*) to find other elements that dereference to the same index (and hence are siblings of the focal node). More efficient are implementations with a sibling array, in which case there is only a single lookup: *O(1)*. Especially as the amount of data (*N*) to operate on becomes large, algorithm efficiency differences become important, with some algorithms becoming prohibitively slow far sooner than others. On the other hand, rich tree structure implementations – which have the potential of providing for more efficient traversal – also have greater memory requirements, which also can become prohibitive as the amount of data increases. There is therefore a trade-off between the two, the optimum of which provides the most scalable solution.

DESIGN PATTERNS

In designing software, reusable techniques can be distinguished that are applicable across programming language boundaries, or even programming paradigms (e.g. object-oriented programming versus procedural programming). To the concepts outlined in the preceding sections, several design patterns (Gamma et al., 1995) are applicable.

NODES AS INSIDE-OUT OBJECTS

An approach to dealing with large numbers of fine-grained objects – such as the nodes on a large tree – has recently been introduced in the Perl community: *Inside-out* objects (Conway, 1999). Originally conceived as a way of enforcing encapsulation where this is not implemented by default in this language it is useful for the purpose of this paper as it is yields compact, pointer-free, memory efficient objects that can be easily serialized. The basic idea is that objects do not carry their own instance data around, rather, the class they belong to contains lists of instance data, and the objects only carry the "key" (in Perl the stringified memory address of a variable initialized in the constructor; since only one variable can start at any given memory address this ensures unique keys – however, any unique key will do) to dereference the values that apply to the instance. In the encapsulated child → parent arrays implemented in Mesquite the nodes (really just unique integers) can be viewed as inside-out objects.

The choice for the array approach in Mesquite was driven by the memory considerations: object representation of nodes turned out to be too costly. In addition, in the case of virtual machines using reference counting it simplifies garbage collection as

the implementation is recursive-by-convention, and so circular references that are ignored by the garbage collector cannot exist.

(Although written in C, so presumably with less memory overhead, nodes in PAML (Yang, 1997) also refer to their parents and children with array indices, but these indices are stored in fields of node structs; so, although recursive-by-convention, this does not follow the inside-out pattern.)

Trees as Strongly Typed Collections

If nodes are implemented as objects, there is often a need for a collection object – the actual tree – to manage, transform and query the whole collection of nodes at once. Considering that the tree shape is the emergent property of the connections between nodes, a tree can be implemented as a structure that only holds a reference to one node (probably the root) from where all others can be reached by traversal, in addition to tree metadata such as the tree's name.

Often, however, there is scope for a tree object that contains all nodes. It is naively appealing to implement this in the form of whatever implementation the language permits for a collection of generic objects. This leads to several problems. Objects of the wrong type might be inserted in the collection, breaking down the assumption that the contained objects implement a common interface that can be called while iterating over them; also the contained objects may have to be coerced to the necessary type to be inserted in the collection (and possibly again when accessed), which leads to noise in the code.

The solution should then be to implement a strongly typed collection – at least in the colloquial sense of being picky about types, either by checking the inheritance tree of

the object to be inserted or, more flexible, by "duck-typing" (i.e. if an object walks like a duck and quacks like a duck, it must be a duck).

Tree Traversal using an Iterator Interface and Visitor Object

As trees are from some perspective collections – containers – of similar objects over which different traversals are performed, an obvious pattern to recognize is that of the *Iterator* object, that is, an object providing an interface to traverse aggregated data, abstracting away the implementation details of the underlying data structure. The iterator provides a number of methods (possibly combined): i) checking whether an element is available; ii) returning the value at the current position; iii) moving to the next element. In simple implementations step i) and ii) are combined, such that whatever value for "undefined" is used in the language is returned to indicate that the requested element (and thus its value) is unavailable. The third step is a method call like `Iterator.next()` that perhaps just implements an index increment on the underlying array, or a complex traversal of some form.

As the iterator is traversing the underlying data, operations or calculations of some form are performed. The operation can be made reusable by implementing it as a method in the class the objects that the operation is applied on belong to. This, however, has two drawbacks. Firstly, this still restricts reuse to that class (and classes that inherit from it); secondly, this leads in the long run to feature-creep and bloated classes. An alternative is the *Visitor* pattern: the operation to be performed on a collection of objects during a traversal is abstracted into a visitor that is invoked during a traversal. The implementation of the visitor can be in the form of a visitor object, a reference to a code block, or a closure.

The PIPRES libraries (Mark T. Holder, http://cvs.sdsc.edu/cgi-bin/cvsweb.cgi/phylo/framework/python/PIPRes/) implement some of this functionality in Python, by providing several iterators (**`iterChildren`**, **`iterEdges`**, **`iterInternals`**, **`iterPostorder`**, **`iterPreorder`**, **`iterTips`**) that take a predicate for node-filtering as an argument. An implementation that follows the notion of a visitor object more closely is that of **`Tree::Simple::Visitor`** (Stevan Little, http://search.cpan.org/~stevan/Tree-Simple-1.15/), where the visitor object has accessors and mutators for node filters, a **`visit`** method to traverse the tree and a getter to retrieve the results (the traversal type is specified in the constructor).

FLYWEIGHT OBJECTS, COMPOSITE PATTERNS AND IMMUTABLE OBJECTS

Naïve object-oriented design is at odds with efficient memory usage. Instantiating rich node and tree objects that hold a lot of data quickly becomes prohibitive, if the goal is to process large sets of big trees. The solution lies in sharing of data between objects, and one of the patterns to implement this is the *Flyweight* object (Gamma et al., 1995).

The canonical example (Calder and Linton, 1990; Calder and Linton, 1992) to illustrate this approach is that of a word processor that needs to keep track of many character objects (i.e. glyph representations, possibly with class data and instance data: font face, weight, point size – in any case something richer than a single ASCII character). Obviously, instantiating a new object for every typed character quickly fills up memory. Instead, a *FlyweightFactory* object instantiates a separate object for every category of characters – say, all bold-face twelve point Times New Roman characters. Then, the individual flyweight objects (the characters on the screen) can be considered as nothing more than references to the single object for the whole category of characters,

each holding virtually no extrinsic – context-dependent – data apart from position, sharing most of their state with the other flyweight objects of the same category.

The concept of sharing state can equally be applied to higher level categories of objects (for example a category object for all for all Times New Roman characters), which can be implemented as higher level flyweight objects. Organizing objects in hierarchies of increasing generality is recognized as the *Composite* pattern – and so flyweight objects and composite patterns are often observed in the wild working in concert.

What happens if an intrinsic state variable of a flyweight instance is modified? Logic dictates that the category object the flyweight object references changes state, and the change cascades down the inheritance tree to the other flyweight objects in the same category, which is perhaps not what is supposed to happen. The solution is to implement *Immutable* objects, which, in the terminology of design patterns, are objects that do not change state, but rather, on attempts to assign new values to instance variables (e.g. changing the font style from bold to italic) the immutable object returns an instance of a flyweight object belonging to the desired category – in other words, the flyweight object is "moved" to some place else in the composite tree, some place where the flyweights holding the desired intrinsic state live.

How is this applicable to phylogenetics? The key lies in separating the extrinsic and intrinsic state of the object. Nodes in a tree almost completely consist of extrinsic state: they exist in relation to other nodes, and may have real number values associated with them. Hence, the flyweight pattern is in most cases not applicable to node objects. The same is not true for (sub)tree objects in a set of trees. A subtree factory object that

maintains a pool of immutable subtree objects can be used to reduce the memory requirements of large sets of similar trees, effectively caching and reusing topologies. Likewise, partial results of computations – for example likelihood computations – can be cached with the subtrees and site patterns the result applies to. This pattern is implemented to some extent in PAL, (Drummond and Strimmer, 2001), in **`pal.eval.FastLikelihoodCalculator`**, which caches partial likelihoods of invariant subtrees. (The flyweight pattern seems even more obviously applicable to handling annotated nucleotide sequences, where each individual site is almost as similar or dissimilar to all others as the character objects from the word processor example.)

SERIALIZATION

Serialization is the process of storing an object as a series of bytes, or in a human-readable (marked-up text) format. These storable representations can subsequently be "de-serialized", in order to retrieve an identical clone of the object – but possibly on a different computer, or represented in a different programming language, or both. In practice, serialization often is a two-step process: in the first step, the data structure to serialize is changed to a more manageable format, which then in the second step is serialized, often by an abstraction layer that does so transparently.

In the present context, I will consider both the serialization of objects and of raw data structures. A number of serialization standards has emerged in recent years, which I will discuss here.

TEXT FORMATS SPECIFIC TO TREE STRUCTURES

Although technically not serialization, the description of tree shapes in a parenthetical statement (known as the "Newick" format), and the introduction of the Nexus file format that builds on this syntax (Maddison et al., 1997) has been instrumental in allowing for the interoperability of software for phylogenetics. The Nexus standard therefore warrants mention; however, it has also forced authors of phylogenetics software to develop their own parsers to deal with a standard that has been extended in various directions – yielding a number of slightly different implementations. It would, to this author's opinion, therefore be of great use for the phylogenetics community that a new, unambiguous standard emerges for which off-the-shelf serializers and de-serializers are available.

XML AND SOAP

Extensible Markup Language is a human-readable mark up format based on recommendations of the W3C (Bray et al., 2004). In fields related to phylogenetics, XML applications have emerged and are in use today. For example, for taxonomy there is SDD (Thiele, 2003), for molecular biology there is a variety of formats defined by the National Center for Biotechnology Information (http://www.ncbi.nih.gov/dtd/, accessed August 9, 2005), and for graph theorists there is GraphML (Brandes et al., 2001) and XGMML (Punin and Krishnamoorthy, 2001).

For phylogenetics, several applications have been proposed (Gilmour, 2000), and discussion has been ongoing for a number of years now, for example in the "phyloxml" project (Cannon and Zmasek, 2005) and in a project that has yet to decide on the name for the root level element (http://evolve.zoo.ox.ac.uk/phyloxml/, accessed August 10, 2005). However, the modular nature of XML could also be put to use by combining for instance SDD, GraphML and TinySeq into a composite standard that mimics the functionality of Nexus files. This reuse could yield, without much effort from the phylogenetics community itself, a useable standard for which parsers and well-defined standards are already available. The advantage is that this would yield a human-readable, text-based standard for data storage. The disadvantage is that this is, strictly speaking, not serialization. True serialization in XML can be achieved, however, using the Simple Object Access Protocol.

SOAP is an XML application based on recommendations of the W3C (Mitra, 2003). It is principally used for web services over HTTP. The SOAP standard defines syntax for the serialization of primitives and certain complex variables. For example,

integer arrays can be SOAP encoded, allowing for the serialization of the array data structure for trees. Serialization of structured data with references can be achieved using **`id`** and **`idref`** attributes. Code generation tools (such as castor) are available to create stubs for serialization and de-serialization following this standard. However, this yields text that is probably too verbose for high-speed computation, especially for large data sets. This is a problem with any XML-based serialization scheme where the data structure is transferred between processes rather than as a storage and query format. For example, the Tree of Life web project (Maddison and Schulz, 2004) as exported to XML has a file size of about 30 Megabytes, which is rather impractical for rapid transmission and processing.

RDBMS

Serializing objects to relational databases while respecting both the object-oriented paradigm and the relational data paradigm is more problematic than it seems. Database tables are not classes, and columns are not attributes, even though data models and object models might look alike in diagrams. Objects are related, one to another, by "is a" relationships, while database records are related via "has a" relationships. Another, perhaps more practical problem is that the records in a database most conform to a table "shape", while objects place no such constraints. Or, in other words, tables can only be a special case of objects. How to reconcile the two?

The database can be made to conform to object-oriented principles, by using an RDBMS that can store objects directly as records of type "object" (e.g. as **`java.sql.Types.JAVA_OBJECT`**), or by using an abstraction layer that provides an object-oriented interface to the entities in the database (which, to provide this

functionality, is restricted in its schema, e.g. see **`Class::DBI`**, http://search.cpan.org/~tmtm/Class-DBI). The former approach limits access to the database to those programming languages that can de-serialize the object type, while the latter approach restricts objects and tables to a one-to-one mapping.

Alternatively, objects can be made to adapt to the relational data model, which in turn has implications for the underlying data structures the object encapsulates. A data structure such as in Mesquite (Maddison and Maddison, 2001), where nodes have unique identifiers, adapts easily to a relational data model. Conversely, true recursive data structures need to be converted to a more record-oriented structure, either by the application that hands the tree over to be inserted in the database, or by an abstraction layer around the database.

CORBA

Common Object Request Broker Architecture is a specification defined by the Object Management Group (http://www.omg.org/). Clients can place method calls on objects implemented by a server via object request brokers (ORBs). The interface that the client and server agree on is specified in interface definition language (IDL), which syntactically is similar to C and C++, though pointer-free. In fields related to phylogenetics (in particular molecular biology), prototype implementations have been deployed in the BioCorba stack (http://cvs.bioperl.org/cgi-bin/viewcvs/viewcvs.cgi/?cvsroot=bioperl). The CIPRES project (http://www.phylo.org) is currently developing a software stack in Java, C++, Python and Perl that deploys CORBA. All pointer-free, recursive-by-convention structures discussed in the previous sections can be implemented in IDL, and hence be used on a CORBA platform.

CONCLUSION

Is there a single ideal data structure for phylogenetics? Obviously not: different contexts have different optima. Whether a program-to-be deals with a lot of data, or little; whether the goal is to do many computationally intensive tasks, or mostly bookkeeping; what programming language and programming paradigm are chosen; all influence the choice of data structure.

The scope of this paper is to explore solutions that are suitable in the context of interoperability between different programming languages and paradigms, with the goal of analyzing large trees. In this context a suitable approach to the internal representation of tree shape – at least during the execution stages surrounding serialization – is one that, to some extent, caters to the lowest common denominator. Choosing an implementation that is idiomatic and non-portable (such as Java objects or hash tables) would be at odds with the requirement of interoperability. Hence, pointer-free and reference-free implementations such as the array based approach are preferable.

To serialize large trees, the implementation data structure has to be light-weight: serialization should not – on top of a processing bottleneck – become a bandwidth bottleneck. Hence, strategies for reuse of components must be employed. For example, when a set of trees is serialized, taxon names should be reused, along the lines of the translation table in Nexus files. Likewise, flyweight-like patterns might be deployed to reuse identical trees and subtrees.

A related issue is that of the memory requirements of the de-serialization data structure. Ideally, data is structured in such a way that processing can commence before all data has been received or parsed. This consideration is not applicable to CORBA, but

it is an important consideration when parsing XML, and when interacting with databases. XML parsers come in two flavors: ones that need to store the entire document object model (DOM) tree in memory, and ones that can process the marked up data as the parser traverses the DOM tree ("stream-based parsers"). Seeing that XML files can quickly grow very large, it would be very much more efficient if internodal relationships can be reconstructed while the XML is being parsed. Stream-based parsers can pass on the chunks of data they have parsed only once the closing tag has been reached. Hence, suggestions for XML applications for phylogenetics that represent a phylogenetic tree as a DOM tree of nested elements (e.g. see: Felsenstein, 2003; Gilmour, 2000) – although intuitively attractive – are problematic, because the parent node of any clade can only ever be processed until the entire DOM subtree it subtends has been processed. One part of the solution to this that plays to the strengths of stream-based parsers is to implement a tree structure as a set of `child -> parent` elements at the same level in the DOM tree. The other part of the solution is the minimization, or elimination, of forward references in the structure.

Eliminating forward references in ancestor functions is easy: as long as the parent node always precedes its child nodes, either in an integer array, in an XML file, or in a database insertion transaction, all is well. A depth-first, breadth-first or pre-order traversal would be enough to generate a structure that entirely eliminates forward references. (Apart from XML parsing, eliminating forward references and presenting the data linearly is also advantageous in the context of database interactions because it allows insertion of all nodes using a prepared statement with SQL placeholders. This greatly

improves performance, and insertions can be done in a single transaction, which goes a long way to preserving referential integrity.)

Serialization to persistent storage (such as files or databases) differs from serialization between processes using CORBA in that, in the latter case, the notion of a "stream" of data is abstracted away. While parsing a file, or reading from a database handle, data is processed as a linear stream of tokens. Although communication between ORBs is also in the form of a stream of data, this process is (perhaps intentionally) abstracted away, so that whatever data structure is transmitted always becomes available to the recipient as the complete structure. Hence, forward references, and their elimination, are not a concern: an ancestor function implemented as an array can be read in whatever direction is most efficient. In this context, other considerations come into play, perhaps most notably – especially in comparison with file parsing – a greater sense of urgency: an architecture that implements real-time communication between processes is not of much use if the real-time communication, and the subsequent unpacking of the received message, takes a long time. Implementations of CORBA where Newick tree descriptions are transmitted are therefore highly inefficient: parsing a string of balanced, nested, parenthetical constructs is a fairly complicated and time consuming procedure compared to iterating over an array or a set of arrays. It is also wasteful to throw away information held by the sender. Much more efficient would be to serialize a tree data structure as a parallel series of integer arrays, for example a parent → offspring array, a first daughter array and a next sister array, where the same index in the three arrays corresponds to the same focal node. With the internodal relationships defined at three levels (parents, siblings and children), more densely connected relationships – at least to

the extent discussed in this review – can be established in a single iteration over the parallel arrays, i.e. in *O(N)*.

The last consideration in choosing a suitable data structure discussed here is also the most important one: is the data structure suitable for representing biological observations and inferences? Whether multiple, contemporaneous speciation events can occur is perhaps a matter of perspective, but invoking polytomies is certainly necessary in some cases, if only to indicate ignorance. Hence, strictly dichotomous data structures do not meet the needs of phylogeneticists. Evolutionary processes take place over (sometimes long) stretches of time. The true Tree of Life has a root and a temporal axis. However, it is sometimes useful to represent a tree structure as unrooted. Does this necessitate an unrooted data structure for serialization? Maybe not: the root in a tree obviously identifies itself (it has no parent), so flagging a rooted tree as *actually* unrooted is probably enough – the data structure can then be converted to a ring structure, if necessary, or methods that pretend the tree is unrooted can be implemented.

Can the diversity of life be represented as an acyclic graph anyway? Hybridization and lateral gene transfer, for example, certainly exist; in some cases a network representation is therefore more suitable. However, this requires approaches that fall outside of the scope of the present review. In all cases discussed here, the problem could be phrased in terms of the representation of one-to-many relationships (one parent, many children), but in networks the relationships are many-to-many: a different problem space that no doubt will become increasingly important in phylogenetics.

REFERENCES

Bayer, R. Year. Binary B-Trees for Virtual Memory in ACM-SIGFIDET Workshop. San Diego, California:219-235.

Bayer, R., and E. M. McCreight. 1972. Organization and Maintenance of Large Ordered Indexes. Acta Informatica 1:173-189.

Brandes, U., M. Eiglsperger, I. Herman, M. Himsolt, and M. S. Marshall. Year. GraphML Progress Report: Structural Layer Proposal in Proc. 9th Intl. Symp. Graph Drawing (GD '01):501-512.

Bray, T., J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. 2004. Extensible Markup Language (XML) 1.0 (Third Edition).

Calder, P. R., and M. A. Linton. Year. Glyphs: Flyweight objects for user interfaces in ACM User Interface Software Technologies Conference, Snowbird, UT:92-101.

