A statistical analysis of the Evolutionary Trace method

Heidi Spratt

Contents

Illustrations

Chapter 1

Introduction

Computational tools for classifying sequences, detecting similarities between DNA and protein sequences, predicting molecular structure and function, and reconstructing the evolutionary history of DNA and protein sequences are an important part of the recent developments in the field of bioinformatics. All of these research areas are important to the understanding of life and evolution, as well as to the discovery of new drugs and therapies (Baldi and Brunak 1998).

The expansion of the field of bioinformatics is being fueled by the demand for sophisticated analyses of biological sequences (Durbin, et. al 1998). Part of the challenge associated with the field is to organize, parse, and classify the immense amount of sequence data. Little is known about the complex relationship between a protein sequence, it's structure, and the function of the protein. Thus, Lichtarge (1998) devised a method to examine the relationship between the protein sequence and the important functions of the protein while utilizing the structure of the protein. Protein active sites control nearly all protein functions and determine the interactions upon which biological pathways and cellular networks are built. Characterization of the active sites in a protein would therefore lead to new methods of controlling proteins and ultimately controlling cells. This thesis work hopes to better facilitate the work of Lichtarge and add some statistical analysis to the procedure as well.

Chapter 2 explains proteins, the process of obtaining a multiple sequence alignment, and phylogenetic trees. Chapter 3 describes the various methods used for inferring evolutionary trees and how to build them. Chapter 4 explains the method of the evolutionary trace and why it is useful. Chapter 5 describes the method of the bootstrap and some preliminary analyses of two protein families. Chapter 6 describes the work that is intended to be done for this thesis.

Chapter 2

Evolution of Proteins

2.1 Proteins

Proteins are unbranched chains made by linking together several hundred amino acids by peptide bonds which are strong covalent links (Wood et. al. 1997). Amino acids, the building blocks of proteins, are formed by combining three nucleic acids together. In a living system, a protein is assembled in a long polypeptide chain, one amino acid at a time. Proteins are formed when one or more chains coil up in certain ways to form three-dimensional structures with certain properties. The sequence of amino acids in a polypeptide chain determines the biological character of the protein molecule; even one small variation in the sequence may alter or destroy the way in which the protein functions (Curtis & Barnes 1989).

Proteins play many different roles in living organisms and take on many different forms. The different amino acids in a protein determine the chemical and structural properties of the protein. Some of the functions that proteins perform are binding and carrying specific molecules or ions from one organ to another (known as transport proteins), acting as chemical messengers between cells in different parts of the body in order to modify the activity of the recipient cell (called hormones), providing protection against invading bacteria and foreign viruses (antibodies), and regulating the expression of genes (regulatory proteins) (Blackstock 1998).

The evolution of proteins is caused by mutations that alter the base sequence of a DNA segment. These mutations cause the alteration of one or more amino acids in a protein depending on how severe the mutation is. Harmful mutations are usually eliminated quickly because they are lethal to the carriers. A mutation at an unimportant site simply changes the amino acid at a site or two but leaves the structure of the protein unchanged. Researchers believe that mutations occur more frequently at unimportant sites that at functionally important ones.

2.2 Multiple Sequence Alignment

Sequences are compared to look for evidence of mutation and selection when it is assumed the sequences diverged from some common ancestor (Durbin et. al. 1998). Substitutions, which change residues in a sequence, as well as insertions and deletions, which add or remove residues in a sequence, are the basic mutational processes. Insertions and deletions are commonly referred to as gaps.

The method of Needlman and Wunsch (1970) is a basic dynamic programming algorithm used for the alignment of two biological sequences. The method is based on the smallest unit of comparison between two protein sequences: amino acids. One amino acid from each protein sequence is compared to the other. The maximum match between the two sequences is the largest number of amino acids that, when aligned, match with those of the other sequence. The best match can be determined by creating a matrix of all possible pair combinations that can be constructed from the two sequences.

