Purvis, A. (1995). A composite estimate of primate phylogeny. *Philos. Trans. R. Soc. Lond. B* **348,** 405–421.

Ragan, M. A. (1992). Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.* **1,** 53–58.

Ronquist, F. (1996). Matrix representation of trees, redundancy, and weighting. *Syst. Biol.* **45,** 247–253.

Ronquist, F., Huelsenbeck, J. P., and Britton, T. (2004). *In* ‘‘Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life’’ (O. R. P. Bininda-Emonds, ed.), Vol. 3, pp. 193–224. Kluwer Academic, Dordrecht, the Netherlands. Au_C38_11

Roshan, U., Moret, B. M. E., Williams, T. L., and Warnow, T. (2004). *In* ‘‘Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life’’ (O. R. P. Bininda-Emonds, ed.), Vol. 3, pp. 301–328. Kluwer Academic, Dordrecht, the Netherlands. Au_C38_12

Salamin, N., Hodkinson, T. R., and Savolainen, V. (2002). Building supertrees: An empirical assessment using the grass family (Poaceae). *Syst. Biol.* **51,** 136–150.

Sanderson, M. J., Purvis, A., and Henze, C. (1998). Phylogenetic supertrees: Assembling the trees of life. *Trends Ecol. Evol.* **13,** 105–109.

Sanderson, M. J., Driskell, A. C., Ree, R. H., Eulenstein, O., and Langley, S. (2003). Obtaining maximal concatenated phylogenetic data sets from large sequence databases. *Mol. Biol. Evol.* **20,** 1036–1042.

Semple, C., and Steel, M. (2000). A supertree method for rooted trees. *Discrete Appl. Math.* **105,** 147–158.

Slowinski, J. B., and Page, R. D. M. (1999). How should species phylogenies be inferred from sequence data? *Syst. Biol.* **48,** 814–825.

Steel, M. (1992). The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classif.* **9,** 91–116.

Wilkinson, M. (1995). Coping with abundant missing entries in phylogenetic inference using parsimony. *Syst. Biol.* **44,** 501–514.

Zimmer, E. A., White, T. J., Cann, R. L. and Wilson, A. C. (eds.) (1993). ‘‘Molecular Evolution: Producing the Biochemical Data.’’ Au_C38_13

[39] Maximum Likelihood Methods for Phylogeny Estimation

By JACK SULLIVAN

Abstract

Maximum likelihood (ML)estimation of phylogenies has reached a rather high level of sophistication because of algorithmic advances, improvements in models of sequence evolution, and improvements in statistical approaches and application of cluster computing. Here, I provide a brief basic background in application of the general principle of ML estimation to phylogenetics and provide an example of selecting among a nested set of ML models using a dynamic approach to hierarchical

0076-6879/05 $35.00

Comp. by:jaykumar Date:4/3/05 Time:21:32:35 Stage:First Proof File Path://fsc/
serials/PRODENV/000000~1/000BD5~1/S00000~1/000000~3/000000~3/
000006192.3D Proof by:N. Pushparaj QC by:Thiru ProjectAcronym:MIE
Volume:395 ChapterNo:0035

likelihood ratio tests. I focus attention on PAUP* because it provides unique ease of switching among alternative optimality criteria (e.g., minimum evolution, parsimony, and ML). Further, examples of parametric bootstrap tests are provided that demonstrate statistical tests of phylogenetic hypotheses and model adequacy, in an absolute rather than relative sense. The increasing availability of clustered, parallelized computation makes use of such parametric approaches feasible.

Application of ML as an Optimality Criterion in Phylogeny Estimation

Maximum-likelihood (ML) estimation is a standard and useful statistical procedure that has become widely applied to phylogenetic analysis. Although this application of ML presents some unique issues, the general idea is the same in phylogeny as in any other application. One calculates the likelihood of an observed dataset given a particular hypothesis and some assumed probabilistic model.

$$L = \text{Prob}(\text{data}|\text{hypothesis}) \tag{1}$$

We evaluate several hypotheses and select the one that maximizes the probability of generating the observed data. When applied to phylogeny estimation, the hypotheses that are examined represent alternative phylogenies and the data are the set of aligned sequences. The likelihood of a tree (τ) is

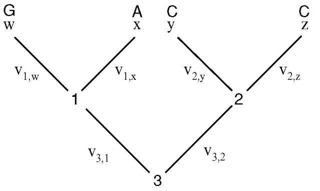
$$L(\tau) = \text{Prob}(\text{D}|\tau) \tag{2}$$

simply the probability of the data (the set of aligned sequences), given the tree (and some assumed model of character evolution). Just as the length of a tree can be calculated as its optimality score in parsimony analyses, the likelihood of a tree can be used as its optimality score in ML estimation. We make the assumption that characters are independent (just as in parsimony) so that we may treat likelihoods for each site separately:

$$\text{L}(\tau) = \prod_{\text{i}=1}^{\text{s}} \text{Prob}(\text{D}^{\text{i}}|\tau) = \prod_{\text{i}=1}^{\text{s}} \text{L}^{\text{i}}(\tau) \tag{3}$$

where s is the number of sites (characters) and $\text{Prob}(\text{D}^{\text{i}} \mid \tau)$ is the probability of site i (character i), given tree τ. This value is the single-site likelihood, and just as the parsimony score for a tree across an entire dataset is the sum of the character lengths, the likelihood of a tree across an entire dataset is the product of the single-site likelihoods. The single-site likelihood is, therefore, analogous to the length of a most parsimonious character reconstruction in MP estimation.

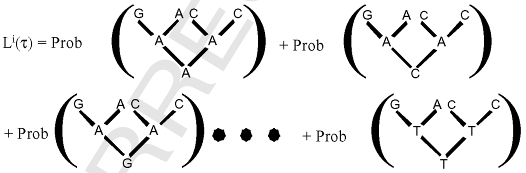
The calculation of single-site likelihoods is accomplished as follows. Let us assume the following rooted, four-taxon tree:



This example is somewhat modified from that provided by Swofford *et al.* (1996), and in this tree, taxa w, x, y, and z have nucleotides G, A, C, and C, respectively, at the first position in the alignment. The branches, which are labeled $v_{x,y}$, and their lengths (in units of expected number of substitutions per site—a function of rate of evolution times the temporal duration of branch) are parameters that need to be estimated. So to calculate the single-site likelihood for this character, we must sum the probabilities for all possible character-state reconstructions. Because there are $n - 1 = 3$ internal nodes (for a rooted tree) and four possible character states at each node, there are $4^{n-1} = 4^3 = 64$ possible reconstructions. So

$$L^i(\tau) = \sum_{r}^{4n-1} \mathrm{Prob}(R^i_r|\tau)$$

or



Of course many reconstructions are extremely unlikely (such as the last one shown, with T at all internal nodes), and they will contribute very little to the single-site likelihood; nevertheless, we consider them as possibilities. So now the issue is how one calculates the probabilities of a particular reconstruction. Let us assume that in reconstruction *r*, *m* is the state at the root node 3, *k* is the state at node 1, and *l* is the state at node 2. We know from our data that nodes w, x, y, and z have states A, G, C, and C, respectively. So the probability of reconstruction *r* at site *i* is

$$P(R^i_r|\tau) = \pi_m \times P_{m,k}(v_{3,1}) \times P_{k,\mathrm{A}}(v_{1,\mathrm{w}}) \times P_{k,\mathrm{G}}(v_{1,\mathrm{x}}) \times P_{m,l}(v_{3,2}) \times P_{l,\mathrm{C}}(v_{2,\mathrm{y}}) \times P_{l,\mathrm{C}}(v_{2,\mathrm{z}}), \quad (5)$$

where π_m is the frequency of the nucleotide A (which provides an estimate of the probability of observing state m at the root node). $P_{i,j}$ is the probability of substitution between states *i* and *j,* which is derived from the model of sequence evolution that we assume (see below). This can be calculated for all $4^{\mathrm{n}-1}$ reconstructions, and these are then summed across reconstructions to calculate the single-site likelihoods. However, Felsenstein (1973) developed a more efficient way to calculate the same value, which uses the structure of the tree, so that the single-site likelihood for character *i* is as follows:

$$\begin{aligned} L^i(\tau) = \quad & \sum_m \pi_m \ \mathrm{x} \left(\sum_k P_{m,k}(v_{3,1}) P_{k,\mathrm{A}}(v_{1,\mathrm{w}}) P_{k,\mathrm{G}}(v_{1,\mathrm{x}}) \right) \\ & \qquad \mathrm{x} \left(\sum_l P_{m,l}(v_{3,2}) P_{l,\mathrm{C}}(v_{2,\mathrm{y}}) P_{l,\mathrm{C}}(v_{2,\mathrm{z}}) \right) \end{aligned} \tag{6}$$

and each summation is across all four nucleotides. The improvements in efficiency achieved here are attributable to the fact that we can calculate the contributions of various subtrees (indicated by the structure of the parentheses) just once and use the subtree values as they are needed. In almost all applications of ML estimation, rather than dealing with the product of extremely small numbers (the single-site likelihoods), their natural logarithms are usually taken and summed. The overall log likelihood of a tree is, therefore,

$$InL(\tau) = \sum_{i=1}^{s} InL^i(\tau) \tag{7}$$

An additional efficiency is achievable by realizing that if more than one site has the same distribution of character states (i.e., has the same site pattern), we only have to calculate $L^i(\tau)$ once. So, if we consider the following dataset:

```
1         A  G  T  A  C  A  .  .  .  .  .  .  .  .  .  .  .  .
2         A  G  T  A  .  .  .  .  .  .  .  .  .  .  .  .  .  .
3         A  G  T  A  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.         .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.         .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
n         A  G  T  A  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Pattern   1  2  3  1  .  .  .  .  .  .  .  .  .  .  .  .  .  . (a)
```

The first and fourth sites have the same site pattern, and with *n* sequences, there are 4^n possible site patterns (because there are four

Comp. by:jaykumar Date:4/3/05 Time:21:32:36 Stage:First Proof File Path://fsc/
serials/PRODENV/000000~1/000BD5~1/S00000~1/000000~3/000000~3/
000006192.3D Proof by:N. Pushparaj QC by:Thiru ProjectAcronym:MIE
Volume:395 ChapterNo:0035

possible nucleotides). Rather than recalculating the single-site likelihood for sites with the same pattern, their values are stored and the frequency of each site pattern is tallied. The overall likelihood score for a particular tree therefore is as follows:

$$InL(\tau) = \sum_{a=1}^{4^n} f_a(InL^a(\tau)) \tag{8}$$

Here, f_a is the frequency of the ath site pattern and $lnL^a(\tau)$ is the single-site log likelihood of tree τ for all sites with pattern a. This value represents a measure of fit between the tree and the data (assuming a model of sequence evolution), and $lnL(\tau)$ is calculated for every tree that is examined during a tree search.

Generating ML Estimates of Phylogeny

Justification for Iterative Approach

In principle, searching trees under the likelihood criterion is no different than doing so under parsimony. However, one qualification is that the optimality score for a given tree under likelihood $lnL^i(\tau)$ is computationally more difficult than the corresponding value (tree length) under parsimony. Furthermore, the $P_{i,j}$ values used in calculating $lnL^i(\tau)$ represent instantaneous rates of substitution from nucleotide i to nucleotide j; these are specified by the model of sequence evolution, and a model must be chosen that makes explicit assumptions. One difficulty is that the optimum values of these parameters are conflated both with each other (i.e., non-independent) and with topology. The ideal solution is to simultaneously optimize all parameters on every tree that one examines during a tree search, an approach that is not feasible for most empirical studies. Fortunately, this problem can be circumvented by adopting an iterative approach (Sullivan *et al*., 1996; Sullivan and Swofford, 1997; Swofford *et al*., 1996), in which one uses a rapid approximate method to find a reasonable initial tree. This initial tree is used both to evaluate alternative models (Frati *et al*., 1997; Sullivan *et al*., 1997) and to derive initial estimates of model parameters (such as $P_{i,j}$ parameters). In the next step, the model parameters are held constant and alternative trees are evaluated (usually using some heuristic search). This process is repeated until the same tree (or set of trees) is found in successive iterations. Sullivan and Au_C39_1 Swofford (in preparation) have demonstrated that the iterative method is a useful approximation to the ideal analytical approach.