Calder, P. R., and M. A. Linton. Year. The object-oriented implementation of a document editor in Object-Oriented Programming Systems, Languages and Applications Conference Proceedings. ACM Press, Vancouver, British Columbia, Canada:154-165.

Cannon, E. K. S., and C. M. Zmasek. 2005. Proposal for pyloXML level 1, version 0.2.

Codd, E. F. 1970. A relational model of data for large shared data banks. Communications of the ACM 13:377-387.

Conway, D. 1999. Object Oriented Perl. Manning Publications, Greenwich, CT.

Diestel, R. 2005. Graph Theory, 3rd edition. Springer, Berlin.

Doolittle, W. F. 1999. Phylogenetic classification and the universal tree. Science 284:2124-2129.

Drummond, A., and K. Strimmer. 2001. PAL: An object-oriented programming library for molecular evolution and phylogenetics. Bioinformatics 17:662-663.

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

Felsenstein, J. 1985. Phylogenies and the Comparative Method. American Naturalist 125:1-15.

Felsenstein, J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics 5:164-166.

Felsenstein, J. 2003. Inferring phylogenies. Sinauer Associates, Inc., Sunderland, Massachussets.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. Design Patterns: Elements of Reusable Software Components. Addison-Wesley Professional, Boston, Massachussets.

Gilmour, R. 2000. Taxonomic markup language: applying XML to systematic data. Bioinformatics 16:406-407.

Guyer, C., and J. B. Slowinski. 1991. Comparisons of observed phylogenetic topologies with null expectations among three monophyletic lineages. Evolution 45:340-350.

Guyer, C., and J. B. Slowinski. 1993. Adaptive radiation and the topology of large phylogenies. Evolution 47:253-263.

Harvey, P. H., and M. D. Pagel. 1991. The comparative method in evolutionary biology. Oxford University Press, Oxford.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97-109.

Huelsenbeck, J. P., and F. Ronquist. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics 17:754-755.

Knuth, D. E. 1976. Big Omicron and big Omega and big Theta. ACM SIGACT News 8:18-24.

Maddison, D. R., and K.-S. Schulz. 2004. The Tree of Life Web Project.

Maddison, D. R., D. L. Swofford, and W. P. Maddison. 1997. NEXUS: An extensible file format for systematic information. Systematic Biology 46:590-621.

Maddison, W. P., and D. R. Maddison. 2001. Mesquite: a modular system for evolutionary analysis.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equations of state calculations by fast computing machines. J. Chem. Phys. 21:1087-1091.

Mitra, N. 2003. SOAP Version 1.2, W3C Recommendation, 24 June 2003.

Nakhleh, L., D. Miranker, F. Barbancon, and W. H. Piel. Year. Requirements of phylogenetic databases in Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03):141-148.

Pagel, M. 1994. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proceedings of the Royal Society London Series B 255:37-45.

Pagel, M. 1997. Inferring evolutionary processes from phylogenies. Zoologica Scripta 26:331-348.

Pagel, M. 1999. Inferring the historical patterns of biological evolution. Nature 401:877-884.

Pagel, M. 1999. The maximum likelihood approach to reconstructing ancestral character states of discrete characters in phylogenies. Systematic Biology 48:612-622.

Punin, J., and M. Krishnamoorthy. 2001. XGMML (eXtensible Graph Markup and Modeling Language) 1.0 Draft Specification.

Ronquist, F., and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572-1574.

Sanderson, M. J., B. Baldwin, G. Bharathan, C. Campbell, D. Ferguson, J. M. Porter, C. V. Dohlen, M. F. Wojciechowski, and M. J. Donoghue. 1993. The growth of phylogenetic information and the need for a phylogenetic database. Systematic Biology 42:562-568.

Slowinski, J. B., and C. Guyer. 1989. Testing the stochasticity of patterns of organismal diversity - An improved null model. American Naturalist 134:907-921.

Thiele, K. 2003. SDD Part 0: Introduction and Primer to the SDD Standard.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. CABIOS 13:555-556.

FIGURE VI-1 A ROOTED PHYLOGENETIC TREE

The child → parent relationship between terminal taxon *C* and internal node *n2* is indicated in bold as a reference for subsequent figures, where the same relationship is shown also, using their respective implementations. See text for details.

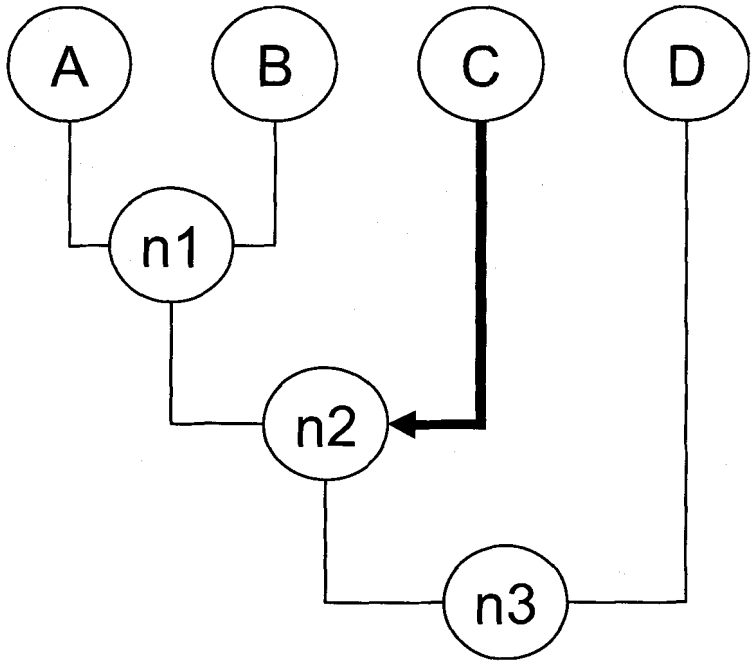


FIGURE VI-2 ANCESTOR FUNCTION

A representation of a rooted tree using an ancestor function, which can be implemented using, e.g., parallel integer arrays. See text for details.

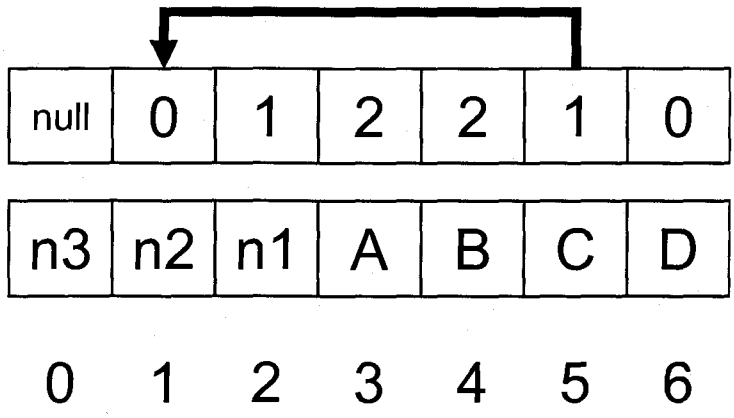


FIGURE VI-3 FIRST DAUGHTER, NEXT SISTER STRUCTURE

A dense representation of a rooted tree with explicit sibling relationships, which can be implemented in a number of ways. See text for details.

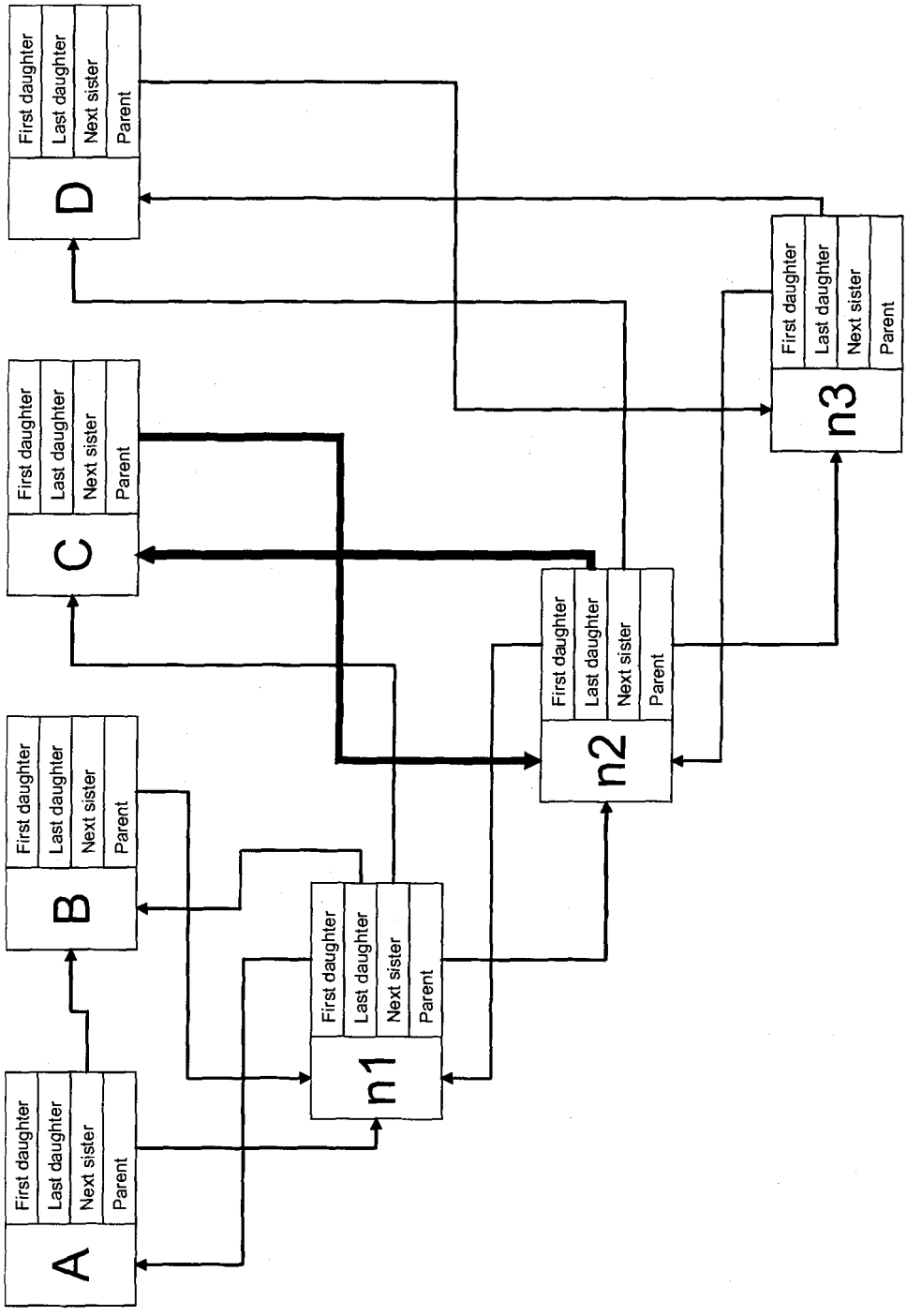


FIGURE VI-4 THE RING STRUCTURE

A tree representation using the "ring" approach, which is implicitly unrooted. The root must therefore be explicitly marked. See text for details.

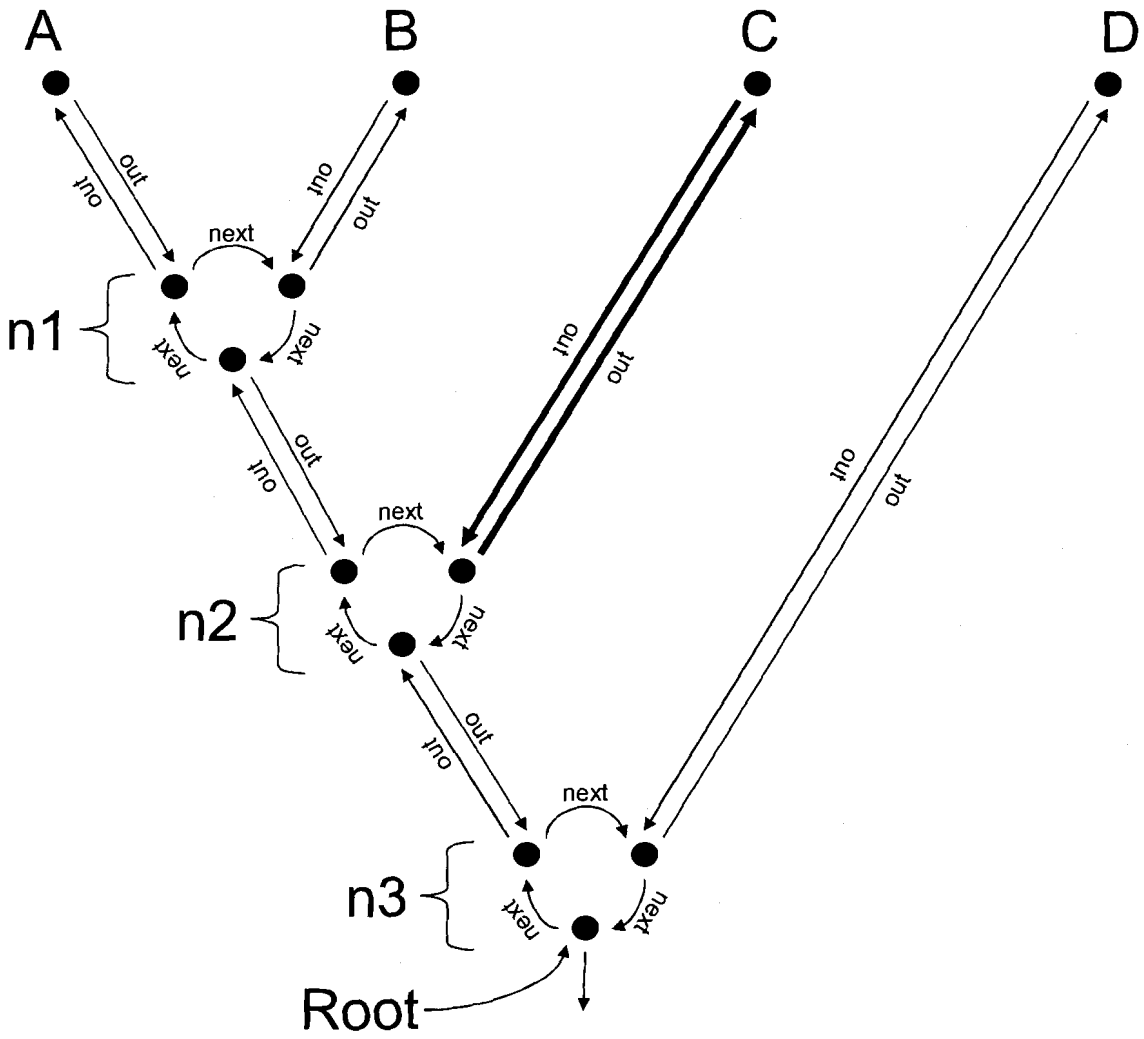


FIGURE VI-5 DATABASE SCHEMA

Representing a tree shape in a relational model, such as an RDBMS. See text for details.

Nodes

Name	**NodeID**
A	1
B	2
C	3
D	4
n1	5
n2	6
n3	7

Edges

ParentID	**ChildID**	**EdgeID**
5	1	1
5	2	2
6	3	3
6	5	4
7	4	5
7	6	6

CHAPTER VII - GENERAL DISCUSSION

Rutger A. Vos

INTRODUCTION

The unifying theme of this dissertation is that of the "Big Tree Problem", that is, the methodological difficulty that surrounds the inference and analysis of large phylogenies. Different aspects of the problem are addressed: i) the combinatorics problem that researchers face during the inference of large phylogenies in the likelihood and Bayesian frameworks; ii) the estimation of molecular divergence dates on supertrees (usually being the largest phylogenies in the literature, and often lacking branch lengths); iii) the modeling of expected divergence dates on supertrees; iv) the problem of the increase in required data to resolve large phylogenies, hence for supertrees the process of tree selection; v) the internal representation of phylogenetic trees in computer programming. The following section summarizes the findings in each of these areas.

SUMMARY

Chapter II discusses an algorithm dubbed the "Likelihood Ratchet". The algorithm is used to explore likelihood landscapes expected to be multimodal, i.e. containing multiple optima. Local optima pose problems for hill-climbing techniques as searches can get stuck on them. The Likelihood Ratchet escapes from these optima by iteratively perturbing the landscape through resampling of characters. The algorithm outperforms standard hill-climbing in tests on simulated and real data, including a data set whose multi-modality had been demonstrated previously (Salter, 2001). This chapter has been published in the journal *Systematic Biology* (Vos, 2003).

The following chapter extends some of the concepts of the Likelihood Ratchet to the Bayesian framework. In Bayesian analyses, a random walk is constructed that should, after a certain number of discarded steps, approach a stationary situation where solutions are visited in proportion to the target distribution of posterior probabilities (Larget and Simon, 1999; Mau et al., 1999; Yang and Rannala, 1997). The initial stage of these analyses is termed "burn-in", and it is this stage that poses problems as it is sometimes unclear when the stationary phase has been reached, and burn-in sometimes takes long (especially when analyzing large data sets). Small improvements sometimes take place long after the Markov chain seems to be stationary perhaps due to conflicting phylogenetic signals or multiple optima in the tree landscape (Huelsenbeck et al., 2002). Chapter III addresses part of this problem by introducing an iterative algorithm that perturbs the optimality landscape such that the burn-in Markov chain can move away from local optima more easily. Tests-of-concept on simulated and real data demonstrate

that this approach performs well: burn-in can be sped up using iterative jackknifing. This chapter is currently 'accepted with minor revisions' by *Systematic Biology*.

Chapter IV introduces a new approach to estimating divergence dates on MRP (Baum, 1992; Ragan, 1992) supertrees. The divergence date estimates from multiple, clock-like (Zuckercandl and Pauling, 1965) DNA sequence alignments whose taxon sets partially cover the supertree's taxon set can be combined as long as the topologies on which the divergence dates are estimated are compatible. To do so, the trees on which the divergence dates are estimated must be calibrated on common nodes. This yields sets of estimates for each node in the supertree to which multiple alignments speak. From these sets of estimates the composite estimate is calculated by taking, for example (to minimize the effect of outliers), the median of the set. The paper (published as Vos and Mooers, 2004) discusses this method as applied to the supertree of the Primates.

The subsequent chapter (Chapter V) presents the topology of the Primate supertree. This new phylogeny of the Primates forms the largest estimate of phylogeny for this order to date. It agrees broadly with earlier work done on this clade (Purvis, 1995), but it is better resolved and includes more species. In addition, this paper introduces a method to obtain expected divergence dates under a model of constant clade growth (Yule, 1925) for nodes for which no estimates are available, and it presents additional support values for clades in the supertree using Bremer (Bremer, 1994) and rQS (Bininda-Emonds et al., 2005) values. The chapter closes with a discussion on macro-evolutionary trends in the history of the Order Primates. This paper is currently in second review with the journal *Systematic Biology*.

Chapter VI is a review of the various ways in which tree shapes can be represented internally in forms useful for computer analysis. The approaches taken in various open source software packages for phylogenetic analysis (Drummond and Strimmer, 2001; Felsenstein, 1989; Huelsenbeck and Ronquist, 2001; Maddison and Maddison, 2001; Ronquist and Huelsenbeck, 2003; Yang, 1997) are discussed, with particular reference to the concept of "design patterns", i.e. reusable solutions for programming problems (Gamma et al., 1995). This paper also draws attention to the issue of "serialization", that is, transferring data structures and objects between programming languages and computers. This issue is of particular importance for large-scale projects seeking to integrate phylogenetic software written in multiple programming languages (e.g. see http://www.phylo.org).