The aim of pairwise alignments is to align two entire homologous protein regions by using a balance between matches and gaps (Hillis et. al. 1996). Because any two sequences could be perfectly aligned given enough gaps, gaps must be penalized somehow. Gap penalties can thus be a combination of the length of the gap as well as the quantity of gaps. The quantity of gaps should be penalized more than the length of the gaps. This is because there is no reason one should believe a priori that insertion/deletion events are more likely to only involve sequences of short length. If two sequences do not have the same length, penalties may also be assigned for leading and trailing gaps (those at the beginning and end of the sequence respectively). However, these penalties are usually lower than those for internal gaps since leading and trailing gaps usually have more to do with equalizing sequence length than evolutionary changes. All substitutions can be assigned the same penalty, or they can be assigned a matrix of penalties giving different penalties for transitions and transversions or for any changes between two amino acids.

For most phylogenetic studies, multiple sequences, instead of just two sequences, must be aligned. The method of Needleman and Wunsch could be extended to deal with multiple sequences, but such an approach would be computationally burdensome. One approach would be to create pairwise alignments from the sequences then add sequences together to form the multiple sequence alignment (MSA) by inserting additional gaps across the alignment as needed (Hillis et. al. 1996). The main problem with this method is that the MSA created is highly dependent on the order of the species considered. Feng and Doolittle (1987) suggested obtaining the order of the sequences based on an initial tree produced from a matrix of distances collected from all pairwise alignments.

The method of Feng and Doolittle (1987) is a progressive alignment method which utilizes the method of Needleman and Wunsch iteratively in order to obtain a MSA and to construct an evolutionary tree to depict the relationship between protein sequences. The method puts more value in the comparison of recently diverged sequences than in those that are very distantly related (Feng and Doolittle 1987). Similarity scores are calculated via the Needleman-Wunsch algorithm and then converted to difference scores by the relationship

$$D = -\ln \mathrm{S}_{eff} \times 100 = -\ln \frac{\mathrm{S}_{real} - \mathrm{S}_{rand}}{\mathrm{S}_{ident} - \mathrm{S}_{rand}} \times 100 \qquad (2.1)$$

where $\mathrm{S}_{\mathrm{real}}$ is the alignment score, $\mathrm{S}_{\mathrm{rand}}$ is the score obtained with random sequences of the same lengths and compositions, and $\mathrm{S}_{\mathrm{ident}}$ is the average score of the two sequences being compared when each is aligned with itself. The value for $\mathrm{S}_{\mathrm{rand}}$ was obtained from many previous observations of the authors. A tree used to establish preliminary branching orders is then constructed. The smallest difference score is determined and a new matrix is constructed containing the average distances between members of the first pair and the remaining sequences. The procedure is repeated until all scores have been incorporated.

2.3 Protein Model of Evolution

Essential to producing reliable results further along in the analysis is the estimation of amino acid replacement frequencies during molecular evolution. Independence of evolution at different residues is assumed so that sites can be examined one by one. Models of evolution in use today rely on Markov processes. Transition probabilities rely only on the current state of the amino acid and not on past history.

Start with an instantaneous rate matrix **Q**, the rate at which one amino acid is replaced by another. It is important to note that the diagonal elements of **Q** are defined by a mathematical requirement that all the rows add to zero. From this, one can write the transition probabilities from one amino acid to another as a matrix **P**(t). This matrix can be defined by the equation

$$\mathbf{P}(t+dt)=\mathbf{P}(t)(\mathbf{I}+\mathbf{Q}dt) \tag{2.2}$$

where dt is a small amount of time and **I** is the identity matrix. Equation (2.2) can be solved to give (Lio and Goldman 1998)

$$\mathbf{P}(t)=e^{\mathbf{Q}t}. \tag{2.3}$$

This Markov process is homogeneous, stationary, and reversible. This means the rate matrix is independent of time, or that the rates of amino acid substitution remain the same throughout the different levels of the phylogenetic tree. Stationarity gives the matrix the quality that the amino acid transition frequencies have remained virtually unchanged throughout the course of evolution. Reversibility means that the process is viewed the same from the present time to the past as it is from the past to the present time.