Starting the Iteration: Selecting a Model

As is true for many statistical methods, there are a number of approaches to model selection, and theory is continually being developed and tested by statisticians. With respect to ML models for phylogeny estimation, the most commonly employed approach is likelihood ratio tests (LRTs) and there are a number of ways to implement LRTs. Model selection via LRTs can be accomplished in an automated fashion, with programs such as ModelTest (Posada and Crandall, 1998), or it can be conducted in a more interactive fashion (sometimes called *dynamic model selection*). Regardless of how one chooses to proceed, LRTs require that the models being examined form a nested family of models, whereby every model in the

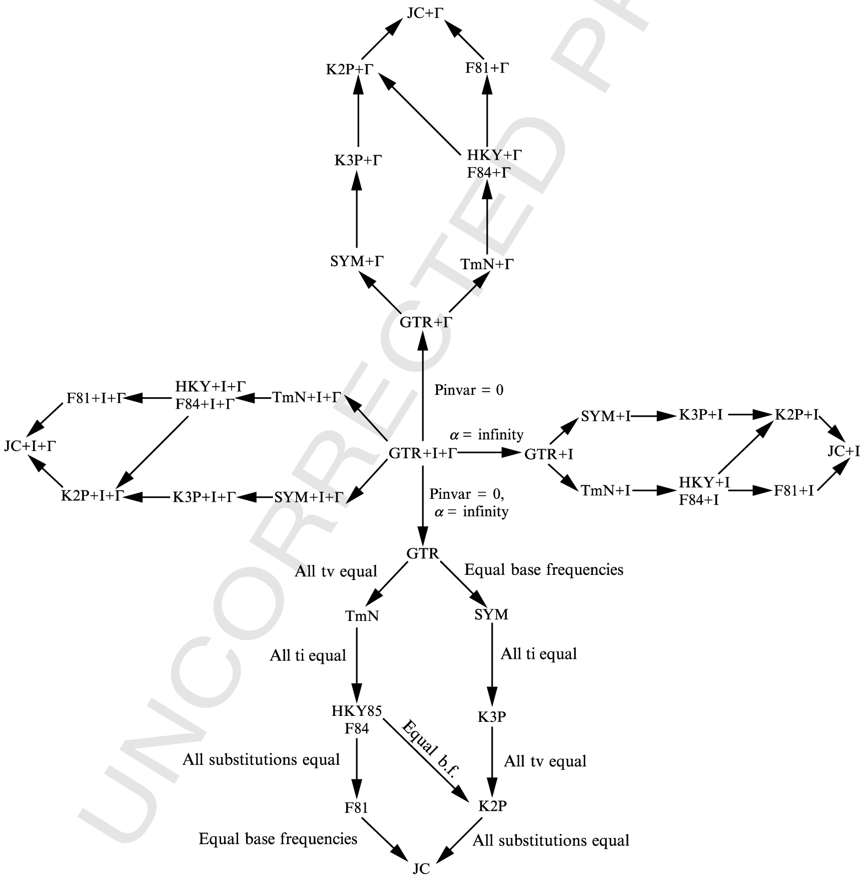


FIG. 1.

collection is a special case of some parameter-rich general model. In most phylogenetic analyses, attention has focused on the GTR+I+Γ family of Au_C39_2 models. One visualization of that family is shown above (Fig. 1).

Here, the most general and parameter-rich model, the GTR+I+Γ has 10 free parameters: three free base frequencies, 5 free relative instantaneous transformation rates (r_{AC}, r_{AG}, r_{AT}, r_{CG}, and r_{CT}; r_{GT} is arbitrarily set to 1), a proportion of the sites that are invariable (p_{inv}), and the gamma distribution shape parameter (α, which describes rate variation across the potentially variable sites). I focus on describing the steps one takes in using PAUP* (Swofford, 1998) to implement the iterative search strategy, because no other package allows one to switch optimality criteria in the same run as easily as PAUP* (Swofford and Sullivan, 2003). The initial step in implementing the iterative searches is to generate an initial tree. Usually, this is accomplished with a very rapid method such as neighbor joining (NJ), typically applied to a distance matrix generated with LogDet distances. Below, I go through a dynamic, top-down approach to model selection (i.e., starting with the most general and parameter-rich model, GTR+I+Γ) for a dataset containing 22 cytochrome *b* sequences from Au_C39_3 sigmodontine rodents (Rinehart *et al.*, unpublished). Once the dataset is loaded, the commands are as follows:

```
dset dist=logdet;
nj;
lset nst=6 rmat=est basefreq=est pinv=est rates=gamma
  shape=est; [This sets the likelihood model to GTR+I+Γ]
lscore;
```

This generates the following output:

```
Tree                        1
---------------------------
-ln L              6449.72254
Base frequencies:
A                    0.360254
C                    0.316726
G                    0.099847
T                    0.223173
Rate matrix R:
AC                    1.29169
AG                    5.43624
AT                    3.14761
CG                    0.13416
CT                   35.49652
GT                    1.00000
P_inv                0.446391
Shape                0.596201
```

Just by looking at these parameter estimates, $r_{AC} \sim r_{GT}$, we should be able to equate r_{AC} with r_{GT} using the `rclass` command. One would type the following command:

```
lscore / rclass = (a b c d e a);
```

And the output is

```
Tree                 1
--------------- --
-ln L       6450.04916
Base  frequencies:
A             0.379897
C             0.324175
G             0.066497
T             0.229431
Rate  matrix  R:
AC             1.00000
AG            10.62273
AT             1.95152
CG             0.44845
CT            28.84545
GT             1.00000
P_inv         0.456866
Shape         0.517319
```

So by eliminating one parameter, we have decreased the likelihood score by 0.326 *lnL* units. The LRT statistic is $\delta = 2(lnL_{\text{general}} - lnL_{\text{restricted}})$, so $\delta = 0.652$ and, making assumptions regarding asymptotic properties, we can use the χ^2 distribution with d.f. equal to the difference in number of parameters between the two models (one in this case). Thus, $p = .419$ and we accept the null hypothesis that there is no significant difference in fit between the two models; that is, we accept the simpler model. Now let us see if we can simplify further. The next most similar relative rate parameter is perhaps the r_{CG}, so we can restrict the matrix further.

```
lscore / rclass = (a b c a d a);
```

And the output is

```
Tree                 1
--------------- --
-ln L       6450.32915
Base  frequencies:
A             0.383239
C             0.324820
G             0.062259
T             0.229682
```

```
Rate  matrix  R:
AC           1.00000
AG          14.46313
AT           2.18638
CG           1.00000
CT          32.44806
GT           1.00000
P_inv       0.459870
Shape       0.512830
```

Again, we see just a slight deterioration in the likelihood score of 0.28 lnL units, so $\delta = 0.560$ and $p = .454$, so again we can accept the simpler model. Let us continue by setting r_{AT} equal as well:

```
lscore / rclass = ( a  b  a  a  c  a);
```