IMPLICATIONS OF FINDINGS, AND FUTURE DIRECTIONS

SEARCH ALGORITHMS

The Likelihood Ratchet [Chapter II, (Vos, 2003)] has been discussed (Bruns and Shefferson, 2004; Mar et al., 2005; Salamin et al., 2005; Vinh and Haeseler, 2004) and applied (Alexander and Breden, 2004; Andersen and Ekman, 2005; Bedoya et al., 2005; Lewis et al., 2004; Tombes et al., 2003) in various articles. As there is great interest in phylogenetic inference in the Bayesian framework (Larget and Simon, 1999; Mau et al., 1999; Yang and Rannala, 1997) and the methodological problems that surround this approach [such as slow burn-in (Huelsenbeck et al., 2002)], it is possible that the iterative jackknifing algorithm (Chapter III, accepted with minor revisions by *Systematic Biology*) will spur similar interest. In any case, ratchet-*like* techniques early on in Bayesian analyses are sure to be a fruitful area of further research [some reweighting functionality in MrBayes (Huelsenbeck and Ronquist, 2001; Huelsenbeck and Ronquist, 2003) was implemented with this in mind (John Huelsenbeck, pers. comm.)].

SUPERTREE CONSTRUCTION

Although criticized by some (Gatesy et al., 2004; Gatesy et al., 2002) [but see (Bininda-Emonds et al., 2003)], supertrees are likely to be the most complete phylogenies for some time to come (at least for animals, given the lack of concerted sequencing efforts similar to the *rbcL* (Chase et al., 1993) effort among botanists). A new, larger and better-resolved supertree of the Primates (as presented in Chapter V, which is in review with *Systematic Biology*) will attract similar attention as the previous composite estimate for this Order (Purvis, 1995) which has been used in comparative (Abbott et al., 2003; Altizer et al., 2003; Alvarez, 2000; Anderson et al., 2004; Anderson et al., 2004;

Anderson et al., 2005; Barton, 1996; Barton, 1997; Barton, 1998; Barton, 2004; Barton and Harvey, 2000; Birdsey et al., 2005; Blomberg et al., 2003; Bohm and Mayhew, 2005; Bonine et al., 2005; Carbone et al., 2005; Carter, 2001; Carter and Mendis, 2002; Conroy, 2003; De Ruiter, 2004; Deaner and Nunn, 1999; Deaner et al., 2000; Diazuriarte and Garland, 1996; Eeley and Foley, 1999; Fa and Purvis, 1997; Fernandez and Vrba, 2005; Fish and Lockwood, 2003; Gardezi and Da Silva, 1999; Gatesy et al., 2004; Geissmann, 2002; Geissmann, 2002; Gittleman and Purvis, 1998; Harcourt, 2000; Heesy, 2004; Heesy and Ross, 2001; Hewitt et al., 2002), macro-evolutionary (Barraclough et al., 1998; Bokma, 2002; Bokma, 2003; Chan and Moore, 2002; Creevey et al., 2004; Diniz et al., 1998; Isaac and Cowlishaw, 2004; Isaac and Purvis, 2004; Jernvall and Wright, 1998; Martin, 2000; Mooers and Heard, 1997) and methodological (Bininda-Emonds, 2004; Bininda-Emonds, 2004; Bininda-Emonds, 2005; Bininda-Emonds and Bryant, 1998; Bininda-Emonds et al., 2002; Bininda-Emonds and Sanderson, 2001; Eulenstein et al., 2004) studies.

Lastly, the method to generate expected divergence dates on supertrees (or any other topology lacking branch lengths, discussed in Chapter IV and V) can be extended to generate the expectations under more complex models of clade growth, such as those incorporating extinction. At present, the technique relies on averaging over large sets of trees with randomly drawn labeled histories. However, an analytical solution is possible (Mike Steel, pers. comm.), which would greatly speed up the calculation of expected ages of splits. A future direction for the Bio::Phylo software package (discussed in the Appendix) is the inclusion of this analytical solution. In addition, a manuscript is in

preparation which will apply this analytical algorithm ('RankProb') to parts of the primate supertree, to test macro-evolutionary hypotheses of clade growth.

PHYLOGENETIC SOFTWARE DEVELOPMENT

The CIPRES project (http://www.phylo.org) is at the time of this writing in the process of expanding its CORBA interfaces to include abstract definitions of tree structures to be passed between 'services' (such as tree inference, tree shape manipulation). Chapter VI was written with this transition in mind. The manuscript may play a role in the discussions surrounding the choice between different tree and node structures as defined in the Interface Definition Language classes for the project architecture. Similarly, the Perl libraries documented in the Appendix play a role in the expansion of the BioPerl project (http://www.bioperl.org) to become more phylogenetics-oriented, including the definition of a CDAT ('Character Data And Tree') object, that is, an intersection object that links phylogenetic trees to (molecular, continuous, or categorical) character sequences (Stoltzfus et al., 2006). In all likelihood, the Bio::Phylo library code will be merged with the BioPerl code base (as well as being part of CIPRES) to implement this functionality (using the 'Bio::Phylo::Taxa::Taxon' object, see the Appendix).

NEW APPROACHES TO ADDRESS THE BIG TREE PROBLEM

These are exciting times for biologists: an understanding of the mechanics of molecular evolution, the ability to collect large amounts of molecular data relatively easily, and the computational approaches to analyze that data allow us to look millions of years into the past, or study to epidemic behaviour of viruses (or anything in between), and give us a better and better understanding of evolution and speciation. But, however fast computer power is growing, very large phylogenies will not be inferred by naïve brute force: the growth of the Big Tree Problem as larger and larger numbers of taxa are analyzed will likely continue to outpace Moore's law (i.e. the observation that the complexity of computer chips grows exponentially, doubling roughly every 24 months, whereas the number of distinct tree shapes grows hyperexponentially with the number of taxa). At present, solutions are sought in several directions.

The approaches discussed in this thesis, the likelihood ratchet and iterative jackknifing, seek to mitigate against pathologies in at present widely used methods of phylogenetic inference: heuristic searching under maximum likelihood and random walks under the Bayesian framework, respectively. The pathology of multiple optima in tree space, the notion of 'Tree Islands' (Maddison, 1991), is enhanced by the Big Tree Problem. Hence, using these techniques to infer *large* trees plays to their strength. However, these techniques by themselves presuppose that a phylogenetic data set can be analyzed on a single computer in reasonable time. This is not always the case, and novel developments that organize the problem in a way that multiple computers can be brought to bear have recently been introduced, in particular, parallelization and divide-and-conquer methods.

Recent versions of the MrBayes program for phylogenetic inference in the Bayesian framework (Larget and Simon, 1999; Mau *et al.*, 1999) implement parallelization by running heated Markov chains on separate CPUs connected by the "Message Passing Interface" (Ronquist and Huelsenbeck, 2003). Under ideal conditions (few swaps between chains), the computational intensity of adding chains scales roughly linearly (Altekar *et al.*, 2004) such that they can be distributed over a linearly growing number of processors. However, this approach can only be applied to inferential techniques that lend themselves to parallelization. This is the case for Metropolis-Hastings coupled Markov chain Monte Carlo, but not so for many other techniques. Also, parallelization using MPI requires a fairly specialized architecture.

A new development that can be applied more generally (a 'meta-technique' that can employ a number of different search techniques) is to iteratively break down the data set into smaller sets that are analyzed independently, and then merged again to refine the total solution. By recursively breaking down (and subsequently reassembling) the data set into smaller and smaller sets until they become manageable in reasonable time the size of the combined data set that can be analyzed is potentially much larger than any of the approaches discussed previously. An implementation of the REC-I-DCM3 algorithm (the most advanced version of the algorithms proposed to address the phylogeny problem through divide-and-conquer, Roshan *et al.*, 2004) is one of the major goals of the CIPRES (http://www.phylo.org) project. As the architecture that facilitates this algorithm becomes operational in the coming years, other algorithms, possibly hybrids of divide-and-conquer methods and parallelization, perhaps using ratcheting approaches, will become easier to implement, explore and automate.

With these powerful techniques becoming available, managing the required larger amounts of data becomes non-trivial (as I noticed during the assembly of the Primate supertree). Another priority of the CIPRES project is to create a new version of the TreeBase database (http://www.treebase.org), and in parallel a version of this database that researchers can install locally to organize and share their data more effectively. These two tracks in the CIPRES project will likely constitute a great improvement for the practice of phylogenetic inference, allowing larger and more powerful analyses. I am very pleased to be able to conclude this dissertation and contribute fulltime to these developments in this exciting era for biologists.

REFERENCES

Abbott, D. H., E. B. Keverne, F. B. Bercovitch, C. A. Shively, S. P. Medoza, W. Saltzman, C. T. Snowdon, T. E. Ziegler, M. Banjevic, T. Garland, and R. M. Sapolsky. 2003. Are subordinates always stressed? A comparative analysis of rank differences in cortisol levels among primates. Hormones And Behavior 43:67-82.

Alexander, H. J., and F. Breden. 2004. Sexual isolation and extreme morphological divergence in the Cumana guppy: a possible case of incipient speciation. Journal of Evolutionary Biology 17:1238-1254.

Altekar, G., S. Dwarkadas, J. P. Huelsenbeck, and F. Ronquist. 2004. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics 20:407-415.

Altizer, S., C. L. Nunn, P. H. Thrall, J. L. Gittleman, J. Antonovics, A. A. Cunningham, A. P. Dobson, V. Ezenwa, K. E. Jones, A. B. Pedersen, M. Poss, and J. R. C. Pulliam. 2003. Social organization and parasite risk in mammals: Integrating theory and empirical studies. Annual Review Of Ecology Evolution And Systematics 34:517-547.

Alvarez, H. P. 2000. Grandmother hypothesis and primate life histories. American Journal Of Physical Anthropology 113:435-450.

Andersen, H. L., and S. Ekman. 2005. Disintegration of the Micareaceae (lichenized Ascomycota): a molecular phylogeny based on mitochondrial rDNA sequences. Mycological Research 109:21-30.

Anderson, M. J., J. K. Hessel, and A. F. Dixson. 2004. Primate mating systems and the evolution of immune response. Journal Of Reproductive Immunology 61:31-38.

Anderson, M. J., J. Nyholt, and A. E. Dixson. 2004. Sperm competition affects the structure of the mammalian vas deferens. Journal Of Zoology 264:97-103.

Anderson, M. J., J. Nyholt, and A. F. Dixson. 2005. Sperm competition and the evolution of sperm midpiece volume in mammals. Journal Of Zoology 267:135-142.

Barraclough, T. G., A. P. Vogler, and P. H. Harvey. 1998. Revealing the factors that promote speciation. Philosophical Transactions Of The Royal Society Of London Series B-Biological Sciences 353:241-249.

Barton, R. A. 1996. Neocortex size and behavioural ecology in primates. Proceedings Of The Royal Society Of London Series B-Biological Sciences 263:173-177.

Barton, R. A. 1997. Neural constructivism: How mammals make modules. Behavioral And Brain Sciences 20:556-+.

Barton, R. A. 1998. Visual specialization and brain evolution in primates. Proceedings Of The Royal Society Of London Series B-Biological Sciences 265:1933-1937.

Barton, R. A. 2004. Binocularity and brain evolution in primates. Proceedings Of The National Academy Of Sciences Of The United States Of America 101:10113-10115.

Barton, R. A., and P. H. Harvey. 2000. Mosaic evolution of brain structure in mammals. Nature 405:1055-1058.

Baum, B. R. 1992. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon 41:3-10.

Bedoya, R. J. U., A. M. Mitzey, M. Obraztsova, and C. Lowenberger. 2005. Molecular cloning and transcriptional activation of lysozyme-encoding cDNAs in the mosquito *Aedes aegypti*. Insect Molecular Biology 14:89-94.

Bininda-Emonds, O. R. P. 2004. The evolution of supertrees. Trends In Ecology & Evolution 19:315-322.

Bininda-Emonds, O. R. P. 2004. Trees versus characters and the supertree/supermatrix "paradox". Systematic Biology 53:356-359.

Bininda-Emonds, O. R. P. 2005. Supertree construction in the genomic age. Pages 745-757 *in* Molecular Evolution: Producing The Biochemical Data, Part B.

Bininda-Emonds, O. R. P., and H. N. Bryant. 1998. Properties of matrix representation with parsimony analyses. Systematic Biology 47:497-508.

Bininda-Emonds, O. R. P., J. L. Gittleman, and S. A. Price. 2005. A complete phylogeny of the whales, dolphins and even-toed hoofed mammals (Cetartiodactyla). Biol. Rev. 80:445-473.

Bininda-Emonds, O. R. P., J. L. Gittleman, and M. A. Steel. 2002. The (Super)tree of life: Procedures, problems, and prospects. Annual Review Of Ecology And Systematics 33:265-289.

Bininda-Emonds, O. R. P., K. E. Jones, S. A. Price, R. Grenyer, M. Cardillo, M. Habib, A. Purvis, and J. L. Gittleman. 2003. Supertrees are a necessary not-so-evil: A comment on Gatesy et al. Systematic Biology 52:724-729.

Bininda-Emonds, O. R. P., and M. J. Sanderson. 2001. Assessment of the accuracy of matrix representation with parsimony analysis supertree construction. Systematic Biology 50:565-579.

Birdsey, G. M., J. Lewin, J. D. Holbrook, V. R. Simpson, A. A. Cunningham, and C. J. Danpure. 2005. A comparative analysis of the evolutionary relationship between diet and enzyme targeting in bats, marsupials and other mammals. Proceedings Of The Royal Society B-Biological Sciences 272:833-840.

Blomberg, S. P., T. Garland, and A. R. Ives. 2003. Testing for phylogenetic signal in comparative data: Behavioral traits are more labile. Evolution 57:717-745.

Bohm, M., and P. J. Mayhew. 2005. Historical biogeography and the evolution of the latitudinal gradient of species richness in the Papionini (Primata: Cercopithecidae). Biological Journal Of The Linnean Society 85:235-246.

Bokma, F. 2002. Detection of punctuated equilibrium from molecular phylogenies. Journal Of Evolutionary Biology 15:1048-1056.

Bokma, F. 2003. Testing for equal rates of cladogenesis in diverse taxa. Evolution 57:2469-2474.

Bonine, K. E., T. T. Gleeson, and T. Garland. 2005. Muscle fiber-type variation in lizards (Squamata) and phylogenetic reconstruction of hypothesized ancestral states. Journal Of Experimental Biology 208:4529-4547.

Bremer, K. 1994. Branch support and tree stability. Cladistics 10:295-304.

Bruns, T. D., and R. P. Shefferson. 2004. Evolutionary studies of ectomycorrhizal fungi: recent advances and future directions. Canadian Journal of Botany 82:1122-1132.

Carbone, C., G. Cowlishaw, N. J. B. Isaac, and J. M. Rowcliffe. 2005. How far do animals go? Determinants of day range in mammals. American Naturalist 165:290-297.

Carter, A. M. 2001. Evolution of the placenta and fetal membranes seen in the light of molecular phylogenetics. Placenta 22:800-807.

Carter, R., and K. N. Mendis. 2002. Evolutionary and historical aspects of the burden of malaria. Clinical Microbiology Reviews 15:564-+.

Chan, K. M. A., and B. R. Moore. 2002. Whole-tree methods for detecting differential diversification rates. Systematic Biology 51:855-865.

Chase, M. W., D. E. Soltis, R. Olmstead, D. Morgan, D. Les, B. D. Mishler, M. Duvall, R. Price, H. Hills, Y.-L. Qiu, K. Kron, J. Rettig, E. Conti, J. Palmer, J. Manhart, K. Sytsma, H. Michaels, W. J. Kress, M. J. Donoghue, W. D. Clark, M. Hedren, B. S. Gaut, R. Jansen, K.-J. Kim, C. Wimpee, J. Smith, G. Furnier, S. Straus, Q.-Y. Xiang, G. Plunkett, P. S. Soltis, S. Swensen, L. Eguiarte, G. Learn Jr., S. Barret, S. Graham, S. Dayanandan, and V. Albert. 1993. Phylogenetics of seed plants: an analysis of nucleotide sequences from the plastid gene *rbc*L. Ann. MO Bot. Gard. 80:528-580.

Conroy, G. C. 2003. The inverse relationship between species diversity and body mass: do primates play by the "rules"? Journal Of Human Evolution 45:43-55.

Creevey, C. J., D. A. Fitzpatrick, G. K. Philip, R. J. Kinsella, M. J. O'connell, M. M. Pentony, S. A. Travers, M. Wilkinson, and J. O. Mcinerney. 2004. Does a tree-

like phylogeny only exist at the tips in the prokaryotes? Proceedings Of The Royal Society Of London Series B-Biological Sciences 271:2551-2558.

De Ruiter, J. R. 2004. Genetic markers in primate studies: Elucidating behavior and its evolution. International Journal Of Primatology 25:1173-1189.

Deaner, R. O., and C. L. Nunn. 1999. How quickly do brains catch up with bodies? A comparative method for detecting evolutionary lag. Proceedings Of The Royal Society Of London Series B-Biological Sciences 266:687-694.

Deaner, R. O., C. L. Nunn, and C. P. Van Schaik. 2000. Comparative tests of primate cognition: Different scaling methods produce different results. Brain Behavior And Evolution 55:44-52.

Diazuriarte, R., and T. Garland. 1996. Testing hypotheses of correlated evolution using phylogenetically independent contrasts: Sensitivity to deviations from Brownian motion. Systematic Biology 45:27-47.

Diniz, J. A. F., C. E. R. De Sant'ana, and L. M. Bini. 1998. An eigenvector method for estimating phylogenetic inertia. Evolution 52:1247-1262.

Drummond, A., and K. Strimmer. 2001. PAL: An object-oriented programming library for molecular evolution and phylogenetics. Bioinformatics 17:662-663.

Eeley, H. A. C., and R. A. Foley. 1999. Species richness, species range size and ecological specialisation among African primates: geographical patterns and conservation implications. Biodiversity And Conservation 8:1033-1056.

Eulenstein, O., D. H. Chen, J. G. Burleigh, D. Fernandez-Baca, and M. J. Sanderson. 2004. Performance of flip supertree construction with a heuristic algorithm. Systematic Biology 53:299-308.

Fa, J. E., and A. Purvis. 1997. Body size, diet and population density in afrotropical forest mammals: A comparison with neotropical species. Journal Of Animal Ecology 66:98-112.

Felsenstein, J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics 5:164-166.

Fernandez, M. H., and E. S. Vrba. 2005. Body size, biomic specialization and range size of African large mammals. Journal Of Biogeography 32:1243-1256.

Fish, J. L., and C. A. Lockwood. 2003. Dietary constraints on encephalization in primates. American Journal Of Physical Anthropology 120:171-181.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. Design Patterns: Elements of Reusable Software Components. Addison-Wesley Professional, Boston, Massachussets.

Gardezi, T., and J. Da Silva. 1999. Diversity in relation to body size in mammals: A comparative study. American Naturalist 153:110-123.

Gatesy, J., R. H. Baker, and C. Hayashi. 2004. Inconsistencies in arguments for the supertree approach: Supermatrices versus supertrees of Crocodylia. Systematic Biology 53:342-355.

Gatesy, J., C. Matthee, R. Desalle, and C. Hayashi. 2002. Resolution of a supertree/supermatrix paradox. Systematic Biology 51:652-664.