There are two widely accepted models in use today: the PAM matrices of Dayhoff et.al (1978) and the BLOSUM matrices of Henikoff and Henikoff (1992). In the PAM matrices amino acid replacement rates are obtained from alignments of protein sequences that are at least 85% identical. The BLOSUM matrices used local, ungapped alignments of distantly related sequences instead. The BLOSUM matrices are also estimated without the reference to a rate matrix **Q**.

2.4 Phylogenetic Trees

The similarity of molecular mechanisms or organisms studied by biologists suggests that all organisms on Earth had a common ancestor (Durbin et. al. 1998). Because of this all species are believed to be related and their relationship is called a **phylogeny**. The relationship is usually represented by a **phylogenetic tree**: a figure depicting which organisms are more closely related to which other organisms. It is a graph composed of branches and nodes (where two species become one). Any given branch connects to exactly two adjacent nodes in a bifurcating tree. The branches represent the relationships among the internal nodes in terms of ancestry and decent. The terminal (external) nodes represent the taxonomic units, or species, present today (Adachi and Hasegawa 1996). The phylogenetic tree is inferred from observations of the organisms existing today (Durbin et. al. 1998).

A true biological phylogeny has an ultimate ancestor of all species. Trees can be either unrooted (a tree describing the relationship between the species but not defining a specific ancestor or a time direction), as depicted in Figure 2.1, or rooted (having a specific common ancestor) as depicted in Figure 2.2.

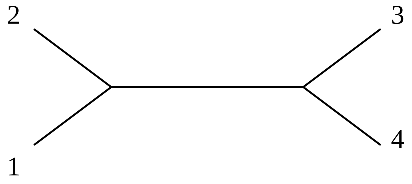

Figure 2.1
An unrooted tree

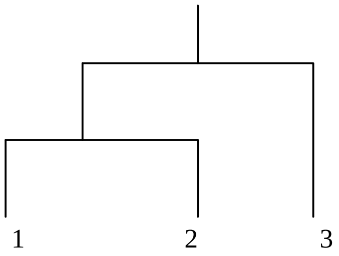

Figure 2.2
A rooted tree

The leaves of the tree represent the species present today. The nodes of the tree represent the time in history when those two species had a common ancestor. Sometimes the leaves of a tree can be swapped without altering the tree (as in switching numbers 4 and 5 in Figure 2.3), but more often than not doing so changes the structure of the tree (as in switching numbers 1 and 2 in Figure 2.3) (Durbin et. al. 1998).

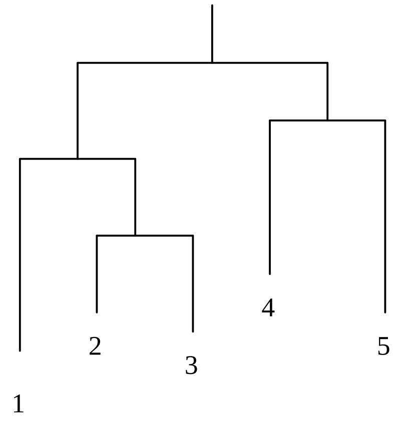

Figure 2.3: A rooted tree

Chapter 3

Tree Building Methodologies

There are several different methods in use today for constructing phylogenetic trees. Some of the methods deal with creating a distance matrix used to construct a phylogenetic tree by joining sequences having the smallest distance between them. Some of the methods are based on maximum parsimony which works by finding the tree that can explain the observed sequences with the minimum number of substitutions (Durbin et. al. 1998). The last of the methods deal with finding the tree topology that maximizes the likelihood of all possible trees.