And the output is

```
Tree                 1
--------------- --
-ln  L      6452.67603
Base  frequencies:
A             0.383349
C             0.323744
G             0.059190
T             0.233717
Rate  matrix  R:
AC             1.00000
AG            12.37828
AT             1.00000
CG             1.00000
CT            23.66113
GT             1.00000
P_inv         0.459595
Shape         0.499518
```

Now we have a deterioration in likelihood score of 2.38 lnL units and a $\delta = 4.772$. This corresponds to a p value of .028926, and we can reject the simpler model. It is unlikely that any further simplifications of the R matrix would be acceptable, so let us try to simplify the rate heterogeneity among sites by looking at $\alpha =$ infinity (this is equivalent to an equal rates model):

```
lscore / rates = equal;
```

Again, the output is

```
Tree                1
-------------- --
-ln L      6673.98739
Base frequencies:
A             0.305462
C             0.335057
G             0.097600
T             0.261880
Rate matrix R:
AC             1.00000
AG             4.12118
AT             1.95195
CG             1.00000
CT             6.94868
GT             1.00000
P_inv         0.525604
```

This represents a huge deterioration of the likelihood score of about 223.7, for a single parameter, so clearly we cannot use an invariable-sites model alone (and in this case, it really does not matter which null distribution we use: The mixed χ^2 [following Goldman and Whelan, 2000] or the χ^2 with 1 d.f.). The base frequencies are so wildly different that it is pointless to even try restricting basefreq = equal. So now we are at a point at which we have three free base frequencies, three free relative-rate parameters (two transitions and two transversions, one of which is set to one), and two rate heterogeneity parameters. This actually is not a named model, and incidentally, it is not a model that any automated model selection method, such as ModelTest (Posada and Crandall, 1998), would choose. This is not to say that the model chosen here is correct, and an alternative that may be chosen by an alternative implementation of an LRT (or some other criterion) is incorrect. All models are approximations of the true underlying process, and the model selected above is a slightly different approximation of the unknown true model.

Nevertheless, one disadvantage of using automated model selection programs is that one is restricting the outcome of model selection to those cases that are hard-coded into the programs. Another advantage of the dynamic approach shown above is that the act of simplifying models manually generates a much better understanding of one's data (and indeed of the relationships among alternative models) than can be achieved by relying on an automated model-selection approach. Note that there are model-selection approaches other than LRTs, such as Akaike Information Content (AIC), Bayesian Information Criterion (BIC), and a new method based on decision theory (DT-ModSel; Minin *et al*., 2003). The ability of different model-selection approaches to select an adequate approximation to the unknown true generating process is a question that is being addressed by a number of groups. At this point, there is evidence that at least one of these, DT-ModSel (Minin *et al*., 2003), will outperform the LRT (as commonly implemented by ModelTest) with respect to accuracy of branch length estimated under the selected models (Minin *et al*., 2003).

Searching Tree Space Using the Chosen Model

Now that we have selected a model, we can begin the process of searching tree space. The following set of commands implements an ML heuristic search, under the fully defined model we have chosen, with 10 replicate searches, each started with a starting tree generated from random addition sequence and using TBR branch swapping.

```
dset dist = logdet;
nj;
lset nst = 6 rclass = (abcada) rmat = est basefreq = est pinv = est
  rates = gamma shape = est;
lscore;
lset rmat = prev basefreq = prev pinv = prev shape = prev;
set criterion = like;
hs addseq = random nrep = 10;
```

Although most systematists who employ this approach stop there, the process really should be iterated. One would use the ML tree just generated to reoptimize model parameters (unless the NJ tree from LogDet distances is identical to the ML tree, which it almost never is), and then conduct another ML search with the refined parameter estimates. This iteration should continue until the same tree or set of trees is generated in successive iterations.

One potential pitfall that can occur when there is a strong rate heterogeneity among sites is that the default number of rate categories (ncat = 4) may overly discretize the gamma distribution (which is a continuous function). This default was selected following Yang (1993), but it is definitely worth assessing the influence of varying the number of rate categories. The indication of such a problem is an estimate of the shape parameter α that is diverging toward zero; the PAUP* output is

```
Shape < 0.001
```

Comp. by:jaykumar Date:4/3/05 Time:21:32:37 Stage:First Proof File Path://fsc/
serials/PRODENV/000000~1/000BD5~1/S00000~1/000000~3/000000~3/
000006192.3D Proof by:N. Pushparaj QC by:Thiru ProjectAcronym:MIE
Volume:395 ChapterNo:0035

If this occurs, more than four rate categories are required, and usually eight (ncat = 8) are sufficient. The `ncat` setting is an option under the `lset` command. This problem is restricted to models in which all sites are assumed to be potentially variable (i.e., pinv = 0).

Assessing Phylogenetic Uncertainty

Again, just as is true for parsimony, non-parametric bootstrap analysis (Felsenstein, 1985) can be used to assess nodal support in ML analyses. Two issues need to be confronted in conducting ML bootstrap analyses. The first of these is that, ideally, one would reevaluate the relative fit of alternative models of sequence evolution and reoptimize model parameters for each pseudo-replicate. This is almost never done because of the computational limits involved. Instead, ML bootstraps are almost always conducted with the model fully defined and fixed to that which was selected in the analyses of the real data. The effect that use of an incorrectly defined model in analyzing each pseudo-replicate dataset will have on bootstrap values has not been directly studied. However, phylogenetic theory (Waddell, 1995) predicts that bootstrap values for nodes that are poorly supported (i.e., are defined by a short internal branch or a long one defined only by change at high-rate sites) will be underestimated, whereas bootstrap values for well-supported nodes should be relatively unaffected.

Au_C39_4

The second issue is also related to computational time. In any application of a bootstrap analysis, one would ideally analyze the pseudo-replicate datasets identically to how the original dataset was analyzed. Thus, one would ideally conduct a multiple random addition heuristic searches with TBR branch swapping on each pseudo-replicate. Such an approach is particularly problematic for datasets where divergence is relatively low because of the chance of constructing a pseudo-replicate that has little or no phylogenetic information. In such cases, the bootstrap analysis will become bogged down by swapping interminably on a particularly information-poor replicate. The most extreme approximation that can be taken is to omit branch swapping altogether and simply use the stepwise-addition tree as the estimate for each pseudo-replicate. In PAUP*, this is accomplished as follows (assuming the optimality criterion has been set to likelihood and the model has been fully defined previously):

```
bootstrap nrep = 1000 search = faststep;
```

A much less extreme approach that still achieves great time savings is to conduct full TBR branch swapping but to only hold a single optimal tree in memory. This can be accomplished with the following:

Comp. by:jaykumar Date:4/3/05 Time:21:32:37 Stage:First Proof File Path://fsc/
serials/PRODENV/000000~1/000BD5~1/S00000~1/000000~3/000000~3/
000006192.3D Proof by:N. Pushparaj QC by:Thiru ProjectAcronym:MIE
Volume:395 ChapterNo:0035

```
    set maxtrees = 1 increase = no;
    bootstrap nrep = 1000 search = hs keepall = yes;
```

DeBry and Olmstead (2000) and Mort *et al*. (2000) have demonstrated that this approach will provide bootstrap values that are not significantly different than those that would be attained by full heuristic searches for large datasets under the parsimony criterion. The same should hold true for ML bootstraps.