Geissmann, T. 2002. Duet-splitting and the evolution of gibbon songs. Biological Reviews 77:57-76.

Geissmann, T. 2002. Taxonomy and evolution of gibbon's. Evolutionary Anthropology 11:28-31.

Gittleman, J. L., and A. Purvis. 1998. Body size and species-richness in carnivores and primates. Proceedings Of The Royal Society Of London Series B-Biological Sciences 265:113-119.

Harcourt, A. H. 2000. Latitude and latitudinal extent: a global analysis of the Rapoport effect in a tropical mammalian taxon: primates. Journal Of Biogeography 27:1169-1182.

Heesy, C. P. 2004. On the relationship between orbit orientation and binocular visual field overlap in mammals. Anatomical Record Part A-Discoveries In Molecular Cellular And Evolutionary Biology 281A:1104-1110.

Heesy, C. P., and C. F. Ross. 2001. Evolution of activity patterns and chromatic vision in primates: morphometrics, genetics and cladistics. Journal Of Human Evolution 40:111-149.

Hewitt, G., A. Maclarnon, and K. E. Jones. 2002. The functions of laryngeal air sacs in primates: A new hypothesis. Folia Primatologica 73:70-94.

Huelsenbeck, J. P., B. Larget, R. E. Miller, and F. Ronquist. 2002. Potential Applications and Pitfalls of Bayesian Inference of Phylogeny. Syst. Biol. 51:673-688.

Huelsenbeck, J. P., and F. Ronquist. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics 17:754-755.

Huelsenbeck, J. P., and F. Ronquist. 2003. MrBayes, version 3.0B4.

Isaac, N. J. B., and G. Cowlishaw. 2004. How species respond to multiple extinction threats. Proceedings Of The Royal Society Of London Series B-Biological Sciences 271:1135-1141.

Isaac, N. J. B., and A. Purvis. 2004. The 'species problem' and testing macroevolutionary hypotheses. Diversity And Distributions 10:275-281.

Jernvall, J., and P. C. Wright. 1998. Diversity components of impending primate extinctions. Proceedings Of The National Academy Of Sciences Of The United States Of America 95:11279-11283.

Larget, B., and D. L. Simon. 1999. Markov Chain Monte Carlo Algorithms for the Bayesian Analysis of Phylogenetic Trees. Mol. Biol. Evol. 16:750-759.

Lewis, R. L., A. T. Beckenbach, and A. Ø. Mooers. 2004. The phylogeny of the subgroups within the *melanogaster* species group: Likelihood tests on *COI* and *COII* sequences and a Bayesian estimate of phylogeny. Mol. Phylogenet. Evol. 37:15-24.

Maddison, D. R. 1991. The discovery and importance of multiple islands of most-parsimonious trees. Syst. Zool. 40:315-328.

Maddison, W. P., and D. R. Maddison. 2001. Mesquite: a modular system for evolutionary analysis.

Mar, J. C., T. J. Harlow, and M. A. Ragan. 2005. Bayesian and maximum likelihood phylogenetic analyses of protein sequence data under relative branch-length differences and model violation. BMC Evolutionary Biology 5.

Martin, R. D. 2000. Origins, diversity and relationships of lemurs. International Journal Of Primatology 21:1021-1049.

Mau, B., M. A. Newton, and B. Larget. 1999. Bayesian Phylogenetic Inference via Markov Chain Monte Carlo Methods. Biometrics 55:1-12.

Mooers, A. O., and S. B. Heard. 1997. Evolutionary process from phylogenetic tree shape. Quarterly Review Of Biology 72:31-54.

Purvis, A. 1995. A composite estimate of primate phylogeny. Philos. Trans. R. Soc. Lond. B Biol. Sci. 348:405-421.

Ragan, M. A. 1992. Phylogenetic inference based on matrix representation of trees. Mol. Phylogenet. Evol. 1:53-58.

Ronquist, F., and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572-1574.

Roshan, U. B. M. E. Moret, T. L. Williams, and T. Warnow, 2004. Performance of supertree methods on various dataset decompositions *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht.

Salamin, N., T. R. Hodkinson, and V. Savolainen. 2005. Towards building the Tree of Life: A simulation study for all angiosperm genera. Syst. Biol. 54:183-196.

Salter, L. A. 2001. Complexity of the likelihood surface for a large DNA dataset. Syst. Biol. 50:970-978.

Stoltzfus, A., R. A. Vos, J. Reese, and W. Qiu. 2006. Bio:CDAT - An Object for Evolutionary Analysis of Data(Draft proposal to the Open Bioinformatics Foundation).

Tombes, R. M., M. O. Faison, and J. M. Turbeville. 2003. Organization and evolution of multifunctional Ca^{2+}/CaM-dependent protein kinase genes. Gene 322:17-31.

Vinh, L. S., and A. V. Haeseler. 2004. IQPNNI: Moving Fast Through Tree Space and Stopping in Time. Mol. Biol. Evol. 21:1565-1571.

Vos, R. A. 2003. Accelerated Likelihood Surface Exploration: The Likelihood Ratchet. Syst. Biol. 52:368-373.

Vos, R. A., and A. Ø. Mooers. 2004. Reconstructing divergence times for supertrees: a molecular approach *in* Phylogenetic supertrees: combining information to reveal the Tree of Life (O. R. P. Bininda-Emonds, ed.). Kluwer Academic Press, Dordrecht.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. CABIOS 13:555-556.

Yang, Z., and B. Rannala. 1997. Bayesian Phylogenetic Inference Using DNA Sequences: A Markov Chain Monte Carlo Method. Mol. Biol. Evol. 14:717-724.

Yule, G. U. 1925. A Mathematical Theory of Evolution, based on the Conclusions of Dr. J. C. Willis, F.R.S. Philos. Trans. R. Soc. Lond. B Biol. Sci. 213:21-87.

Zuckercandl, E., and L. Pauling. 1965. Evolutionary divergence and convergence in proteins. Pages 97-165 *in* Evolving Genes and Proteins (V. Bryson, and H. J. Vogel, eds.). Academic Press, New York.

APPENDIX - BIO::PHYLO:

PHYLOGENETIC ANALYSIS USING THE PERL PROGRAMMING LANGUAGE

Rutger A. Vos

INTRODUCTION

Many different computer programs are available to perform specialized phylogenetic analyses. Unfortunately, some of these programs require their own input text files, or interpret the nexus file format (Maddison et al., 1997) in non-standard ways. This makes many types of analyses cumbersome because data files need to be edited by hand.

Here, I present software I have developed to remedy some of this: Bio::Phylo, which gives scriptable access to the entities commonly encountered in phylogenetic analyses (i.e. trees, nodes in trees, taxa, data matrices). I discuss the object model, the data model, and I present usage examples. The final section comprises detailed documentation of the application programming interface.

The Bio::Phylo package consists of a set of libraries written in perl5. By using these libraries in Perl scripts, parsing, transforming, analyzing and visualizing phylogenetic data becomes easier. Bio::Phylo is available from the comprehensive Perl archive network (http://search.cpan.org/rvosa/Bio-Phylo/). The libraries are open source software, specifically, they are licensed under the same terms as Perl itself [either the General Public License (http://www.gnu.org/copyleft/gpl.html) or the Perl Artistic License (http://www.perl.com/pub/a/language/misc/Artistic.html)].

Up to date information on the status of Bio::Phylo can be obtained from the CPAN site mentioned above; in addition, there is an online discussion forum for users (http://www.cpanforum.com/dist/Bio-Phylo) and a bug tracking server (http://rt.cpan.org/NoAuth/ReportBug.html?Queue=Bio-Phylo).

OBJECT AND DATA MODEL

The following sections discuss, in general terms, the nested objects that model phylogenetic information and entities.

THE BIO::PHYLO BASE CLASS

The Bio::Phylo object is never used directly. However, all other objects inherit from it, which means that all objects have getters and setters for their name, description, score. They can all return a globally unique ID, can all be stringified to XML, and keep track of more administrative things such as the version number of the release.

THE BIO::PHYLO::FOREST NAMESPACE

According to Bio::Phylo, there is a Forest (which is modeled by the Bio::Phylo::Forest object), which contains Bio::Phylo::Forest::Tree objects, which contain Bio::Phylo::Forest::Node objects.

Tree nodes – A node 'knows' a couple of things: its name, its branch length (i.e. the length of the branch connecting it and its parent), who its parent is, its next sister (on its right), its previous sister (on the left), its first daughter and its last daughter. Also, a taxon can be specified that the node refers to (this makes most sense when the node is terminal). These properties can be retrieved and modified by methods classified as accessors and mutators, respectively.

From this set of properties follows a number of things which must be either true or false. For example, if a node has no children it is a terminal node. By asking a node whether it "is_terminal", it replies either with true (i.e. 1) or false (undef). Methods such as this are classified as tests.

Likewise, based on the properties of an individual node one can perform a query to retrieve nodes related to it. For example, by asking the node to "get_ancestors" it returns a list of its ancestors, being all the nodes and the path from its parent to, and including, the root. These methods are queries.

Lastly, some calculations can be performed by the node. By asking the node to "calc_path_to_root" it calculates the sum of the lengths of the branches connecting it and the root. Of course, in order to make all this possible, a node has to exist, so it needs to be constructed. The constructor is the Bio::Phylo::Node->new() method.

Once a node has served its purpose it can be destroyed. For this purpose there is a destructor, which cleans up once the node reference goes out of lexical scope. However, in most cases the user does not have to worry about constructing and destroying nodes as this is done by a parser or a generator as needs arise.

For a detailed description of all the node methods, their arguments and return values, consult the node documentation, in the API section below.

Trees – A tree knows very little. All it really holds is a set of nodes, which are there because of tree population, i.e. the process of inserting nodes in the tree. The tree can be queried in a number of ways, for example, one can ask the tree to "get_entities", to which the tree replies with a list of all the nodes it holds. Be advised that this does not mean that the nodes are connected in a meaningful way, if at all. The tree does not care, the nodes are supposed to know who their parents, sisters, and daughters are. But, one can still get, for example, all the terminal nodes (i.e. the tips) in the tree by retrieving all the nodes in the tree and asking each one of them whether it "is_terminal", discarding the ones that are not.

Based on the set of nodes the tree holds it can perform calculations, such as "calc_tree_length", which means that the tree iterates over all its nodes, summing their branch lengths, and returning the total.

The tree object also has a constructor and a destructor, but these are not normally used directly. All the tree methods are listed in the API section.

Forests – The object containing all others is the Forest object. It serves merely as a container to hold multiple trees, which are inserted in the Forest object using the "insert()" method, and retrieved using the "get_entities" method. More information can be found in the API section.

THE BIO::PHYLO::MATRICES NAMESPACE

Objects in the Bio::Phylo::Matrices namespace are used to handle comparative data, as single observations, and in larger container objects.

Datum objects – The datum object holds a single observation of a predefined type, such as molecular data, or a continuous character observation. The Datum object can be linked to a taxon object, to specify which OTU the observation refers to. A 'single observation' does not imply a single character state: Datum objects can hold a DNA sequence as well - which Bio::Phylo considers a single observation.

Sequence objects – The Sequence object holds a string of characters of a predefined type, such as a molecular sequence, or a series of continuous character observations. The Sequence object can be linked to a taxon object, to specify which OTU the characters refer to. The sequence object is often more suitable for larger data sets (e.g. DNA sequences), the datum object is more memory intensive, but provides for more per

character metadata - hence it is more appropriate for individual morphological observations.

Matrix objects – The matrix object is used to aggregate datum objects into a larger, iterator object, which can be accessed using the methods of the Bio::Phylo::Listable class.

Alignment objects – The alignment object is used to aggregate sequence objects into a larger, iterator object, which can be accessed using the methods of the Bio::Phylo::Listable class.

Sets of matrices – The top level object in the Bio::Phylo::Matrices namespace is used to contain multiple matrix or alignment objects, again implementing an iterator interface.

THE BIO::PHYLO::TAXA NAMESPACE

Sets of taxa are modeled by the Bio::Phylo::Taxa object. It is a container that holds Bio::Phylo::Taxa::Taxon objects. The taxon objects at present provide no other functionality than to serve as a means of cross-referencing nodes in trees, and datum or sequence objects. This, however, is an important feature. In order to be able to write, for example, files formatted for Mark Pagel's Discrete, Continuous and Multistate (Pagel, 1994; Pagel, 1997; Pagel, 1999; Pagel, 1999) programs a taxa object, a matrix and a tree object must be cross-referenced.

Sets of taxa – The taxa object is analogous to a taxa block as implemented by Mesquite (Maddison and Maddison, 2001). Multiple matrix objects and forests can be linked to a single taxa object, using $taxa->set_matrix($matrix). Conversely, the relationship from matrix to taxa and from forest to taxa is a one-to-one relationship.

Just as forests can be linked to taxa objects, so too can indidividual node and datum objects be linked to individual taxon objects. Again, the taxon can hold references to multiple nodes or multiple datum objects, but conversely there is a one-to-one relationship. There is a constraint on these relationships: a node can only refer to a taxon that belongs to a taxa object that the forest object that contains the node references.

'IS-A' RELATIONSHIPS: INHERITANCE

The objects in Bio::Phylo are related in various ways. Some objects inherit from superclasses. Hence the object *is a* special case of the superclass. This type of relationship is shown below:

Child classes of parent Bio::Phylo:

- Bio::Phylo::Forest::Node
- Bio::Phylo::Matrices::Datum
- Bio::Phylo::Matrices::Sequence
- Bio::Phylo::Taxa::Taxon
- Bio::Phylo::Generator
- Bio::Phylo::Util::CONSTANT
- Bio::Phylo::Util::Exceptions
- Bio::Phylo::Util::IDPool
- Bio::Phylo::Treedrawer
- Bio::Phylo::Treedrawer::SVG
- Bio::Phylo::Listable, Child classes:
    - Bio::Phylo::Forest
    - Bio::Phylo::Forest::Tree

        - Bio::Phylo::Matrices
        - Bio::Phylo::Matrices::Matrix
        - Bio::Phylo::Matrices::Alignment
        - Bio::Phylo::Taxa

Child classes of parent Bio::Phylo::IO:

- Bio::Phylo::Parsers::Newick
- Bio::Phylo::Parsers::Nexus
- Bio::Phylo::Parsers::Table
- Bio::Phylo::Parsers::Taxlist
- Bio::Phylo::Unparsers::Newick
- Bio::Phylo::Unparsers::Pagel

'HAS-A' RELATIONSHIPS

Some objects contain other objects. For example, a Bio::Phylo::Forest::Tree contains Bio::Phylo::Forest::Node objects, a matrix object holds datum objects, and so on. The container objects all behave like Bio::Phylo::Listable objects: one can iterate over them (also recursively). The container relationships implemented by Bio::Phylo are as follows:

Bio::Phylo::Forest, contains:

- Bio::Phylo::Forest::Tree, contains:
    - Bio::Phylo::Forest::Node

Bio::Phylo::Matrices, contains:

- Bio::Phylo::Matrices::Alignment, contains:

        - Bio::Phylo::Matrices::Sequence

    - Bio::Phylo::Matrices::Matrix, contains

        - Bio::Phylo::Matrices::Datum

Bio::Phylo::Taxa, contains

- Bio::Phylo::Taxa::Taxon

USAGE EXAMPLES

The following sections demonstrate some of the basic functionality, with immediate, useful results.

ONE-LINERS

No concept is valid in Perl if it cannot be expressed in a one-liner. For the Bio::Phylo package, small operations can be expressed using a one-liner from the shell terminal. One-liners are commands run from the terminal using the -e '...command...' switch, invoking the Perl interpreter directly rather than from a script. The -MFoo::Bar switch is used to include library 'Foo::Bar' at runtime. (Note for windows users: in the following examples, switch the quotes around, i.e. use double quotes where single quotes are used and vice versa.) The following demonstrates this:

```
perl -MBio::Phylo::IO=parse -e 'print \
parse(-format=>"newick",-string=>"((A,B),C);") \
->first->calc_imbalance'
```

The `-MModule` switch includes an extension package. Here, the `Bio::Phylo::IO` module is used. The `-e` switch is used to evaluate the subsequent expression. The command shown here parses a string, ((A,B),C);, in the parenthetical 'Newick' format. The parser returns a `Bio::Phylo::Forest` object (i.e. a set of trees, in this case a set of one). From this set the first element is retrieved, and Colless' imbalance (Colless, 1982) is calculated, which returns a number, which is printed to standard out. This would print "1".

The next example iterates over a set of trees:

```
perl -MBio::Phylo::IO=parse -lne 'print \
parse(-format=>"newick",-string=>$_)->first \
->calc_i2' file
```

The -n switch wraps a '`while(<>){ ... }`' around the program, so the trees from *file* (that is, if they are one Newick tree description per line) are copied into `$_` one tree at a time. The `-l` switch appends a line break to the printed output.

If a tree object reference is printed, what is written is something like `Bio::Phylo::Forest::Tree=SCALAR(0x1a337dc)` (that is, the memory address that the object references). This is often not very useful, so the tree object has a `$tree->to_newick` method that stringifies the object to a Newick string. Tree descriptions can be written back to the terminal window, like so:

```
perl -MBio::Phylo::IO=parse -e 'print \
parse(-format=>"newick",-string=>"((A,B),C);") \
->first->to_newick'
```

INPUT AND OUTPUT

The `Bio::Phylo::IO` module is the unified front end for parsing and unparsing phylogenetic data objects. It is a non-OO module that optionally exports the parse and unparse subroutines into the caller's namespace, using the '`use Bio::Phylo::IO qw(parse unparse);`' directive. Alternatively, one can call

the subroutines as class methods. The 'parse' and 'unparse' subroutines load and dispatch the appropriate sub-modules at runtime, depending on the `-format` argument.

The following script demonstrates Newick tree parsing:

```
use Bio::Phylo::IO;
# get a newick string from some source
my $tree_string = '(((A,B),C),D);';
# Call class method parse from Bio::Phylo::IO.
# Newick parser returns 'Bio::Phylo::Forest'
# Call ->first to retrieve first tree of forest.
my $tree = Bio::Phylo::IO->parse(
    -string => $tree_string,
    -format => 'fastnewick'
)->first;
# prints 'Bio::Phylo::Forest::Tree'
print ref $tree, "\n";
```

The `Bio::Phylo::IO` module invokes specific parser modules. It is essentially a façade for the parsers. In the example script the `Bio::Phylo::Parsers::Fastnewick` parser turns a tree description into a `Bio::Phylo::Forest` object.

Note that there are currently two Newick parsers to choose between, 'newick' and 'fastnewick'. The former is an older implementation, which appends unique node labels to all the nodes in the tree. It is an implementation that has been tested more thoroughly. On

the other hand, 'fastnewick' has so far worked without problems. It does not introduce node labels, and parses large trees at greater speed than 'newick' (similar considerations apply to 'nexus' versus 'fastnexus').

The returned forest object subclasses `Bio::Phylo::Listable`, as a forest models a set of trees that one can iterate over. Calling the `'first'` method, returns the first tree in the forest - a `Bio::Phylo::Forest::Tree` object (in the example it's a very small forest, consisting of just this single tree).