3.1 Distance Matrix Procedures

For every pair of sequences in the MSA, distance matrix methods calculate an estimate of the branch length separating the sequences. Branch length here is defined as the product of time and rate of evolution (Felsenstein 1996). The evolutionary tree chosen is then the one that makes the best prediction of these pairwise distances based on some criterion. For protein sequences, a probabilistic model of evolution needs to be chosen to base the distances on. Jukes and Cantor (1969) created a simple model where all pairs of amino acids have an equal probability of change. This model is oversimplified since it is known that some amino acids prefer to change to certain other amino acids more frequently than others. Thus Dayhoff, Schwartz, and Orcutt (1978) created a table of probabilities (known as the PAM matrices) based on empirical probabilities of change between amino acids over short periods of evolutionary time. Another set of substitution matrices, known as the BLOSUM matrices, was created by Henikoff and Henikoff (1992). The BLOSUM matrices were created from multiple alignments of more distantly related proteins than those of the PAM matrices.

3.1.1 Unweighted Pair Group Method Using Arithmetic Averages

Some of the methods of tree building begin with a set of distances d_{ij} between each pair i,j of sequences in the data set. There are several different ways of defining distances. One method to do so is employed in the UPGMA (Unweighted Pair Group Method using Arithmetic Averages) (Sokal and Michener 1958) method which works by clustering the sequences of the MSA. The method takes the initial clusters and joins a new sequence at each level, thus creating a new node in the tree. The tree in this case can be envisioned as being built upwards from the nodes to the ancestor. The UPGMA method produces a rooted tree with edge lengths that can be viewed as measured by a molecular clock with a constant rate. This means the divergence of sequences is assumed to occur at some constant rate throughout the tree.

The distance d_{ij} between two clusters C_i and C_j is defined as the average distance between pairs of sequences from each cluster:

$$d_{ij} = \frac{1}{\left|C_i\right| \quad \left|C_j\right|} \sum_{p\, in\, C_i,\, q\, in\, C_j} d_{pq} \tag{3.1}$$

where $\left|C_i\right|$ and $\left|C_j\right|$ are the number of sequences in clusters i and j (Durbin et al 1998).

The method begins with the assignment of each sequence i to its own cluster C_i. Then one leaf of the tree is defined for each sequence and placed at height equal to zero. Two clusters i,j for which d_{ij} is minimal are determined. However, if there are several minimal pairs with the same value one pair should be randomly chosen. Next a new cluster k is defined by $C_k = C_i \cup C_j$ and the distance d_{kl} is defined by

$$d_{kl} = \frac{d_{il}\left|C_i\right| + d_{jl}\left|C_j\right|}{\left|C_i\right| + \left|C_j\right|} \qquad (3.2)$$

for all l. A new node k is defined next with the two daughter nodes being i and j. The new node is placed at a height of $d_{ij}/2$. K is added to the current clusters and i and j are removed. The last five steps are repeated until only two clusters i,j remain and the root is placed at the height $d_{ij}/2$.

3.1.2 The Neighbor-Joining Method

In the case of the tree constructed via the UPGMA method, the tree is assumed to follow a molecular clock (having a constant rate of divergence of sequences) as well as have the additive property. The additive property means that the sum of the lengths of the branches connecting any two nodes on a tree is the distance between any pair of nodes. The neighbor-joining method (Saitou and Nei 1987) was created to deal with trees which follow a molecular clock but are not additive.

The neighbor-joining algorithm works by starting out with a star tree (an unrooted tree where all nodes radiate from the center) and joining neighboring leaves according to some distance criteria. The tree is then constructed by linking the least-distant pair of nodes as defined by a distance matrix where the separation between each pair of nodes is adjusted on the basis of their average divergence from all other nodes (Swofford et. al 1996). When two nodes are linked, their common ancestral node is added to the tree and the terminal nodes are removed from the tree. This process converts the tree from a star-like tree to a true phylogeny. At each stage in the neighbor-joining method, two terminal nodes are thus replaced by one internal node. The process is complete when two nodes remain which are separated by a single branch.