Hypothesis Testing

Finding Trees Constrained to Fit Hypotheses

Perhaps the greatest advance in systematic biology over the last 10 years is the development of explicitly statistical approaches to phylogenetic hypothesis testing. Many hypotheses in evolutionary biology make specific predictions about phylogenetic relationships, and these predicted relationships form the basis of phylogenetic hypothesis testing. The idea is that the ML (or MP) tree for a particular dataset may contradict the relationships predicted by some hypothesis one wants to test. By using topological constraints, one may assess how much worse than the optimal tree is the best tree consistent with the predictions of the hypothesis. In order to do this with PAUP, one needs to define a constraint tree in a tree file, which in this example is called ''constraint.nex.'' It is a simple file that contains only a trees block.

```
# nexus
begin trees;
   utree constraint1 = (1 – 5,(6,7,8)); [Taxa 8 – 22 will be
      unresolved in the constraint tree]
end;
```

There are a few important points to note here. First, the constraint tree will not be fully resolved. Ideally, it should be resolved to the minimum extent possible while still fitting the predictions of the hypothesis being tested. The above constraint tree would be used to test some evolutionary hypothesis that predicts that taxa 6, 7, and 8 exclusively share a common ancestor. Second, not all taxa in the data matrix need to be specified in the constraint tree. So, if there were more than eight sequences in the test dataset, but the hypothesis under examination does not address them, they should be left unresolved in the constraint tree and need not even be included in the constraint tree. To test this hypothesis (that taxa 6, 7, and 8 exclusively share a common ancestor), we first need to find the unconstrained ML tree (as above). Let us assume that we have chosen the HKY+I+Γ model and found that the ML tree has a score of −6456.14360, and that the clade (6, 7, 8) is not present on the ML tree. The ML tree must be saved to a file:

```
savetree file = ML.tre;
```

We now need to run a constrained search to find the best tree which contains clade (6, 7, 8).

```
loadconst file = constraint.nex;
showconst; [make sure the constraint tree is correct]
hs enforce = yes;
lsc nst = 2 trat = est ba = est ra = g sha = est pinv = est;
   [we should reoptimize parameters on this tree to find
   the best possible fit between data and hypothesis]
lsc trat = prev ba = prev sha = prev pinv = prev;
savetree file = hypothesis.tre;
```

The `showconst` command allows one to view the constraint and ensure that the constraint tree was written correctly; it is a good idea to check this before running long searches. In our example, the best tree constrained to contain the clade (6, 7, 8) has a likelihood score of −6481.94451, a deterioration of 25.80091 lnL units. It is critical to save the ML branch lengths if one is interested in employing the parametric bootstrap for significance testing.

The two trees are as follows, with the tree on the left being the ML tree and the tree on the right being the best tree constrained for taxa 6, 7, and 8 constrained to form a group (Fig. 2). Au_C39_5

Note that, because there are no characters supporting that clade (6, 7, 8) in the dataset, the group is united by an internal branch length of zero. This is sometimes the case in constrained trees.

Evaluating the Test Statistic

In this example, the value of the test statistic is, therefore, 25.80. For several years, the only approach available to assessing the significance of the test statistic, and therefore testing the hypothesis that predicts the presence of clade (6, 7, 8), was through the use of the Kishino-Hasegawa test (K-H test) (Kishino and Hasegawa, 1989). Assuming that there are no trees in the display buffer (i.e., that the best constrained tree was saved to the file ‘‘hypothesis.tre’’ and the ML tree was saved to the file ‘‘ml.tre’’) and we have selected the HKY+I+Γ model of evolution, this is accomplished in PAUP as follows:

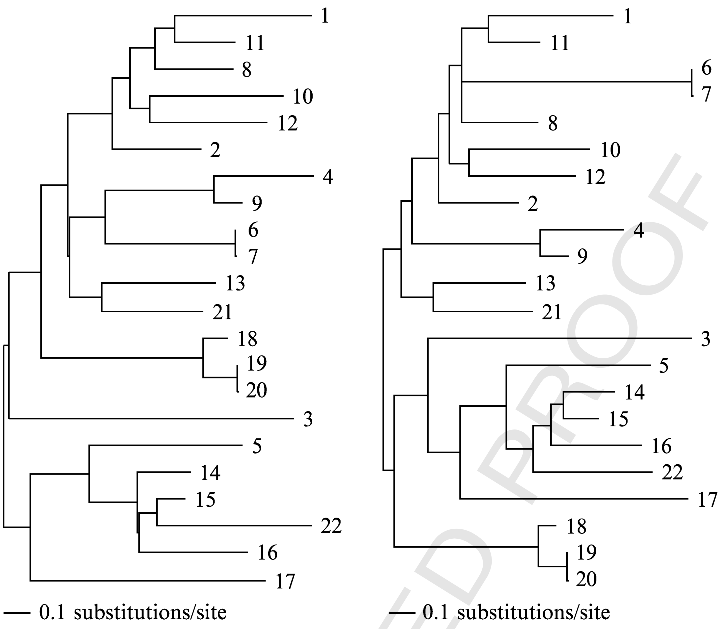


Fig. 2.

```
gettree file = ml.tre;
gettree file = hypothesis.tre mode = 7; [mode = 7 retains any
   trees currently in buffer]
lset nst = 2 ba = est trat = est ra = g sha = est pinv = est;
lscore/khtest = normal;
```

The output is as follows

```
Kishino-Hasegawa test:
  KH test using normal approximation, one-tailed test
                                                  KH-test
Tree             -ln L       Diff -ln L              P
----------------------------------------------------
 1          6456.14360           (best)
 2          6481.94451         25.80091        0.000*
*P < 0.05
```

This has been the most commonly used of the K-H tests, with a normal distribution of single-site lnL differences assumed for the null distribution.