The following example script demonstrates how to parse character-delimited tables:

```
use Bio::Phylo::IO;
# parsing a table
my $table_string = qq(A,1,2|B,1,2|C,2,2|D,2,1);
my $matrix = Bio::Phylo::IO->parse(
    -string   => $table_string,
    # see Bio::Phylo::Parsers::Table
    -format => 'table',
    # Data type
    -type     => 'STANDARD',
    -fieldsep => ',',  # field separator
    -linesep  => '|'   # line separator
);
# prints 'Bio::Phylo::Matrices::Matrix'
print ref $matrix, "\n";
```

Here the `Bio::Phylo::Parsers::Table` module parses a string `A,1,2|B,1,2|C,2,2|D,2,1`, where the '|' is a record or line separator, and the ',' is a field separator.

The following example demonstrates how to parse a list of taxa:

```
use Bio::Phylo::IO;
# parsing a list of taxa
my $taxa_string = 'A:B:C:D';
my $taxa = Bio::Phylo::IO->parse(
    -string   => $taxa_string,
    -format   => 'taxlist',
    -fieldsep => ':'
);
# prints 'Bio::Phylo::Taxa'
print ref $taxa, "\n";
```

Here the `Bio::Phylo::Parsers::Taxlist` module parses a string A:B:C:D, where the ':' is used as a field separator. The parser returns a `Bio::Phylo::Taxa` object. Note that the same result can be obtained by building the taxa object from scratch (a more feasible proposition than building trees or matrices from scratch):

```
use Bio::Phylo::Taxa;
use Bio::Phylo::Taxa::Taxon;
my $taxa = Bio::Phylo::Taxa->new;
for ( 'A', 'B', 'C', 'D' ) {
    $taxa->insert(
        Bio::Phylo::Taxa::Taxon->new(
            -name => $_
        )
    );
}
# prints 'Bio::Phylo::Taxa';
print ref $taxa, "\n";
```

Iterating

The `Bio::Phylo::Listable` module is the superclass of all container objects. Container objects are objects that contain a set of objects of the same type. For example, a `Bio::Phylo::Forest::Tree` object is a container for `Bio::Phylo::Forest::Node` objects. Hence, the `Bio::Phylo::Forest::Tree` object inherits from the `Bio::Phylo::Listable` class, and one can iterate over the nodes in a tree using the methods defined by `Bio::Phylo::Listable`. The following example demonstrates this functionality:

```
use Bio::Phylo::IO qw(parse);
my $string = '((A,B),(C,D));(((A,B),C)D);';
my $forest = parse(
    -format => 'fastnewick',
    -string => $string
);
# prints 'Bio::Phylo::Forest'
print ref $forest;
# access trees in $forest
for my $tree ( @{ $forest->get_entities } ) {
    # prints 'Bio::Phylo::Forest::Tree';
    print ref $tree;
    # access nodes in $tree
    for my $node ( @{ $tree->get_entities } ) {
        # prints 'Bio::Phylo::Forest::Node';
        print ref $node;
    }
}
```

`Bio::Phylo::Forest` and `Bio::Phylo::Forest::Tree` are nested subclasses of the iterator class `Bio::Phylo::Listable`. Nested iterator calls (such as `'get_entities'`) can be invoked on the objects.

The following example demonstrates more truly 'iterator-like' functionality of Bio::Phylo::Listable:

```
use Bio::Phylo::IO qw(parse);
my $string = 'A|B|C|D|E|F|G|H';
my $taxa = parse(
    -string   => $string,
    -format   => 'taxlist',
    -fieldsep => '|'
);
print ref $taxa; # prints 'Bio::Phylo::Taxa';
while ( my $taxon = $taxa->next ) {
    # prints 'Bio::Phylo::Taxa::Taxon'
    print ref $taxon;
}
```

A Bio::Phylo::Taxa object is a subclass of the Bio::Phylo::Listable class. Hence, one can call `'get_entities'` on the taxa object, which returns a reference to an array of taxon objects contained by the taxa object. Note however the shorthand:

```
while ( my $taxon = $taxa->next ) { ... }
```

The next example shows how objects contained by Bio::Phylo::Listable objects can be retrieved by their index in the container object:

```
use Bio::Phylo::IO;
# parsing a table
my $table_string = qq(A,1,2|B,1,2|C,2,2|D,2,1);
my $matrix = Bio::Phylo::IO->parse(
    -string   => $table_string,
     #  See Bio::Phylo::Parsers::Table
    -format   => 'table',
    -type     => 'STANDARD', # Data type
    -fieldsep => ',',        # field separator
    -linesep  => '|'         # line separator
);
# prints 'Bio::Phylo::Matrices::Matrix'
print ref $matrix, "\n";
my $datum = $matrix->get_by_index( 0, -1 );
# prints 'Bio::Phylo::Matrices::Datum'
print ref $datum;
```

The `Bio::Phylo::Matrices::Matrix` object subclasses the `Bio::Phylo::Listable` object. Hence, its iterator methods are applicable here as well. In the above example, the `get_by_index` method is used. With a single argument it returns a Bio::Phylo object. With multiple arguments the semantics are nearly identical to array slicing, except that an array reference is returned. Bio::Phylo generally passes by reference.

SIMULATING TREES

The Bio::Phylo::Generator module simulates trees under various models of clade growth. For example, here is how to generate a forest of ten Yule trees (Yule, 1925) with ten tips:

```
use Bio::Phylo::Generator;
my $gen = Bio::Phylo::Generator->new;
my $trees = $gen->gen_rand_pure_birth(
    -trees => 10,
    -tips  => 10,
    -model => 'yule'
);
print ref $trees; # prints 'Bio::Phylo::Forest'
```

The generator object simulates trees under the Yule or the Hey model. The `gen_rand_pure_birth` method call returns branch lengths drawn from the appropriate distribution, while `gen_exp_pure_birth` returns the expected waiting times (e.g. 1/n where n=number of lineages for the Yule model).

FILTERING

The objects contained by a Bio::Phylo::Listable subclass can be filtered in various ways. For example, the following retrieves the nodes no more than 2 ancestors away from the root. Any method that returns a numerical value can be specified with the '-value' argument. The '-le' flag specifies that the returned value is *l*ess-than-or-*e*qual to 2.

```
my @deep_nodes = @{
    $tree->get_by_value(
        -value => 'calc_nodes_to_root',
        -le    => 2
    )
};
```

String values that are returned by objects can be filtered using a compiled regular expression. For example, the following retrieves all nodes whose genus name matches *Eulemur*, *Lemur* or *Hapalemur*:

```
my @lemurs = @{
    $tree->get_by_regular_expression(
        -value => 'get_name',
        -match => qr/[Ll]emur_.+$/
    )
};
```

DRAWING TREES

SVG drawings of tree objects can be created using the `Bio::Phylo::Treedrawer` module:

```
use Bio::Phylo::Treedrawers; use Bio::Phylo::IO;
my $treedrawer = Bio::Phylo::Treedrawers->new(
    -width  => 400,
    -height => 600,
    -shape  => 'CURVY',
    -mode   => 'CLADO',
    -format => 'SVG'
);
my $tree = Bio::Phylo::IO->parse(
    -format => 'newick',
    -string => '((A,B),C);'
)->first;
$treedrawer->set_tree($tree);
$treedrawer->set_padding(50);
my $string = $treedrawer->draw;
```

FURTHER NOTES ON USING BIO::PHYLO

ENCAPSULATION

Unlike most other implementations of tree structures (or any other Perl objects) the Bio::Phylo objects are truly encapsulated: most Perl objects are hash references, so in most cases one can access the underlying data directly: $obj->{'key'} = 'value'. Not so for Bio::Phylo. The objects are implemented as 'Inside Out' objects. How they work exactly is outside of the scope of this document, but the implication as that the state of an object can only be changed through its methods. This is a feature that helps keep the Bio::Phylo code base maintainable as this project grows. Also, the way it is implemented is more memory-efficient and faster than the standard approach. The encapsulation forces users to use the documented interfaces of the objects. This, however, is a good thing: as long as the interfaces stay the same, any code using Bio::Phylo will continue to work, regardless of the implementation under the surface.

NAMED ARGUMENTS

When the number of arguments to a method call exceeds 1, named arguments are used. The order in which the arguments are specified does not matter, but the arguments must be all lower case and preceded by a dash:

```
use Bio::Phylo::Forest::Tree;
my $tree = Bio::Phylo::Forest::Tree->new(
    -name  => 'PHYLIP_1',
    -score => 123,
);
```

TYPE CHECKING

Argument type is always checked. Numbers are checked for being numbers; names are checked for being of allowed string format. Objects are checked for type. The only intentional exception is in object constructors, e.g. when instantiating a node, the arguments passed to the constructor are not checked:

```
use Bio::Phylo::Forest::Node;
my $node = Bio::Phylo::Forest::Node->new(
    -name => 'Node name',
    -branch_length => 0.439
);
```

This can be abused to gain a performance advantage, but the responsibility to ensure that the resulting object's internal state is sane now lies with the user.

RETURN VALUES

Apart from scalar variables, all other return values are passed by reference, either as a reference to an object or to an array or hash. Multiple return values are never returned as a list, always as an array reference:

```
my $nodes = $tree->get_entities;
#prints ARRAY.
print ref $nodes;
```

To receive nodes in `@nodes`, dereference the returned array reference (for clarity, all array dereferencing in this document is indicated by using braces in addition to the @ sigil):

```
my @nodes = @{ $tree->get_entities };
```

Mutator method calls always return the modified object, and so they can be chained:

```
$node->set_name('Homo_sapiens')->
set_branch_length(0.2343);
```

When a value requested through an accessor has not been set, the return value is 'undef'. Here one should take care what to test. For example, the value '0' is considered 'defined', but evaluates to 'false' in Boolean context:

```
# This works as expected.
# $node has no parent, hence it must be the root.
if ( ! $node->get_parent ) {
    $root = $node;
}
# The following warrants caution.
# Zero is evaluated as false-but-defined.
if ( ! $node->get_branch_length ) {
    # is there really no branch length?
    if ( defined $node->get_branch_length ) {
        # perhaps there is, but of length 0.
    }
}
```

EXCEPTIONS

The `Bio::Phylo` modules throw exceptions that subclass `Exception::Class`. Exceptions are thrown when something exceptional has happened. Not when the value requested through an accessor method is undefined. If a node has no parent, `undef` is returned. Usually, one will encounter exceptions in response to invalid input. If some method call returns an exception, wrap the call inside an 'eval' block. The error now becomes non-fatal:

```
# try something:
eval {
    $node->set_branch_length('a bad value');
};
# handle exception, if any
if ($@) {
    # do something, e.g.:
    print $@->trace->as_string; # $@ has methods
}
```

If an exception is caught, one can print a stack trace and find out what might have gone wrong starting from the script drilling into the module code.

```
# exception caught.
if (
$@->isa('Bio::Phylo::Util::Exceptions::BadNumber') ) {
    # prints stack trace in addition to error
    warn $@->error, $@->trace->as_string;
    # further metadata from exception object
    warn $@->euid, $@->uid, $@->gid, $@->pid;
    exit;
}
```

Several exception classes are defined. The type of the thrown exception should indicate what might be wrong. The types are specified in `Bio::Phylo::Util::Exceptions`.

GENERIC METADATA

One can append generic key/value pairs to any object, by calling `$obj->set_generic( 'key' => 'value');`. Subsequently calling `$obj->get_generic('key');` returns `'value'`. This is a very useful feature in many situations where one may want to attach, for example, results from analyses by outside programs (e.g. likelihood scores) to the tree objects they refer to. Likewise, multiple numbers (e.g. bootstrap values, posteriors, Bremer values) can be attached to the same node in this way.

The following subsections list all classes and their respective public methods.

BIO::PHYLO

This is the base class for the Bio::Phylo package. All other modules inherit from it; the methods defined here are applicable to all.

METHODS

new

Usage: `$phylo=Bio::Phylo->new;`
Function: Instantiates Bio::Phylo object
Returns: a Bio::Phylo object
Arguments:

```
-name    => (object name)
-desc    => (object description)
-score   => (numerical score)
-generic => (generic key/value pair)
```

set_name

Usage: `$obj->set_name($name);`
Function: Assigns an object's name.
Returns: Modified object.
Arguments: Argument must be a string, single quoted if it contains `[;|,|:\(|\)]`

set_desc

Usage: `$obj->set_desc($desc);`
Function: Assigns an object's description.
Returns: Modified object.
Arguments: Argument must be a string.

set_score

Usage: `$obj->set_score($score);`
Function: Assigns an object's numerical score.
Returns: Modified object.
Arguments: Argument must be any of perl's number formats.

set_generic

Usage: `$obj->set_generic(%hash)`
Function: Assigns generic key/value pairs to the invocant.
Returns: Modified object.
Arguments: Valid arguments constitute key/value pairs, for example:

```
$node->set_generic( '-posterior' => 0.87565 );
```

get_name

Usage: `$name=$obj->get_name;`
Function: Returns the object's name (if any).
Returns: A string

Arguments: None

get_desc

Usage: `$desc=$obj->get_desc;`
Function: Returns the object's description (if any).
Returns: A string
Arguments: None

get_score

Usage: `$score=$obj->get_score;`
Function: Returns the object's numerical score (if any).
Returns: A number
Arguments: None

get_generic

Usage: `$value=$obj->get_generic($key);`
Function: Returns the object's generic data. If an argument is used, it is considered a key for which the associated value is return. Without arguments, a reference to the whole hash is returned.
Returns: A string or hash reference.
Arguments: None required, `$key` optional

get_id

Usage: `$id=$obj->get_id;`
Function: Returns the object's unique ID
Returns: INT
Arguments: None

get

Usage: `$var_value=$obj->get($var);`
Function: Alternative syntax for safely accessing any of the object data; useful for interpolating runtime `$vars`.
Returns: (context dependent)
Arguments: a SCALAR variable, e.g. `$var='get_name';`
Comments : All objects in the package subclass the Bio::Phylo object, and so, for example, you can do `$node->get('get_branch_length');` instead of `$node->get_branch_length`. This is a useful feature for listable objects especially, as they have the `get_by_value` method, which allows you to retrieve, for instance, a list of nodes whose branch length exceeds a certain value. That method (and `get_by_regular_expression`) uses this `$obj->get` method.

clone

Usage: `$clone=$object->clone;`
Function: Creates a copy of the invocant object.
Returns: A copy of the invocant.
Arguments: none.

VERBOSE(0|1)

Usage: `$phylo->VERBOSE(0|1)`
Function: Sets/gets verbose level
Returns: Verbose level
Arguments: 0=no messages; 1=warning messages
Comments:

CITATION

Usage: `$phylo->CITATION;`
Function: Returns suggested citation.
Returns: Returns suggested citation.
Arguments: None
Comments:

VERSION

Usage: `$phylo->VERSION;`
Function: Returns version number (including CVS revision number).
Returns: SCALAR
Arguments: NONE
Comments:

to_cipres

Usage: `$xml=$obj->to_xml;`
Function: Turns the invocant object into an XML string.
Returns: SCALAR
Arguments: NONE

BIO::PHYLO::TAXA

The Bio::Phylo::Taxa object models a set of operational taxonomic units. The object subclasses the Bio::Phylo::Listable object, and so the filtering methods of that class are available. A taxa object can link to multiple forest and matrix objects.

METHODS

new

Usage: `$taxa=Bio::Phylo::Taxa->new;`
Function: Instantiates a Bio::Phylo::Taxa object.
Returns: A Bio::Phylo::Taxa object.
Arguments: none.

set_forest

Usage: `$taxa->set_forest($forest);`
Function: Associates forest with the invocant taxa object (i.e. creates reference).
Returns: Modified object.
Arguments: A Bio::Phylo::Forest object
Comments: A taxa object can link to multiple forest and matrix objects.

set_matrix

Usage: `$taxa->set_matrix($matrix);`
Function: Associates matrix with the invocant taxa object (i.e. creates reference).
Returns: Modified object.
Arguments: A Bio::Phylo::Matrices::Matrix object
Comments: A taxa object can link to multiple forest and matrix objects.

unset_forest

Usage: `$taxa->unset_forest($forest);`
Function: Disassociates forest from the invocant taxa object (i.e. removes reference).
Returns: Modified object.
Arguments: A Bio::Phylo::Forest object

unset_matrix

Usage: `$taxa->unset_matrix($matrix);`
Function: Disassociates matrix from the invocant taxa object (i.e. removes reference).
Returns: Modified object.
Arguments: A Bio::Phylo::Matrices::Matrix object

set_ntax

Usage: `$taxa->set_ntax(10);`
Function: Assigns the intended number of taxa for the invocant.
Returns: Modified object.
Arguments: Optional: An integer. If no value is given, ntax is reset to the undefined default.
Comments: This value is only necessary for the `$taxa->validate` method. If you don't need to call that, this value is better left unset.

get_forests

Usage: `@forests=@{$taxa->get_forests};`
Function: Retrieves forests associated with the current taxa object.
Returns: An ARRAY reference of Bio::Phylo::Forest objects.
Arguments: None.

get_matrices

Usage: `@matrices=@{$taxa->get_matrices};`
Function: Retrieves matrices associated with the current taxa object.
Returns: An ARRAY reference of Bio::Phylo::Matrices::Matrix objects.
Arguments: None.

get_ntax

Usage: `$ntax=$taxa->get_ntax;`
Function: Retrieves the intended number of taxa for the invocant.
Returns: An integer, or undefined.
Arguments: None.
Comments: The return value is whatever was set by the '`set_ntax`' method call. '`get_ntax`' is used by the 'validate' method to check if the computed number of taxa

matches with what is asserted here. In other words, calling `$taxa->get_ntax` doesn't return the *actual* number of taxa in the matrix, but the number it is intended to contain.

merge_by_name

Usage: `$taxa->merge_by_name($other_taxa);`
Function: Merges two taxa objects such that internally different taxon objects with the same name become a single object with the combined references to datum objects and node objects contained by the two.
Returns: A merged Bio::Phylo::Taxa object.
Arguments: A Bio::Phylo::Taxa object.

validate

Usage: `$taxa->validate;`
Function: Compares computed ntax asserted. Reacts violently if a mismatch is encountered.
Returns: Void.
Arguments: None
Comments: '`set_ntax`' needs to be assigned for this to work.

BIO::PHYLO::TAXA::TAXON

The taxon object models a single operational taxonomic unit. It is useful for cross-referencing datum objects and tree nodes.

METHODS

new

Usage: `$taxon=Bio::Phylo::Taxa::Taxon->new;`
Function: Instantiates a Bio::Phylo::Taxa::Taxon object.
Returns: A Bio::Phylo::Taxa::Taxon object.
Arguments: none.

set_data

Usage: `$taxon->set_data($datum);`
Function: Associates data with the current taxon.
Returns: Modified object.
Arguments: Must be an object of type Bio::Phylo::Matrices::Datum

set_nodes

Usage: `$taxon->set_nodes($node);`
Function: Associates tree nodes with the current taxon.
Returns: Modified object.
Arguments: A Bio::Phylo::Forest::Node object

unset_datum

Usage: `$taxon->unset_datum( $datum );`
Function: Disassociates datum from the invocant taxon (i.e. removes reference).
Returns: Modified object.
Arguments: A Bio::Phylo::Matrix::Datum object

unset_node

Usage: `$taxon->unset_node($node);`
Function: Disassociates tree node from the invocant taxon (i.e. removes reference).
Returns: Modified object.
Arguments: A Bio::Phylo::Forest::Node object

get_data

Usage: `@data=@{$taxon->get_data};`
Function: Retrieves data associated with the current taxon.
Returns: An ARRAY reference of Bio::Phylo::Matrices::Datum objects.
Arguments: None.

get_nodes

Usage: `@nodes=@{$taxon->get_nodes};`
Function: Retrieves tree nodes associated with the current taxon.
Returns: An ARRAY reference of Bio::Phylo::Trees::Node objects
Arguments: None.