This is a somewhat stringent assumption that can be relaxed by using the reestimated log likelihood (RELL) bootstrap procedure:

```
lscore/khtest = rell nrep = 1000;
```

with the output as follows:

```
Kishino-Hasegawa test:
   KH test using RELL bootstrap, one-tailed test
   Number of bootstrap replicates = 1000
                                                            KH-test
Tree             -ln L        Diff -ln L                        P
-----------------------------------------------------------------
 1        6456.14360             (best)
 2        6481.94451          25.80091              0.015*
* P < 0.05
```

Use of the RELL bootstrap to assess significance of the test statistic is preferable because it eliminates assumptions about the actual shape of the null distribution. If the null hypothesis (the support for the two trees is not significantly different) is false, the distribution of single-site lnL's will be skewed (i.e., not normal). Note that this is a one-tailed test. This seems appropriate because we are using the test incorrectly; one tree is the estimated ML tree, whereas one is known *a priori* to be suboptimal (Goldman *et al*., 2000; Shimodaira and Kishino, 1999). For the test to be used appropriately, the two trees being compared must be selected *a priori* and a two-tailed test would be more appropriate:

```
lscore/khtest = rell nrep = 1000 tailkh = 2;
```

With the output of

```
Kishino-Hasegawa test:
   KH test using RELL bootstrap, two-tailed test
   Number of bootstrap replicates = 1000
                                                            KH-test
Tree            -ln L       Diff -ln L                          P
-----------------------------------------------------------------
 1       6456.14360           (best)
 2       6481.94451         25.80091            0.018**
P < 0.05
```

Shimodaira and Hasegawa (S-H test) have attempted to correct this bias by including a set of trees into consideration and centering the null distribution. However, the collection of trees to consider still must be erected *a priori,* and with just two trees that were selected *a posteriori* considered, the S-H test reverts to the H-K test.

Therefore, the most appropriate way to assess the significance of the test statistic ($\text{lnL}_{\text{MLTree}} - \text{lnL}_{\text{ConstrainedTree}}$) is through use of the parametric bootstrap (Goldman *et al*., 2000; Hillis *et al*., 1996). This entails generating the null distribution by simulation, with the best constrained tree used as the model (true) tree for simulation. The idea is that we have the test statistic measure how much more poorly the data fit the hypothesis than they do the ML tree. We now want to derive a probability (given the hypothesis is true) of observing a test statistic at least as large as that observed in the real data that is simply due to stochasticity. This provides an assessment (conditional on the model; see below) of whether phylogenetic uncertainty can plausibly explain the difference between the ML tree and the relationships predicted by the hypothesis. To accomplish a parametric bootstrap test, one needs a program that can simulate sequence evolution given a tree with branch lengths and a fully specified model of sequence evolution. For the example above, we can use the tree saved to simulate sequences using the program Seq-Gen (Rambault and Grassley, 1997), which is available for several platforms. Because Seq-Gen does not model gaps, missing data, or ambiguities, the test statistic, branch lengths, and model parameters must be recalculated after excluding any characters that contain any such issues. Assuming the tree is loaded into PAUP's tree buffer, one accomplishes this as follows:

```
exclude gapped missambig;
lsc all/nst = 2 ba = est trat = est ra = g sha = est pinv = est;
savetree file = NoGapHypoth.tre brlens = y;
```

For Mac systems, the easiest thing to do is to convert the tree file that we just saved into one that Seq-Gen can read. This is done by deleting everything from the tree file except the tree (the file should be just a single line that specifies the tree in Newic format with branch lengths); here, I will rename it ''NoGapHypothIn.tre.'' In the dialogue box that appears when starting the program, one hits the ''file'' button on the bottom left and then navigates to the appropriate file (NoGapHypothIn.tre). It is best to save the output to a file (by clicking the ''file'' button on the bottom right). In the field labeled ''Argument,'' one specifies the model of sequence evolution (in our case HKY+I+Γ), the sequence length (here, 720 bp), the number of replicate datasets to generate, and any formatting information. This includes type of output file (i.e., PAUP vs. Phylip) and any set of commands that one wants to use in analyzing the simulated sequences. This last task is accomplished by reference to a text file that contains a PAUP block (e.g., PaupBlock.txt). The program allows a number of options, including use of mixed models for

Comp. by:jaykumar Date:4/3/05 Time:21:32:38 Stage:First Proof File Path://fsc/
serials/PRODENV/000000~1/000BD5~1/S00000~1/000000~3/000000~3/
000006192.3D Proof by:N. Pushparaj QC by:Thiru ProjectAcronym:MIE
Volume:395 ChapterNo:0035

multilocus (or otherwise partitioned) data and the ability to use different true trees for different partitions. For the purpose of simulating null distributions to assess the significance of our test statistic, we want to simulate a number of replicate datasets on the best constrained tree (with its ML branch estimates of branch lengths and model parameters). We then must find the difference between the ML tree and the best constrained tree for each replicate. Thus, the argument line for this example would look like this:

```
–mHKY –f0.379386, 0.329111, 0.053277, 0.238226
    –t9.094694 –a0.465274 –i0.461462 –l720 –n100 –on
    –xPaupBlock.txt,
```

where the file PaupBlock.txt contains the following:

```
begin paup;
    set monitor=no autoclose=y;
    dset dist=logdet;
    nj; [get an approximate tree for parameter
      estimation]
    lsc/nst=2 ba=est trat=est ra=g sha=est pinv=est;
    lset ba=prev trat=prev sha=prev pinv=prev;
    set crit=like;
    hs;
    lsc 1/ba=est trat=est sha=est pinv=est
        scorefile=mltree.score append; [reoptimize
          parameters
        on the ML tree to find the best possible fit]
    lset ba=prev trat=prev sha=prev pinv=prev;
    loadconstr file=constraint.tre;
    hs enforce=y; [find the best tree constrained to
        fit the hypothesis]
    lsc 1/ba=est trat=est sha=est pinv=est
        scorefile=hypotree.score append;[reoptimize
        parameters on best constrained tree to find best
        possible fit to hypothesis]
end;
```

Seq-Gen will generate a single file with 100 simulated datasets (i.e., 100 data blocks) under the hypothesis being examined (i.e., using the best tree constrained to fit the hypothesis of interest). After each of the data blocks, the output file will contain the above PAUP block, which will find the best unconstrained tree (and write the ML score to a file named `mltree.score`) and the best tree constrained to fit the hypothesis (and write the ML score the `hypotree.score` file). These two score files can be merged into a single database file (e.g., using Microsoft Excel) and the distribution of differences across the 100 (or more replicates) is the null distribution with which to assess the significance of the test statistic (25.80091 lnL units). If one is using a Unix version of Seq-Gen, the simulation is conducted with a single command:

```
seq-gen -mHKY -f0.379386,0.329111,0.053277,0.238226
  -t9.094694 -a0.465274 -i0.461462 -l722 -n100
  <NoGapHypothIn.tre>OutFile.dat -on -xPaupBlock.txt
```