BIO::PHYLO::FOREST

The Bio::Phylo::Forest object models a set of trees. The object subclasses the *Bio::Phylo::Listable* object, so look there for more methods available to forest objects.

METHODS

new

Usage: `$trees=Bio::Phylo::Forest->new;`
Function: Instantiates a Bio::Phylo::Forest object.
Returns: A Bio::Phylo::Forest object.
Arguments: None required, though see the superclass Bio::Phylo::Listable from which this object inherits.

set_taxa

Usage: `$forest->set_taxa($taxa);`
Function: Links the invocant forest object to a taxa object. Individual terminal node objects are linked to individual taxon objects by name, i.e. by what is returned by `$node->get_name`
Returns: `$forest`
Arguments: A Bio::Phylo::Taxa object.
Comments: This method checks whether any of the nodes in the trees in the invocant link to Bio::Phylo::Taxa::Taxon objects not contained by $taxa. If found, these are set to undef and the following message is displayed:

```
"Reset X references from nodes to taxa outside taxa block"
```

get_taxa

Usage: `$taxa=$forest->get_taxa;`
Function: Retrieves the taxa object linked to the invocant.
Returns: Bio::Phylo::Taxa

Arguments: NONE

to_cipres

Usage: `$cipresforest=$forest->to_cipres;`
Function: Turns the invocant forest object into a CIPRES CORBA compliant data structure
Returns: ARRAYREF
Arguments: NONE

make_taxa

Usage: `$taxa=$forest->make_taxa;`
Function: Creates a Bio::Phylo::Taxa object from the terminal nodes in invocant.
Returns: Bio::Phylo::Taxa
Arguments: NONE
Comments: N.B.!: the newly created taxa object will replace all earlier references to other taxa and taxon objects.

BIO::PHYLO::FOREST::TREE

The object models a phylogenetic tree, a container of *Bio::Phylo::Forest::Node* objects. The tree object inherits from *Bio::Phylo::Listable*, so look there for more methods.

METHODS

new

Usage: `$tree=Bio::Phylo::Forest::Tree->new;`
Function: Instantiates a Bio::Phylo::Forest::Tree object.
Returns: A Bio::Phylo::Forest::Tree object.
Arguments: No required arguments.

new_from_bioperl

Usage: `$tree=Bio::Phylo::Forest::Tree->new_from_bioperl($bptree);`
Function: Instantiates a Bio::Phylo::Forest::Tree object.
Returns: A Bio::Phylo::Forest::Tree object.
Arguments: A tree that implements Bio::Tree::TreeI

get_terminals

Usage: `@terminals=@{$tree->get_terminals};`
Function: Retrieves all terminal nodes in the Bio::Phylo::Forest::Tree object.
Returns: An array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE
Comments: If the tree is valid, this method retrieves the same set of nodes as `$node->get_terminals($root)`. However, because there is no recursion it may be faster. Also, the node method by the same name does not see orphans.

get_internals

Usage: `@internals=@{$tree->get_internals};`
Function: Retrieves all internal nodes in the Bio::Phylo::Forest::Tree object.

Returns: An array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE
Comments: If the tree is valid, this method retrieves the same set of nodes as `$node->get_internals($root)`. However, because there is no recursion it may be faster. Also, the node method by the same name does not see orphans.

get_root

Usage: `$root=$tree->get_root;`
Function: Retrieves the first orphan in the current Bio::Phylo::Forest::Tree object - which should be the root.
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_tallest_tip

Usage: `$tip=$tree->get_tallest_tip;`
Function: Retrieves the node furthest from the root in the current Bio::Phylo::Forest::Tree object.
Returns: Bio::Phylo::Forest::Node
Arguments: NONE
Comments: This method assumes the invocant tree has branch lengths.

get_mrca

Usage: `$mrca=$tree->get_mrca(\@nodes);`
Function: Retrieves the most recent common ancestor of `\@nodes`
Returns: Bio::Phylo::Forest::Node
Arguments: A reference to an array of Bio::Phylo::Forest::Node objects in $tree.

is_binary

Usage: `if ( $tree->is_binary ) { # do something }`
Function: Tests whether the invocant object is bifurcating.
Returns: BOOLEAN
Arguments: NONE

is_ultrametric

Usage: `if ( $tree->is_ultrametric(0.01) ) { # do something }`
Function: Tests whether the invocant is ultrametric.
Returns: BOOLEAN
Arguments: Optional margin between pairwise comparisons (default=0).
Comments: The test is done by performing all pairwise comparisons for root-to-tip path lengths. Since many programs introduce rounding errors in branch lengths the optional argument is available to test TRUE for nearly ultrametric trees. For example, a value of 0.01 indicates that no pairwise comparison may differ by more than 1 Note: behaviour is undefined for negative branch lengths.

is_monophyletic

Usage: `if ( $tree->is_monophyletic(\@tips, $node) ) { # do something }`
Function: Tests whether the set of `\@tips` is monophyletic w.r.t. $outgroup.

Returns: BOOLEAN
Arguments: A reference to a list of nodes, and a node.
Comments: This method is essentially the same as &Bio::Phylo::Forest::Node::is_outgroup_of.

is_clade

Usage: `if ( $tree->is_clade(\@tips) ) { # do something }`
Function: Tests whether the set of `\@tips` forms a clade
Returns: BOOLEAN
Arguments: A reference to an array of Bio::Phylo::Forest::Node objects.
Comments:

calc_tree_length

Usage: `$tree_length=$tree->calc_tree_length;`
Function: Calculates the sum of all branch lengths (i.e. the tree length).
Returns: FLOAT
Arguments: NONE

calc_tree_height

Usage: `$tree_height=$tree->calc_tree_height;`
Function: Calculates the height of the tree.
Returns: FLOAT
Arguments: NONE
Comments: For ultrametric trees this method returns the height, but this is done by averaging over all root-to-tip path lengths, so for additive trees the result should consequently be interpreted differently.

calc_number_of_nodes

Usage: `$number_of_nodes=$tree->calc_number_of_nodes;`
Function: Calculates the number of nodes (internals AND terminals).
Returns: INT
Arguments: NONE

calc_number_of_terminals

Usage: `$number_of_terminals=$tree->calc_number_of_terminals;`
Function: Calculates the number of terminal nodes.
Returns: INT
Arguments: NONE

calc_number_of_internals

Usage: `$number_of_internals=$tree->calc_number_of_internals;`
Function: Calculates the number of internal nodes.
Returns: INT
Arguments: NONE

calc_total_paths

Usage: `$total_paths=$tree->calc_total_paths;`
Function: Calculates the sum of all root-to-tip path lengths.

Returns: FLOAT
Arguments: NONE

calc_redundancy

Usage: `$redundancy=$tree->calc_redundancy;`
Function: Calculates the amount of shared (redundant) history on the total.
Returns: FLOAT
Arguments: NONE
Comments: Redundancy is calculated as:

```
1/($treelength-$height/($ntax*$height-$height))
```

calc_imbalance

Usage: `$imbalance=$tree->calc_imbalance;`
Function: Calculates Colless' coefficient of tree imbalance.
Returns: FLOAT
Arguments: NONE
Comments: As described in Colless (1982)

calc_i2

Usage: `$ci2=$tree->calc_i2;`
Function: Calculates I2 imbalance.
Returns: FLOAT
Arguments: NONE
Comments:

calc_gamma

Usage: `$gamma=$tree->calc_gamma();`
Function: Calculates the Pybus gamma statistic
Returns: FLOAT
Arguments: NONE
Comments: As described in Pybus and Harvey (2000)

calc_fiala_stemminess

Usage: `$fiala_stemminess=$tree->calc_fiala_stemminess;`
Function: Calculates stemminess measure Fiala and Sokal (1985).
Returns: FLOAT
Arguments: NONE
Comments: As described in Fiala and Sokal (1985)

calc_rohlf_stemminess

Usage: `$rohlf_stemminess=$tree->calc_rohlf_stemminess;`
Function: Calculates stemminess measure from Rohlf et al. (1990).
Returns: FLOAT
Arguments: NONE
Comments: As described in Rohlf et al. (1990)

calc_resolution

Usage: `$resolution=$tree->calc_resolution;`

Function: Calculates the total number of internal nodes over the total number of internal nodes on a fully bifurcating tree of the same size.
Returns: FLOAT
Arguments: NONE

calc_branching_times

Usage: `$branching_times=$tree->calc_branching_times;`
Function: Returns a two-dimensional array. The first dimension consists of the "records", so that in the second dimension `$AoA[$first][0]` contains the internal node references, and `$AoA[$first][1]` the branching time of the internal node. The records are orderered from root to tips by time from the origin.
Returns: SCALAR[][] or FALSE
Arguments: NONE

calc_ltt

Usage: `$ltt=$tree->calc_ltt;`
Function: Returns a two-dimensional array. The first dimension consists of the "records", so that in the second dimension `$AoA[$first][0]` contains the internal node references, and `$AoA[$first][1]` the branching time of the internal node, and `$AoA[$first][2]` the cumulative number of lineages over time. The records are orderered from root to tips by time from the origin.
Returns: SCALAR[][] or FALSE
Arguments: NONE

calc_symdiff

Usage: `$symdiff=$tree->calc_symdiff($other_tree);`
Function: Returns the symmetric difference metric between `$tree` and `$other_tree`, sensu Penny and Hendy (1985)
Returns: SCALAR
Arguments: A Bio::Phylo::Forest::Tree object
Comments: Trees in comparison must span the same set of terminal taxa or results are meaningless.

calc_fp

Usage: `$fp=$tree->calc_fp();`
Function: Returns the Fair Proportion value for each terminal
Returns: HASHREF
Arguments: NONE

calc_es

Usage: `$es=$tree->calc_es();`
Function: Returns the Equal Splits value for each terminal
Returns: HASHREF
Arguments: NONE

calc_pe

Usage: `$es=$tree->calc_pe();`

Function: Returns the Pendant Edge value for each terminal
Returns: HASHREF
Arguments: NONE

calc_shapley

Usage: `$es=$tree->calc_shapley();`
Function: Returns the Shapley value for each terminal
Returns: HASHREF
Arguments: NONE

ultrametricize

Usage: `$tree->ultrametricize;`
Function: Sets all root-to-tip path lengths equal by stretching all terminal branches to the height of the tallest node.
Returns: The modified invocant.
Arguments: NONE
Comments: This method is analogous to the 'ultrametricize' command in Mesquite, i.e. no rate smoothing or anything like that happens, just a lengthening of terminal branches.

scale

Usage: `$tree->scale($height);`
Function: Scales the tree to the specified height.
Returns: The modified invocant.
Arguments: `$height`=a numerical value indicating root-to-tip path length.
Comments: This method uses the `$tree->calc_tree_height` method, and so for additive trees the *average* root-to-tip path length is scaled to `$height` (i.e. some nodes might be taller than `$height`, others shorter).

resolve

Usage: `$tree->resolve;`
Function: Breaks polytomies by inserting additional internal nodes orderered from left to right.
Returns: The modified invocant.
Arguments:
Comments:

prune_tips

Usage: `$tree->prune_tips(\@taxa);`
Function: Prunes specified taxa from invocant.
Returns: A pruned Bio::Phylo::Forest::Tree object.
Arguments: A reference to an array of taxon names.
Comments:

keep_tips

Usage: `$tree->keep_tips(\@taxa);`
Function: Keeps specified taxa from invocant.
Returns: The pruned Bio::Phylo::Forest::Tree object.
Arguments: A list of taxon names.

Comments:

negative_to_zero

Usage: `$tree->negative_to_zero;`
Function: Converts negative branch lengths to zero.
Returns: The modified invocant.
Arguments: NONE
Comments:

exponentiate

Usage: `$tree->exponentiate($power);`
Function: Raises branch lengths to `$power`.
Returns: The modified invocant.
Arguments: A `$power` in any of perl's number formats.

log_transform

Usage: `$tree->log_transform($base);`
Function: Log `$base` transforms branch lengths.
Returns: The modified invocant.
Arguments: A `$base` in any of perl's number formats.

remove_unbranched_internals

Usage: `$tree->remove_unbranched_internals;`
Function: Collapses internal nodes with fewer than 2 children.
Returns: The modified invocant.
Arguments: NONE
Comments:

to_newick

Usage: `$string=$tree->to_newick;`
Function: Turns the invocant tree object into a newick string
Returns: SCALAR
Arguments: NONE

to_cipres

Usage: `$ciprestree=$tree->to_cipres;`
Function: Turns the invocant tree object into a CIPRES CORBA compliant data structure
Returns: HASHREF
Arguments: NONE

BIO::PHYLO::FOREST::NODE

This module defines a node object and its methods. The node is fairly syntactically rich in terms of navigation, and additional getters are provided to further ease navigation from node to node. Typical first daughter -> next sister traversal and recursion is possible, but there are also shrink-wrapped methods that return for example all terminal descendants of the focal node, or all internals, etc. Node objects are inserted into tree objects, although technically the tree object is only a container holding all the nodes together. Unless there are orphans all nodes can be reached without recourse to the tree object.

METHODS

new

Usage: `$node=Bio::Phylo::Forest::Node->new;`
Function: Instantiates a Bio::Phylo::Forest::Node object
Returns: Bio::Phylo::Forest::Node
Arguments: All optional:

```
-parent          => $parent,
-taxon           => $taxon,
-branch_length   => 0.423e+2,
-first_daughter  => $f_daughter,
-last_daughter   => $l_daughter,
-next_sister     => $n_sister,
-previous_sister => $p_sister,
-name            => 'node_name',
-desc            => 'this is a node',
-score           => 0.98,
-generic => { -posterior => 0.98, -bootstrap => 0.80 }
```

new_from_bioperl

Usage: `$node=Bio::Phylo::Forest::Node->new_from_bioperl( $bpnode );`
Function: Instantiates a Bio::Phylo::Forest::Node object from a bioperl node object.
Returns: Bio::Phylo::Forest::Node
Arguments: An objects that implements Bio::Tree::NodeI

set_taxon

Usage: `$node->set_taxon($taxon);`
Function: Assigns taxon crossreferenced with node.
Returns: Modified object.
Arguments: If no argument is given, the currently assigned taxon is set to undefined. A valid argument is a Bio::Phylo::Taxa::Taxon object.

set_parent

Usage: `$node->set_parent($parent);`
Function: Assigns a node's parent.
Returns: Modified object.
Arguments: If no argument is given, the current parent is set to undefined. A valid argument is Bio::Phylo::Forest::Node object.

set_first_daughter

Usage: `$node->set_first_daughter($f_daughter);`
Function: Assigns a node's leftmost daughter. Returns: Modified object.
Arguments: Undefines the first daughter if no argument given. A valid argument is a Bio::Phylo::Forest::Node object.

set_last_daughter

Usage: `$node->set_last_daughter($l_daughter);`

Function: Assigns a node's rightmost daughter.
Returns: Modified object.
Arguments: A valid argument consists of a Bio::Phylo::Forest::Node object. If no argument is given, the value is set to undefined.

set_previous_sister

Usage: `$node->set_previous_sister($p_sister);`
Function: Assigns a node's previous sister (to the left).
Returns: Modified object.
Arguments: A valid argument consists of a Bio::Phylo::Forest::Node object. If no argument is given, the value is set to undefined.

set_next_sister

Usage: `$node->set_next_sister($n_sister);`
Function: Assigns or retrieves a node's next sister (to the right).
Returns: Modified object.
Arguments: A valid argument consists of a Bio::Phylo::Forest::Node object. If no argument is given, the value is set to undefined.

set_child

Usage: `$node->set_child($child);`
Function: Assigns a new child to `$node`
Returns: Modified object.
Arguments: A valid argument consists of a Bio::Phylo::Forest::Node object.

set_branch_length

Usage: `$node->set_branch_length(0.423e+2);`
Function: Assigns a node's branch length.
Returns: Modified object.
Arguments: If no argument is given, the current branch length is set to undefined. A valid argument is a number in any of Perl's formats.

set_generic

Usage: `$node->set_generic( $key => $value );`
Function: Attaches a generic key => value pair to `$node`.
Returns: Modified object.
Arguments: Comma separated key => value pairs.

get_taxon

Usage: `$taxon=$node->get_taxon;`
Function: Retrieves taxon crossreferenced with node.
Returns: Bio::Phylo::Taxa::Taxon
Arguments: NONE

get_parent

Usage: `$parent=$node->get_parent;`
Function: Retrieves a node's parent.
Returns: Bio::Phylo::Forest::Node

Arguments: NONE

get_first_daughter

Usage: `$f_daughter=$node->get_first_daughter;`
Function: Retrieves a node's leftmost daughter.
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_last_daughter

Usage: `$l_daughter=$node->get_last_daughter;`
Function: Retrieves a node's rightmost daughter.
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_previous_sister

Usage: `$p_sister=$node->get_previous_sister;`
Function: Retrieves a node's previous sister (to the left).
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_next_sister

Usage: `$n_sister=$node->get_next_sister;`
Function: Retrieves a node's next sister (to the right).
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_branch_length

Usage: `$branch_length=$node->get_branch_length;`
Function: Retrieves a node's branch length.
Returns: FLOAT
Arguments: NONE
Comments: Test for "`defined($node->get_branch_length)`" for zero-length (but defined) branches. Testing "`if ( $node->get_branch_length ) { ... }`" yields false for zero-but-defined branches!

get_ancestors

Usage: `@ancestors=@{$node->get_ancestors};`
Function: Returns an array reference of ancestral nodes, ordered from young to old.
Returns: Array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE

get_sisters

Usage: `@sisters=@{$node->get_sisters};`
Function: Returns an array reference of sisters, ordered from left to right.
Returns: Array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE

get_children

Usage: `@children=@{$node->get_children};`
Function: Returns an array reference of immediate descendants, ordered from left to right.
Returns: Array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE

get_descendants

Usage: `@descendants=@{$node->get_descendants};`
Function: Returns an array reference of descendants, recursively ordered breadth first.
Returns: Array reference of Bio::Phylo::Forest::Node objects.
Arguments: none.

get_terminals

Usage: `@terminals=@{$node->get_terminals};`
Function: Returns an array reference of terminal descendants.
Returns: Array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE

get_internals

Usage: `@internals=@{$node->get_internals};`
Function: Returns an array reference of internal descendants.
Returns: Array reference of Bio::Phylo::Forest::Node objects.
Arguments: NONE

get_mrca

Usage: `$mrca=$node->get_mrca($other_node);`
Function: Returns the most recent common ancestor of `$node` and `$other_node`.
Returns: Bio::Phylo::Forest::Node
Arguments: A Bio::Phylo::Forest::Node object in the same tree.

get_leftmost_terminal

Usage: `$leftmost_terminal=$node->get_leftmost_terminal;`
Function: Returns the leftmost terminal descendant of $node.
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_rightmost_terminal

Usage: `$rightmost_terminal=$node->get_rightmost_terminal;`
Function: Returns the rightmost terminal descendant of `$node`.
Returns: Bio::Phylo::Forest::Node
Arguments: NONE

get_generic

Usage: `$generic_value=$node->get_generic($key);`
Function: Retrieves value of a generic key/value pair attached to `$node`, given `$key`. If no `$key` is given, a reference to the entire hash is returned.
Returns: A SCALAR string, or a HASH ref