These runs can take substantial CPU time. One way to reduce this is to confine the tests to the parsimony searches. One then uses the difference in tree length of constrained versus unconstrained trees as the test statistic, still uses ML to estimate branch lengths and model parameters of the best constrained tree as the model tree/parameters, and then searches each replicated for the best constrained and unconstrained MP trees. A preferable alternative is to use emerging technology in parallel clusters (i.e., a Beowulf cluster). For example, at the University of Idaho, the Bioinformatics computing core facility has two clusters. The larger of these, a modest cluster of eighty-eight 2.8-GHz processors can run a 500-replicate full ML parametric bootstrap analysis for a moderate dataset of 66 taxa and 1.5 Kb in just 2 days. Given that it takes much more time than that to generate the real data (and the growing affordability/availability of Beowulf clusters), it makes little sense to take shortcuts in data analysis.

Parametric Bootstrap Test of Absolute Goodness of Fit

One caveat that must be given in the use of parametric bootstraps is their reliance on the chosen model of evolution. In relying on the chosen model to simulate the null distribution, one makes the assumption that the model is adequate (Felsenstein, 2003). In the example given above, despite that we have selected the HKY+I+Γ model objectively based on its fit/performance relative to others examined, we still have no indication about its absolute goodness of fit. Goldman (1993) introduced an absolute goodness-of-fit test that is based on simulation. Here, the null hypothesis is a perfect fit between model and data. As we can see from Eqn. 8, the ML score that a dataset can have occurs when the model predicts the data exactly, that is, when the probability of observing each site pattern is equal to the frequency of each site pattern in the dataset. Thus, the maximum possible likelihood score can be calculated as

$$\ln L_{\max} = \sum_{a=1}^{4^n} f_a(\mathrm{In} f_a) \tag{9}$$

This is the unconstrained likelihood and is sometimes called the *multinomial likelihood*. One can think of it as the likelihood score under a scenario in which each site is allowed its own model and tree. This value is calculated by PAUP every time one invokes the lscore or hs commands. The difference between the ML score under the model and the unconstrained likelihood measures the deterioration in fit associated with forcing all the data to a single model (in this case HKY+I+Γ) and tree. Thus, we have a test statistic:

$$\delta = \ln L_{\max} - \ln L_{\mathrm{HKY+I+\Gamma}}(\mathrm{Data}|\tau) \qquad (10)$$

We can now use Seq-Gen to simulate sequences under the model and find the distribution of the difference between the $\ln L_{\max}$ and the ML score under the model. Here, we know that the fit between model and data is perfect, because the model was used to generate the data, and therefore, any deviation between $\ln L_{\max}$ and the ML score is attributable to stochasticity. For the real data, the ML score under HKY+I+Γ is −6374.32274, whereas the $\ln L_{\max}$ is −2938.93723; the test statistic is $\delta = 3435.38551$. To use Seq-Gen to simulate the expectation of this statistic under the null hypothesis of a perfect fit, we use the ML tree under the model and the ML estimates of model parameters. Furthermore, we embed the following text (using `-xfilename` in the argument field) into after each replicate data block:

```
begin paup;
    log file=ABGoF.log append;
    dset dist=logdet;
    nj;
    lsc/nst=2 ba=est trat=est ra=g sha=est pinv=est;
    lset ba=prev trat=prev sha=prev pinv=prev;
    set crit=like;
    hs;
    lsc 1/ba=est trat=est sha=est pinv=est
        scorefile=ABGoFML.score append;
end;
```

The log file can then be searched to extract lines containing the string "`-lnL (unconstrained) =`" (the "Copy Lines Containing" tool in text editors can be used for this), which will occur three times in the log file for each replicate. The file `ABGoFML.score` will contain the ML scores under the HKY+I+Γ model for each replicate, as well as the distribution of differences forms the null distribution. Here, any deviation between the ML score under HKY+I+Γ and the unconstrained (multinomial) model is due simply to stochasticity, because HKY+I+Γ was used to generate the data (i.e., the data fit it perfectly).

In our example, the observed difference of $\delta = 3435.38551$ is well within the distribution simulated under the null hypothesis of a perfect fit between the model and the data (Fig. 3). Au_C39_6

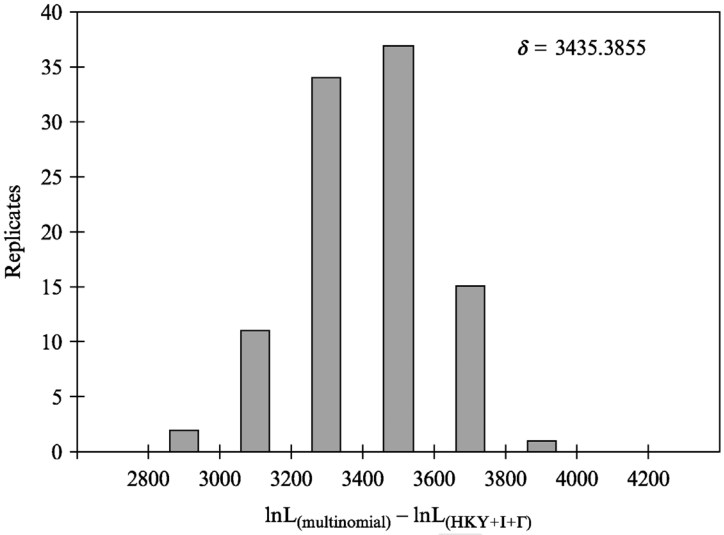


Fig. 3.

The HKY+I+Γ, therefore, seems to be a statistically adequate description of the processes that generated this dataset; the fact that the model assumes independence among sites and ignores codon structure should not lead to biases in application of the model to statistical hypothesis testing.

Concluding Remarks

Advances in model complexity (Yang, 1994; Yang *et al.*, 1994), algorithmic efficiency, and cluster computing have made ML estimation of phylogeny applicable to increasingly large datasets. This is certainly true for incorporation of likelihood into a bayesian framework (see Chapter Au_C39_7 XX). It is also true under the traditional frequentist framework, in which point estimates of parameters of interest are sought (e.g., optimal topologies) in conjunction with an analysis of the uncertainty associated with the point estimate. Given the amount of time and grant money that are invested in generating sequence data, it makes little sense to analyze data in a less than rigorous fashion.