Arguments: Key/value pairs are stored in a hashref. If: `$node->set_generic(posterior=>0.3543)` has been set, the value can be retrieved using `$node->get_generic('posterior');` if multiple key/value pairs were set, e.g. `$node->set_generic(x=>12,y=>80)` and `$node->get_generic` is called without arguments, a hash reference `{ x => 12, y => 80 }` is returned.

is_terminal

Usage: `if ( $node->is_terminal ) { # do something }`
Function: Returns true if node has no children (i.e. is terminal).
Returns: BOOLEAN
Arguments: NONE

is_internal

Usage: `if ( $node->is_internal ) { # do something }`
Function: Returns true if node has children (i.e. is internal).
Returns: BOOLEAN
Arguments: NONE

is_descendant_of

Usage: `if ( $node->is_descendant_of($grandparent) ) { # do something }`
Function: Returns true if the node is a descendant of the argument.
Returns: BOOLEAN
Arguments: putative ancestor - a Bio::Phylo::Forest::Node object.

is_ancestor_of

Usage: `if ( $node->is_ancestor_of($grandchild) ) { # do something }`
Function: Returns true if the node is an ancestor of the argument.
Returns: BOOLEAN
Arguments: putative descendant - a Bio::Phylo::Forest::Node object.

is_sister_of

Usage: `if ( $node->is_sister_of($sister) ) { # do something }`
Function: Returns true if the node is a sister of the argument.
Returns: BOOLEAN
Arguments: putative sister - a Bio::Phylo::Forest::Node object.

is_outgroup_of

Usage: `if ( $node->is_outgroup_of(\@ingroup) ) { # do something }`
Function: Tests whether the set of `\@ingroup` is monophyletic with respect to the $node.
Returns: BOOLEAN
Arguments: A reference to an array of Bio::Phylo::Forest::Node objects;
Comments: This method is essentially the same as &Bio::Phylo::Forest::Tree::is_monophyletic.

calc_path_to_root

Usage: `$path_to_root=$node->calc_path_to_root;`
Function: Returns the sum of branch lengths from `$node` to the root.
Returns: FLOAT
Arguments: NONE

calc_nodes_to_root

Usage: `$nodes_to_root=$node->calc_nodes_to_root;`
Function: Returns the number of nodes from `$node` to the root.
Returns: INT
Arguments: NONE

calc_max_nodes_to_tips

Usage: `$max_nodes_to_tips=$node->calc_max_nodes_to_tips;`
Function: Returns the maximum number of nodes from `$node` to tips.
Returns: INT
Arguments: NONE

calc_min_nodes_to_tips

Usage: `$min_nodes_to_tips=$node->calc_min_nodes_to_tips;`
Function: Returns the minimum number of nodes from `$node` to tips.
Returns: INT
Arguments: NONE

calc_max_path_to_tips

Usage: `$max_path_to_tips=$node->calc_max_path_to_tips;`
Function: Returns the path length from `$node` to the tallest tip.
Returns: FLOAT
Arguments: NONE

calc_min_path_to_tips

Usage: `$min_path_to_tips=$node->calc_min_path_to_tips;`
Function: Returns the path length from `$node` to the shortest tip.
Returns: FLOAT
Arguments: NONE

calc_patristic_distance

Usage: `$pd=$node->calc_patristic_distance($other_node);`
Function: Returns the patristic distance between `$node` and `$other_node`.
Returns: FLOAT
Arguments: Bio::Phylo::Forest::Node

to_xml

Usage: `$xml=$obj->to_xml;`
Function: Turns the invocant object into an XML string.
Returns: SCALAR
Arguments: NONE

BIO::PHYLO::MATRICES

The *Bio::Phylo::Matrices* object models a set of matrices. It inherits from the *Bio::Phylo::Listable* object, and so the filtering methods of that object are available to apply to a set of matrices.

METHODS

new

Usage: `$matrices=Bio::Phylo::Matrices->new;`
Function: Initializes a Bio::Phylo::Matrices object.
Returns: A Bio::Phylo::Matrices object.
Arguments: None required.

BIO::PHYLO::MATRICES::MATRIX

This module defines a container object that holds *Bio::Phylo::Matrices::Datum* objects. The matrix object inherits from *Bio::Phylo::Listable*, so the methods defined there apply here.

METHODS

new

Usage: `$matrix=Bio::Phylo::Matrices::Matrix->new;`
Function: Instantiates a Bio::Phylo::Matrices::Matrix object.
Returns: A Bio::Phylo::Matrices::Matrix object.
Arguments: NONE required, but look up the inheritance tree to the SUPER class Bio::Phylo::Listable, and its parent Bio::Phylo

set_taxa

Usage: `$matrix->set_taxa($taxa);`
Function: Links the invocant matrix object to a taxa object. Individual datum objects are linked to individual taxon objects by name, i.e. by what is returned by `$datum->get_name`
Returns: `$matrix`
Arguments: A Bio::Phylo::Taxa object.
Comments: This method checks whether any of the datum objects in the invocant link to Bio::Phylo::Taxa::Taxon objects not contained by `$matrix`. If found, these are set to undef and the following message is displayed:

```
"Reset X references from datum objects to taxa outside
taxa block"
```

set_type

Usage: `$matrix->set_type($type);`
Function: Assigns a matrix's type.
Returns: Modified object.
Arguments: `$type` must be one of [DNA|RNA|STANDARD|PROTEIN|NUCLEOTIDE|CONTINUOUS]. If no argument supplied, matrix type is set to undefined.

set_symbol

Usage: `$matrix->set_symbols($symbols);`
Function: Assigns/adds an array ref of allowed symbols
Returns: Modified object.
Arguments: A reference to an array of symbols. When no argument is given, the symbol table is reset.

set_missing

Usage: `$matrix->set_missing('?');`
Function: Assigns the missing character symbol.
Returns: Modified object.
Arguments: A symbol used to indicate missing data. Default is '?'.

set_gap

Usage: `$matrix->set_gap('-');`
Function: Assigns the gap (indel?) character symbol.
Returns: Modified object.
Arguments: A symbol used to indicate gaps. Default is '-'.

set_ntax

Usage: `$matrix->set_ntax(10);`
Function: Assigns the intended number of taxa for the matrix.
Returns: Modified object.
Arguments: Optional: An integer. If no value is given, ntax is reset to the undefined default.
Comments: This value is only necessary for the `$matrix->validate` method. If you don't need to call that, this value is better left unset.

set_nchar

Usage: `$matrix->set_nchar(10);`
Function: Assigns the intended number of characters for the matrix. Returns: Modified object.
Arguments: Optional: An integer. If no value is given, nchar is reset to the undefined default.
Comments: This value is only necessary for the $matrix->validate method. If you don't need to call that, this value is better left unset.

get_type

Usage: `$type=$matrix->get_type;`
Function: Retrieves a matrix's type.
Returns: SCALAR
`=~(DNA|RNA|STANDARD|PROTEIN|NUCLEOTIDE|CONTINUOUS);`
Arguments: NONE

get_symbols

Usage: `$symbols=$matrix->get_symbols;`
Function: Retrieves a matrix's symbol table.
Returns: ARRAY

Arguments: NONE

get_num_characters

Usage: `$nchar=$matrix->get_num_characters;`
Function: Retrieves number of characters
Returns: ARRAY
Arguments: NONE

get_num_states

Usage: `$nstates=$matrix->get_num_states;`
Function: Retrieves the number of distinct states in the matrix
Returns: SCALAR
Arguments: NONE

get_num_taxa

Usage: `$ntax=$matrix->get_num_taxa;`
Function: Retrieves the number of distinct taxa in the matrix
Returns: SCALAR
Arguments: NONE

get_taxa

Usage: `$taxa=$matrix->get_taxa;`
Function: Retrieves the Bio::Phylo::Taxa object linked to the invocant.
Returns: Bio::Phylo::Taxa
Arguments: NONE
Comments: This method returns the Bio::Phylo::Taxa object to which the invocant is linked. The returned object can therefore contain *more* taxa than are actually in the matrix.

get_chars_for_taxon

Usage: `@chars=@{$matrix->get_chars_for_taxon($taxon)};`
Function: Retrieves the datum objects for $taxon
Returns: ARRAY
Arguments: A Bio::Phylo::Taxa::Taxon object

get_cols

Usage: `$cols=$matrix->get_cols(0..100);`
Function: Retrieves columns in `$matrix`
Returns: Bio::Phylo::Matrices::Matrix (shallow copy)
Arguments: Column numbers, zero-based, throws exception if out of bounds.
Notes : This method can be used as a makeshift bootstrapper/jackknifer. The trick is to create the appropriate argument list, i.e. for bootstrapping one with the same number of elements as there are columns in the matrix - but resampled with replacement; for jackknifing a list where the number of elements is that of the number of columns to keep. You can generate such a list by iteratively calling `shift(shuffle(@list))` where shuffle comes from the List::Util package.

get_rows

Usage: `$rows=$matrix->get_rows(0..100);`
Function: Retrieves rows in $matrix Returns: Bio::Phylo::Matrices::Matrix (shallow copy)
Arguments: Row numbers, zero-based, throws exception if out of bounds.

get_missing

Usage: `$missing=$matrix->get_missing;`
Function: Retrieves the missing data symbol.
Returns: A single character.
Arguments: None.

get_gap

Usage: `$gap=$matrix->get_gap;`
Function: Retrieves the gap (indel?) character symbol.
Returns: A single character.
Arguments: None.

get_ntax

Usage: `$ntax=$matrix->get_ntax;`
Function: Retrieves the intended number of taxa for the matrix.
Returns: An integer, or undefined.
Arguments: None.
Comments: The return value is whatever was set by the '`set_ntax`' method call. '`get_ntax`' is used by the 'validate' method to check if the computed number of taxa matches with what is asserted here. In other words, this method does not return the *actual* number of taxa in the matrix (use '`get_num_taxa`' for that), but the number it is supposed to have.

get_nchar

Usage: `$matrix->get_nchar;`
Function: Retrieves the intended number of characters for the matrix.
Returns: An integer, or undefined.
Arguments: None.
Comments: The return value is whatever was set by the '`set_nchar`' method call. '`get_nchar`' is used by the '`validate`' method to check if the computed number of characters matches with what is asserted here.

validate

Usage: `$matrix->validate;`
Function: Compares computed ntax and nchar with asserted. Reacts violently if something doesn't match.
Returns: Void.
Arguments: None
Comments: '`set_ntax`' and '`set_nchar`' need to be assigned for this to work.

copy_atts

Usage: `$copy=$matrix->copy_atts;`
Function: Creates an empty copy of invocant (i.e. no data, but all the attributes).
Returns: Bio::Phylo::Matrices::Matrix (shallow copy)
Arguments: None

to_nexus

Usage: `$data_block=$matrix->to_nexus;`
Function: Converts matrix object into a nexus data block.
Alias :
Returns: Nexus data block (SCALAR).
Arguments: none
Comments:

to_cipres

Usage: `$cipres_matrix=$matrix->to_cipres;`
Function: Converts matrix object to CipresIDL
Returns: CIPRES compliant data structure
Arguments: none
Comments:

make_taxa

Usage: `$taxa=$matrix->make_taxa;`
Function: Creates a Bio::Phylo::Taxa object from the data in invocant.
Returns: Bio::Phylo::Taxa
Arguments: NONE
Comments: N.B.!: the newly created taxa object will replace all earlier references to other taxa and taxon objects.

BIO::PHYLO::MATRICES::DATUM

The datum object models a single observation or a sequence of observations, which can be linked to a taxon object.

METHODS

new

Usage: `$datum=Bio::Phylo::Matrices::Datum->new;`
Function: Instantiates a Bio::Phylo::Matrices::Datum object.
Returns: A Bio::Phylo::Matrices::Datum object.
Arguments: None required. Optional:

```
-taxon  => $taxon,
-weight => 0.234,
-type   => DNA,
-char   => ['G','A','T','T','A','C','A'],
-pos    => 2,
```

set_taxon

Usage: `$datum->set_taxon($taxon);`
Function: Assigns the taxon a datum refers to.

Returns: Modified object.
Arguments: `$taxon` must be a Bio::Phylo::Taxa::Taxon object.

set_weight

Usage: `$datum->set_weight($weight);`
Function: Assigns a datum's weight.
Returns: Modified object.
Arguments: The `$weight` argument must be a number in any of Perl's number formats.

set_type

Usage: `$datum->set_type($type);`
Function: Assigns a datum's type.
Returns: Modified object.
Arguments: `$type` must be one of [DNA|RNA|STANDARD|
PROTEIN|NUCLEOTIDE|CONTINUOUS]. If DNA, RNA or NUCLEOTIDE is defined, the subsequently set char is validated against the IUPAC nucleotide one letter codes. If PROTEIN is defined, the char is validated against IUPAC one letter amino acid codes. Likewise, a STANDARD char has to be a single integer [0-9], while for CONTINUOUS all of Perl's number formats are allowed.

set_char

Usage: `$datum->set_char($char);`
Function: Assigns a datum's character value.
Returns: Modified object.
Arguments: The `$char` argument is checked against the allowed ranges for the various character types: IUPAC nucleotide (for types of DNA|RNA|NUCLEOTIDE), IUPAC single letter amino acid codes (for type PROTEIN), integers (STANDARD) or any of perl's decimal formats (CONTINUOUS). The `$char` can be: a single character; a string of characters; an array reference of characters;
Comments: Note that on assigning characters to a datum, previously set annotations are removed.

set_position

Usage: `$datum->set_position($pos);`
Function: Assigns a datum's position.
Returns: Modified object.
Arguments: $pos must be an integer.

set_annotation

Usage: `$datum->set_annotation(-char=>1,-annotation=>{-codonpos=> 1});`
Function: Assigns an annotation to a character in the datum.
Returns: Modified object.
Arguments: Required: `-char => $int` Optional: `-annotation => $hashref`
Comments: Use this method to annotate a single character. To annotate multiple characters, use '`set_annotations`' (see below).

set_annotations

Usage: `$datum->set_annotations({-codonpos=>1},{-codonpos=>2});`
Function: Assign annotations to characters in the datum.
Returns: Modified object.
Arguments: Hash references, where position in the argument list matches that of the specified characters in the character list.
Comments: Use this method to annotate multiple characters. To annotate a single character, use '`set_annotation`' (see above).

get_taxon

Usage: `$taxon=$datum->get_taxon;`
Function: Retrieves the taxon a datum refers to.
Returns: Bio::Phylo::Taxa::Taxon
Arguments: NONE

get_weight

Usage: `$weight=$datum->get_weight;`
Function: Retrieves a datum's weight.
Returns: FLOAT
Arguments: NONE

get_type

Usage: `$type=$datum->get_type;`
Function: Retrieves a datum's type.
Returns: One of [DNA|RNA|STANDARD|PROTEIN| NUCLEOTIDE|CONTINUOUS]
Arguments: NONE

get_char

Usage: `$char=$datum->get_char;`
Function: Retrieves a datum's character value.
Returns: In scalar context, returns a single character, or a string of characters (e.g. a DNA sequence, or a space delimited series of continuous characters). In list context, returns a list of characters (of zero or more characters).
Arguments: NONE

get_position

Usage: `$pos=$datum->get_position;`
Function: Retrieves a datum's position.
Returns: a SCALAR integer.
Arguments: NONE

get_annotation

Usage: `$datum->get_annotation(-char=>1,-key=>'-codonpos');`
Function: Retrieves an annotation to a character in the datum.
Returns: SCALAR or HASH
Arguments: Optional: `-char => $int` Optional: `-key => $key`

copy_atts

Usage: `$copy=$datum->copy_atts;`
Function: Creates an empty copy of invocant (i.e. no data, but all the attributes).
Returns: Bio::Phylo::Matrices::Datum (shallow copy)
Arguments: None

reverse

Usage: `$reversed=$datum->reverse;`
Function: Reverse a datum's character string.
Returns: Reversed datum. Arguments: NONE

to_xml

Usage: `$xml=$datum->to_xml;`
Function: Reverse a datum's XML representation.
Returns: Valid XML string.
Arguments: NONE

BIO::PHYLO::MATRICES::ALIGNMENT

This module aggregates sequence objects in a larger container object. The alignment object inherits from the Bio::Phylo::Listable object, so look there for more methods applicable to alignment objects.

METHODS

new

Usage: `$alignment=Bio::Phylo::Matrices::Alignment->new;`
Function: Instantiates a Bio::Phylo::Matrices::Alignment object.
Returns: A Bio::Phylo::Matrices::Alignment object.
Arguments: NONE required.

BIO::PHYLO::MATRICES::SEQUENCE

The sequence object models a character sequence, which can be crossreferenced with a taxon object, and inserted in an alignment object.

METHODS

new

Usage: `$sequence=Bio::Phylo::Matrices::Sequence->new;`
Function: Instantiates a Bio::Phylo::Matrices::Sequence object.
Returns: A Bio::Phylo::Matrices::Sequence object.
Arguments: Optional arguments:

```
-type   => 'DNA', (a string)
-seq    => 'ACGCATCGACTACGCAG', (a string)
-taxon  => $taxon (a Bio::Phylo::Taxa::Taxon object)
```

set_taxon

Usage: `$sequence->set_taxon($taxon);`
Function: Assigns the taxon a sequence refers to.
Returns: Modified Bio::Phylo::Matrices::Sequence object.
Arguments: `$taxon` must be a Bio::Phylo::Taxa::Taxon object.

set_type

Usage: `$sequence->set_type($type);`
Function: Assigns a sequence's type.
Returns: Modified object.
Arguments: `$type` must be one of [DNA|RNA|STANDARD |PROTEIN|NUCLEOTIDE|CONTINUOUS]. If DNA, RNA or NUCLEOTIDE is defined, the subsequently set seq is validated against the IUPAC nucleotide one letter codes. If PROTEIN is defined, the seq is validated against IUPAC one letter amino acid codes. Likewise, a STANDARD seq has to be a single integer [0-9], while for CONTINUOUS all of Perl's number formats are allowed.

set_seq

Usage: `$sequence->set_seq('GATTACA');`
Function: Assigns a character string to the sequence object.
Returns: The modified invocant.
Arguments: A character string.
Comments: The string argument is checked against the allowed ranges for the various character types: IUPAC nucleotide (for types of DNA|RNA| NUCLEOTIDE), IUPAC single letter amino acid codes (for type PROTEIN), integers (STANDARD) or any of perl's decimal formats (CONTINUOUS). The character type must be specified first using the `$sequence->set_type` method.

get_taxon

Usage: `$taxon=$sequence->get_taxon;`
Function: Retrieves the taxon a sequence refers to.
Returns: Bio::Phylo::Taxa::Taxon
Arguments: NONE

get_type

Usage: `$type=$sequence->get_type;`
Function: Retrieves a sequence's type.
Returns: One of [DNA|RNA|STANDARD|PROTEIN| NUCLEOTIDE|CONTINUOUS]

Arguments: NONE

get_seq

Usage: `$string=$sequence->get_char;`
Function: Retrieves a sequence object's raw character string;
Returns: A character string.
Arguments: NONE

BIO::PHYLO::GENERATOR

The generator module is used to simulate trees under the Yule, Hey, or equiprobable model.

METHODS

new

Usage: `$gen=Bio::Phylo::Generator->new;`
Function: Initializes a Bio::Phylo::Generator object.
Returns: A Bio::Phylo::Generator object.
Arguments: NONE

gen_rand_pure_birth

Usage: `$trees=$gen->gen_rand_pure_birth(-tips=>10,-model=>'yule');`
Function: Generates markov tree shapes, with branch lengths sampled from a user defined model of clade growth, for a user defined number of tips.
Returns: A Bio::Phylo::Forest object.
Arguments:

```
-tips  => number of terminal nodes,
-model => either 'yule' or 'hey',
-trees => number of trees to generate
```

gen_exp_pure_birth

Usage: `$trees=$gen->gen_exp_pure_birth(-tips=>10,-model=>'yule');`
Function: Generates markov tree shapes, with branch lengths following the expectation under a user defined model of clade growth, for a user defined number of tips. Returns: A Bio::Phylo::Forest object.
Arguments:

```
-tips  => number of terminal nodes,
-model => either 'yule' or 'hey'
-trees => number of trees to generate
```

gen_equiprobable

Usage: `$trees=$gen->gen_equiprobable(-tips=>10,-trees=>5);`
Function: Generates an equiprobable tree shape, with branch lengths=1;
Returns: A Bio::Phylo::Forest object.
Arguments:

```
-tips  => number of terminal nodes,
```

```
-trees => number of trees to generate
```

BIO::PHYLO::IO

The IO module is the unified front end for parsing and unparsing phylogenetic data objects. It is a non-OO module that optionally exports the 'parse' and 'unparse' subroutines into the caller's namespace, using the `use Bio::Phylo::IO qw(parse unparse);` directive. Alternatively, you can call the subroutines as class methods, as in the synopsis. The parse and unparse subroutines load and dispatch the appropriate sub-modules at runtime, depending on the '-format' argument.

CLASS METHODS

The parse method makes assumptions about the capabilities of Bio::Phylo::Parsers::* modules: i) their names match those of the `-format =>` (something) arguments, insofar that `ucfirst`(something) . '`.pm`' is an existing module; ii) the modules implement a `_from_handle`, or a `_from_string` method. Exceptions are thrown if either assumption is violated.

parse

Usage: `$obj=Bio::Phylo::IO->parse( )`
Function: Creates (file) handle, instantiates appropriate parser.
Returns: A Bio::Phylo::* object

```
Arguments:
-file    => (path), or
-string  => (scalar),
-format  => (description format),
-(other) => (parser specific options)
```

unparse

Usage: `$string=Bio::Phylo::IO->unparse( );`
Function: Turns Bio::Phylo object into a string according to specified format.
Returns: SCALAR
Arguments:

```
-phylo   => (Bio::Phylo object),
-format  => (description format),
-(other) => (parser specific options)
```

BIO::PHYLO::PARSERS::FASTNEWICK

This module parses tree descriptions in parenthetical format. It is called by the *Bio::Phylo::IO* facade, don't call it directly. It is different from *Bio::Phylo::Parsers::Newick* in that it does not add unique labels to internal nodes (it does respect the ones that are already there, though) and it is about four times faster. However, it is not considered 'stable', yet (i.e. there might be bugs).

BIO::PHYLO::PARSERS::FASTNEXUS

This module parses nexus files. It is called by the *Bio::Phylo::IO* module, there is no direct usage. The parser can handle files and strings with multiple tree, taxon, and

characters blocks whose links are defined using Mesquite's "TITLE='some_name'" and "LINK TAXA='some_name'" tokens.
The parser returns a reference to an array containing one or more taxa, trees and matrices objects. Nexus comments are stripped, spaces in single quoted strings are replaced with underscores, private nexus blocks (and the 'assumptions' block) are skipped. It currently doesn't handle 'interleaved' matrices and 'mixed' data.

BIO::PHYLO::PARSERS::NEWICK

This module parses tree descriptions in parenthetical format. It is called by the *Bio::Phylo::IO* facade, don't call it directly.

BIO::PHYLO::PARSERS::NEXUS

This module parses nexus files. It is called by the *Bio::Phylo::IO* module, there is no direct usage. The parser can only handle files with a single tree, taxon, and characters block. It returns a reference to an array containing one or more taxa, trees and matrices objects.

BIO::PHYLO::PARSERS::TABLE

This module is used to import data and taxa from plain text files or strings. The following additional argument must be used in the call to Bio::Phylo::IO:

```
 -type => 'DNA'
# or RNA, STANDARD, PROTEIN, NUCLEOTIDE, CONTINUOUS
```

In addition, these arguments may be used to indicate line separators (default is "\n") and field separators (default is "\t"):

```
 -fieldsep => '\t', -linesep => '\n'
```

BIO::PHYLO::PARSERS::TAXLIST

This module is used for importing sets of taxa from plain text files, one taxon on each line. It is called by the Bio::Phylo::IO object, so look there for usage examples. If you want to parse from a string, you may need to indicate the field separator (default is '\n') to the Bio::Phylo::IO->parse call:

```
 -fieldsep => '\n',
```

BIO::PHYLO::UNPARSERS::MRP

This module turns a *Bio::Phylo::Forest* object into an MRP nexus formatted matrix. It is called by the *Bio::Phylo::IO* facade, don't call it directly.

BIO::PHYLO::UNPARSERS::NEWICK

This module turns a tree object into a newick formatted (parenthetical) tree description. It is called by the *Bio::Phylo::IO* facade, don't call it directly.

BIO::PHYLO::UNPARSERS::NEXUS

This module turns a *Bio::Phylo::Matrices::Matrix* object into a nexus formatted matrix. It is called by the *Bio::Phylo::IO* facade, don't call it directly.

BIO::PHYLO::UNPARSERS::PAGEL

This module unparses a Bio::Phylo data structure into an input file for Discrete/Continuous/Multistate. The pagel file format (as it is interpreted here) consists of:

- Line 1: the number of tips, the number of characters
- Subsequent lines: offspring name, parent name, branch length, character state(s).

During unparsing, the tree is randomly resolved, and branch lengths are formatted to %f floats (i.e. integers, decimal point, integers).
The pagel module is called by the Bio::Phylo::IO object, so look there to learn how to create Pagel formatted files.

BIO::PHYLO::TREEDRAWER

This module prepares a tree object for drawing (calculating coordinates for nodes) and calls the appropriate format-specific drawer.

METHODS

new

Usage: `$treedrawer=Bio::Phylo::Treedrawer->new;`
Function: Initializes a Bio::Phylo::Treedrawer object.
Returns: A Bio::Phylo::Treedrawer object.
Arguments: none.

set_format

Usage: `$treedrawer->set_format('svg');`
Function: Sets the drawer submodule.
Returns: Invocant.
Arguments: Name of an image format (currently only svg supported)

set_width

Usage: `$treedrawer->set_width(1000);`
Function: sets the width of the drawer canvas.
Returns: Invocant.
Arguments: Integer width in pixels.

set_height

Usage: `$treedrawer->set_height(1000);`
Function: sets the height of the canvas.
Returns: Invocant.
Arguments: Integer height in pixels.

set_mode

Usage: `$treedrawer->set_mode('clado');`
Function: Sets the tree mode, i.e. cladogram or phylogram.
Returns: Invocant.
Arguments: String, `[clado|phylo]`

set_shape

Usage: `$treedrawer->set_shape('rect');`
Function: Sets the tree shape, i.e. rectangular, diagonal or curvy.
Returns: Invocant.
Arguments: String, `[rect|diag|curvy]`

set_padding

Usage: `$treedrawer->set_padding(100);`
Function: Sets the canvas padding.
Returns: Invocant.
Arguments: Integer value in pixels.

set_node_radius

Usage: `$treedrawer->set_node_radius(20);`
Function: Sets the node radius in pixels.
Returns: Invocant.
Arguments: Integer value in pixels.

set_text_horiz_offset

Usage: `$treedrawer->set_text_horiz_offset(5);`
Function: Sets the distance between tips and text, in pixels.
Returns: Invocant.
Arguments: Integer value in pixels.

set_text_vert_offset

Usage: `$treedrawer->set_text_vert_offset(3);`
Function: Sets the text baseline relative to the tips, in pixels.
Returns: Invocant.
Arguments: Integer value in pixels.

set_text_width

Usage: `$treedrawer->set_text_width(150);`
Function: Sets the canvas width for terminal taxon names.
Returns: Invocant.
Arguments: Integer value in pixels.

set_tree

Usage: `$treedrawer->set_tree($tree);`
Function: Sets the Bio::Phylo::Forest::Tree object to unparse.
Returns: Invocant.
Arguments: A Bio::Phylo::Forest::Tree object.

set_scale_options

Usage: `$treedrawer->set_scale_options(%options);`
Function: Sets the options for time (distance) scale
Returns: Invocant,
Arguments:

```
-width => (pixels or percentage)
-major => (likewise, value for major tick marks )
-minor => (likewise, value for minor tick marks )
-label => (text string displayed next to scale )
```

get_format

Usage: `$format=$treedrawer->get_format;`
Function: Gets the image format.
Returns: SCALAR
Arguments: None.

get_width

Usage: `$width=$treedrawer->get_width;`
Function: Gets the width of the drawer canvas.
Returns: SCALAR
Arguments: None.

get_height

Usage: `$height=$treedrawer->get_height;`
Function: Gets the height of the canvas.
Returns: SCALAR
Arguments: None.

get_mode

Usage: `$mode=$treedrawer->get_mode('clado');`
Function: Gets the tree mode, i.e. cladogram or phylogram.
Returns: SCALAR, one of (CLADO|PHYLO)
Arguments: None.

get_shape

Usage: `$shape=$treedrawer->get_shape;`
Function: Gets the tree shape, i.e. rectangular, diagonal or curvy.
Returns: SCALAR, on of (RECT|CURVY|DIAG)
Arguments: None.

get_padding

Usage: `$padding=$treedrawer->get_padding;`
Function: Gets the canvas padding.
Returns: SCALAR
Arguments: None.

get_node_radius

Usage: `$node_radius=$treedrawer->get_node_radius;`
Function: Gets the node radius in pixels.
Returns: SCALAR
Arguments: None.

get_text_horiz_offset

Usage: `$text_horiz_offset=$treedrawer->get_text_horiz_offset;`
Function: Gets the distance between tips and text, in pixels.
Returns: SCALAR
Arguments: None.

get_text_vert_offset

Usage: `$text_vert_offset=$treedrawer->get_text_vert_offset;`
Function: Gets the text baseline relative to the tips, in pixels.
Returns: SCALAR
Arguments: None.

get_text_width

Usage: `$textwidth=$treedrawer->get_text_width;`
Function: Returns the canvas width for terminal taxon names.
Returns: SCALAR
Arguments: None.

get_tree

Usage: `$tree=$treedrawer->get_tree;`
Function: Returns the Bio::Phylo::Forest::Tree object to unparse.
Returns: A Bio::Phylo::Forest::Tree object.
Arguments: None.

get_scale_options

Usage: `%options=%{$treedrawer->get_scale_options};`
Function: Returns the time/distance scale options.
Returns: A hash ref.
Arguments: None.

draw

Usage: `$drawing=$treedrawer->draw;`
Function: Unparses a Bio::Phylo::Forest::Tree object into a drawing.
Returns: SCALAR
Arguments:

BIO::PHYLO::TREEDRAWER::SVG

This module creates a scalable vector graphic from a Bio::Phylo::Trees::Tree object. It is called by the *Bio::Phylo::Treedrawer* object, so look there to learn how to create tree drawings. (For extra per-node formatting, attach a hash reference to the node, like so: `$node->set_generic( 'svg' => { 'stroke' => 'red' } )`, which outlines the node, and branch leading up to it, in red.)

BIO::PHYLO::LISTABLE

A listable object is an object that contains multiple smaller objects of the same type. For example: a tree contains nodes, so it's a listable object. This class contains methods that are useful for all listable objects: Matrices, Matrix objects, Alignment objects, Taxa, Forest, Tree objects.

METHODS

new

Usage: `$obj=Bio::Phylo::Listable->new;`
Function: Instantiates a Bio::Phylo::Listable object
Returns: A Bio::Phylo::Listable object.

Arguments: none

insert

Usage: `$obj->insert($other_obj);`
Function: Pushes an object into its container.
Returns: A Bio::Phylo::Listable object.
Arguments: A Bio::Phylo::* object.

delete

Usage: `$obj->delete($other_obj);`
Function: Deletes an object from its container.
Returns: A Bio::Phylo::Listable object.
Arguments: A Bio::Phylo::* object.
Note : Be careful with this method: deleting a node from a tree like this will result in undefined references in its neighbouring nodes. Its children will have their parent reference become undef (instead of pointing to their grandparent, as collapsing a node would do). The same is true for taxon objects that reference datum objects: if the datum object is deleted from a matrix (say), the taxon will now hold undefined references.

cross_reference

Usage: `$obj->cross_reference($taxa);`
Function: The cross_reference method links node and datum objects to the taxa they apply to. After crossreferencing a matrix with a taxa object, every datum object has a reference to a taxon object stored in its `$datum->get_taxon` field, and every taxon object has a list of references to datum objects stored in its `$taxon->get_data` field.
Returns: string
Arguments: A Bio::Phylo::Taxa object
Comments: Returns a reference to an array of objects contained by the listable object.

get_entities

Usage: `@entities=@{$obj->get_entities};`
Function: Retrieves all entities in the invocant.
Returns: A reference to a list of Bio::Phylo::* objects.
Arguments: none.

contains

Usage: `if ( $obj->contains($other_obj) ) { # do something }`
Function: Tests whether the invocant object contains the argument object
Returns: BOOLEAN
Arguments: A Bio::Phylo::* object

first

Usage: `$first_obj=$obj->first;`
Function: Retrieves the first entity in the invocant.
Returns: A Bio::Phylo::* object
Arguments: none.

last

Usage: `$last_obj=$obj->last;`
Function: Retrieves the last entity in the invocant.
Returns: A Bio::Phylo::* object
Arguments: none.

current

Usage: `$current_obj=$obj->current;`
Function: Retrieves the current focal entity in the invocant.
Returns: A Bio::Phylo::* object
Arguments: none.

next

Usage: `$next_obj=$obj->next;`
Function: Retrieves the next focal entity in the invocant.
Returns: A Bio::Phylo::* object
Arguments: none.

previous

Usage: `$previous_obj=$obj->previous;`
Function: Retrieves the previous focal entity in the invocant.
Returns: A Bio::Phylo::* object
Arguments: none.

current_index

Usage: `$current_index=$obj->current_index;`
Function: Returns the current internal index of the invocant.
Returns: An integer
Arguments: none.

last_index

Usage: `$last_index=$obj->last_index;`
Function: Returns the highest valid index of the invocant.
Returns: An integer
Arguments: none.

get_by_index

Usage: `$contained_obj=$obj->get_by_index($i);`
Function: Retrieves the i'th entity from a listable object.
Returns: An entity stored by a listable object (or array ref for slices).
Arguments: An index or range. This works the way you dereference any perl array including through slices, i.e. `$obj->get_by_index(0..10)` or `$obj->get_by_index(0, -1)` and so on.
Comments: Throws if out-of-bounds

get_by_value

Usage: `@objects=@{$obj->get_by_value(-value=>$method,-ge=>$number)};`

Function: The get_by_value method can be used to filter out objects contained by the listable object that meet a numerical condition. The method iterates through all objects contained by `$obj` and returns those for which the output of `$method` (e.g. `get_tree_length`) is less than (`-lt`), less than or equal to (`-le`), equal to (`-eq`), greater than or equal to (`-ge`), or greater than (`-gt`) `$number`.
Returns: A reference to an array of objects
Arguments:

`-value` => any of the numerical obj data (e.g. tree length)
`-lt` => less than
`-le` => less than or equals
`-eq` => equals
`-ge` => greater than or equals
`-gt` => greater than

get_by_regular_expression

Usage: `@objects=@{ $obj->get_by_regular_expression(-value=>$method,-match=>$re) };`
Function: The `get_by_regular_expression` method can be used to filter out objects contained by the listable object that match a regular expression. The method retrieves the data in the current Bio::Phylo::Listable object whose $method output matches `$re`
Returns: A list of Bio::Phylo::* objects.

```
Arguments:
-value => (method returning a string, e.g. 'get_type')
-match => (compiled regex, e.g. qr/^[D|R]NA$/)
```

visit

Usage: `$obj->visit( sub{ print $_[0]->get_name, "\n" } );`
Function: The visit method can be used to iterate over all objects in the Listable object. At every iteration, the CODE reference in the argument is applied to the focal object. The object enters the CODE reference as `$_[0]`. The objects are visited in the order in which they were inserted in the Listable object.
Returns: The invocant, possibly modified.
Arguments: a CODE reference.

BIO::PHYLO::UTIL::CONSTANT

This package defines globals used in the Bio::Phylo libraries. The constants are called internally by the other packages. There is no direct usage.

BIO::PHYLO::UTIL::EXCEPTIONS

This package defines exceptions that can be thrown by other modules. There is no direct usage. The package subclasses *Exception::Class* and thus has the same methods for throwing and catching exceptions and for showing stack traces.

BIO::PHYLO::UTIL::IDPOOL

This package defines utility functions for generating and reclaiming object IDs. These functions are called by object constructors and destructors, respectively. There is no direct usage.

ACKNOWLEDGEMENTS

Many ideas from BioPerl have been incorporated in Bio::Phylo. I would like to thank Jason Stajich of the BioPerl project for making this possible: for writing excellent code and releasing it under an open source license. I am glad that there is movement afoot to allow me to reciprocate and contribute back to BioPerl in the future. As well, Aki Mimoto has been very helpful and has contributed substantial patches and extensions to Bio::Phylo.

REFERENCES

Colless, D. H. 1982. Phylogenetics: The theory and practice of phylogenetic systematics II (book review). Syst. Zool. 31:100-104.

Fiala, K. L., and R. R. Sokal. 1985. Factors determining the accuracy of cladogram estimation: evaluation using computer simulation. Evolution 39:609-622.

Maddison, D. R., D. L. Swofford, and W. P. Maddison. 1997. NEXUS: An extensible file format for systematic information. Syst. Biol. 46:590-621.

Maddison, W. P., and D. R. Maddison. 2001. Mesquite: a modular system for evolutionary analysis.

Pagel, M. 1994. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proc. R. Soc. Lond. B Biol. Sci. 255:37-45.

Pagel, M. 1997. Inferring evolutionary processes from phylogenies. Zoologica Scripta 26:331-348.

Pagel, M. 1999. Inferring the historical patterns of biological evolution. Nature 401:877-884.

Pagel, M. 1999. The maximum likelihood approach to reconstructing ancestral character states of discrete characters in phylogenies. Syst. Biol. 48:612-622.

Penny, D., and M. P. Hendy. 1985. The use of tree comparison metrics. Syst. Zool. 34:75-82.

Pybus, O. G., and P. H. Harvey. 2000. Testing macro-evolutionary models using incomplete molecular phylogenies. Proc. R. Soc. Lond. B Biol. Sci. 267:2267-2272.

Rohlf, F. J., W. S. Chang, R. R. Sokal, and J. Kim. 1990. Accuracy of Estimated Phylogenies: Effects of Tree Topology and Evolutionary Model. Evolution 44:1671-1684.

Yule, G. U. 1925. A Mathematical Theory of Evolution, based on the Conclusions of Dr. J. C. Willis, F.R.S. Philos. Trans. R. Soc. Lond. B Biol. Sci. 213:21-87.