Acknowledgments

This research is part of the University of Idaho Initiative in Bioinformatics and Evolutionary Studies (IBEST). Funding was provided by NSF EPSCoR EPS-0080935 (to IBEST), NSF Systematic Biology Panel DEB-9974124 (to JS), and NIH NCRR grant NIH NCRR 1P20RR016448-01 (to IBEST). The following provided much appreciated guidance, advice, editorial comments and/or suggestions with regard to content: Dave Althoff, Ken Berger, Bryan Carstens, Jeremiah Degenhardt, Sarah Hird, Barley Hyde, Eric Roalson, Kari Segraves, Angie Stevenson, Karina Villa, and Liz Zimmer.

References

DeBry, R. W., and Olmstead, R. G. (2000). A simulation study of reduced tree-search effort in bootstrap resampling analysis. *Syst. Biol.* **49,** 171–179.

Felsenstein, J. (1973). Maximum likelihood and minimum-steps methods for estimating evolutionary trees from discrete characters. *Syst. Zool.* **22,** 240–249.

Felsenstein, J. (1985). Confidence limits on phylogeny: An approach using the bootstrap. *Evolution* **39,** 783–791.

Felsenstein, J. (2003). ''Inferring Phylogenies.'' Sinauer, Sunderland, MA.

Frati, F., Simon, C., Sullivan, J., and Swofford, D. L. (1997). Evolution of the mitochondrial COII gene in Collembola. *J. Mol. Evol.* **44,** 145–158.

Goldman, N. (1993). Statistical tests of models of DNA substitution. *J. Mol. Evol.* **36,** 182–198.

Goldman, N., and Whelan, S. (2000). Statistical tests of gamma-distributed rate heterogeneity in model of sequence evolution in phylogenetics. *Mol. Biol. Evol.* **17,** 975–978.

Goldman, N., Andersen, J. P., and Rodrigo, A. G. (2000). Likelihood-based tests of topologies in phylogenetics. *Syst. Biol.* **49,** 652–670.

Hillis, D. M., Mable, B. K., and Moritz, C. (1996). Applications of molecular phylogenetics: The state of the field and a look to the future. *In* ''Molecular Systematics,'' (D. M. Hillis, C. Moritz, and B. K. Mable, eds.), 2nd Ed., pp. 515–543. Sinauer, Sunderland, MA.

Kishino, H., and Hasegawa, M. (1989). Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order of Hominoidea. *J. Mol. Evol.* **29,** 170–179.

Minin, V., Abdo, Z., Joyce, P., and Sullivan, J. (2003). Performance-based selection of likelihood models for phylogeny estimation. *Syst. Biol.* **52,** 674–683.

Mort, M. E., Soltis, P. S., Soltis, D. E., and Marby, M. L. (2000). A comparison of three methods for estimating internal support on phylogenetic trees. *Syst. Biol.* **49,** 160–170.

Posada, D., and Crandall, K. A. (1998). Modeltest: Testing the model of DNA substitution. *Bioinformatics* **14,** 817–818.

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

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

Sullivan, J., and Swofford, D. L. (1997). Are guinea pigs rodents? The importance of adequate models in molecular phylogenetics. *J. Mamm. Evol.* **4,** 77–86.

Au_C39_8 Sullivan, J., Swofford, D. L. (in preparation). Starting point dependence and the successive approximations approach to maximum likelihood estimation of phylogeny from DNA.
Au_C39_9 *In* ''Statistical Methods in Molecular Evolution'' (R. Nielson, ed.),

Sullivan, J., Holsinger, K. E., and Simon, C. (1996). The effect of topology on estimates of among-site rate variation. *J. Mol. Evol.* **42,** 308–312.

Sullivan, J., Markert, J. A., and Kilpatrick, C. W. (1997). Phylogeography and molecular systematics of the *Peromyscus aztecus* group (Rodentia: Muridae) inferred using parsimony and likelihood. *Syst. Biol.* **46,** 426–440.

Swofford, D. L. (1998). ''PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods),'' Version 4.0b10a. Sinauer Associates, Sunderland, MA.

Swofford, D. L., and Sullivan, J. (2003). Phylogenetic inference using parsimony and maximum likelihood using PAUP*. *In* "The Phylogenetic Handbook" (M. Salemi and A. M. Vandamme, eds.). Cambridge University Press, Cambridge, UK.

Swofford, D. L., Olsen, G. J., Waddell, P. J., and Hillis, D. M. (1996). Phylogenetic inference. *In* "Molecular Systematics," (D. M. Hillis, C. Moritz, and B. K. Mable, eds.), 2nd Ed., pp. 407–514. Sinauer, Sunderland, MA.

Waddell, P. (1995). "Statistical methods of phylogenetic analysis, including Hadamard conjugations, LogDet transforms, and maximum likelihood." Ph.D. Thesis, Massey University, Palmerston North, New Zealande.

Yang, Z. (1993). Maximum likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Mol. Biol. Evol.* **10,** 1396–1401.

Yang, Z. (1994). Estimating the pattern of nucleotide substitution. *J. Mol. Evol.* **39,** 105–111.

Yang, Z., Goldman, N., and Friday, A. (1994). Comparison of models for nucleotide used in maximum-likelihood phylogenetic estimation. *Mol. Biol. Evol.* **11,** 316–324.

[40] Context Dependence and Coevolution among Amino Acid Residues in Proteins

By Zhengyuan O. Wang and David Pollock

Abstract

As complete genomes accumulate and the generation of genomic biodiversity proceeds at an accelerating pace, the need to understand the interaction between sequence evolution and protein structure and function rises in prominence. The pattern and pace of substitutions in proteins can provide important clues to functional importance, functional divergence, and adaptive response. Coevolution between amino acid residues and the context dependence of the evolutionary process are often ignored, however, because of their complexity, but they are critical for the accurate interpretation of reconstructed evolutionary events. Because residues interact with one another, and because the effect of substitutions can depend on the structural and physiological environment in which they occur, an accurate science of evolutionary functional genomics and a complete understanding of selection in proteins require a better understanding of how context dependence affects protein evolution. Here, we present new evidence from vertebrate cytochrome oxidase sequences that pairwise coevolutionary interactions between protein residues are highly dependent on tertiary and secondary structure. We also discuss theoretical predictions that impinge on our expectations of how protein residues may interact over long distances because of their shared need to maintain protein stability.

0076-6879/05 $35.00