3.2 Maximum Parsimony

Maximum parsimony methods are the most widely used methods in which to infer phylogenetic trees directly from character data. Maximum parsimony does not use a model of sequence evolution. This method simply selects trees that minimize the total tree length, the number of transformations from one character state to another required to explain a given set of sequences (Swofford et. al. 1996). Unlike distance methods, maximum parsimony assigns a cost to a given tree and searches through all possible tree topologies to find the tree with the smallest cost. Parsimony treats each site independently and then adds the number of substitutions necessary for all sites. However, Felsenstein (1978) has shown the method of parsimony to be inconsistent when the evolutionary rate differs among lineages.

3.3 Maximum Likelihood Methods

If one assumes that a phylogenetic tree for a set of sequences is an unknown entity to be estimated then the method of maximum likelihood estimation is well-suited for the data at hand. In this sense, maximum likelihood estimation involves finding the evolutionary tree which has the highest probability of evolving the given data (Felsenstein 1981). More specifically, one seeks the evolutionary tree and its branch lengths that have the highest probability of giving rise to the given amino acid sequences. The probability of such a tree is computed via a stochastic model on residue substitutions in an evolutionary process. Such a model assumes that a residue substitution at a particular residue occurs independently of any substitution at another residue location.

3.3.1 Method of Felsenstein

According the Felsenstein, the problem of computing the maximum likelihood estimate can be reduced to computing the probability of a given set of sequences on an evolutionary tree and maximizing the probability over all trees. Given a model stating the probability sequence S_1 changes to sequence S_2 during some time interval t one can compute the probability of obtaining a given set of sequences at the tips of an evolutionary tree. Independence of site changes needs to be assumed for the method to be computationally feasible. However, independence is not entirely a valid assumption to make because gaps (insertion and/or deletion events) can not be properly modeled if one assumes sites evolve independently of each other. But, by assuming independence, the probability of a given set of sequences arising on a tree can be computed site by site. The product can then be taken across sites to compute the likelihood. For a single site we calculate $P_{ij}(t)$, the probability that a lineage initially in state i will be in state j after t time units. i and j take the values 1, 2,…, 20 corresponding to the twenty amino acids. Assuming two lineages evolve independently after speciation and the same base substitution process occurs in all lineages, calculations of the likelihood are made possible.

If the states at particular sites on nodes of the tree not currently present are known then the likelihood of the given tree would be the product of the probabilities of change in each leaf of the tree times the prior probability of the ancestor of the tree.

The general expression for the likelihood can be written, but it will be more insightful to go through a small example to understand the process better. Figure 3.1 shows the phylogeny tree that will be used in the calculation of the likelihood.

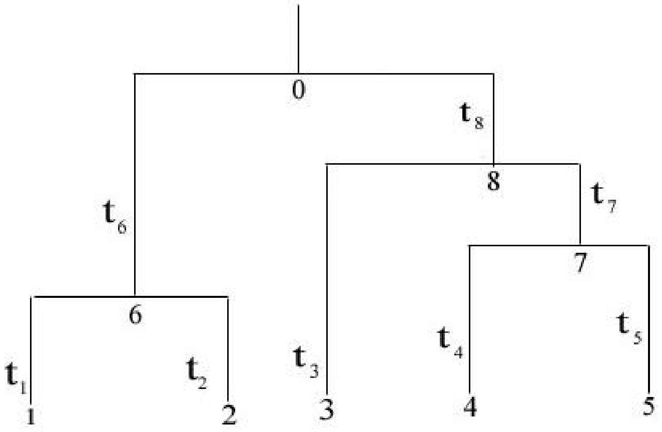

Figure 3.1: Tree used in computation of the likelihood

The lengths of the branches of the tree are denoted by t_i. If the particular amino acids are known at points 0,6,7, and 8 on the tree then the likelihood would simply be the product of the probabilities of change in each segment of the tree times the prior probability of the original base at the origin. The likelihood would then be (Felsenstein 1981)

$$L = p_{s_0} P_{s_0 s_6}(t_6) P_{s_6 s_1}(t_1) P_{s_6 s_2}(t_2) P_{s_0 s_8}(t_8) P_{s_8 s_3}(t_3) P_{s_8 s_7}(t_7) P_{s_7 s_4}(t_4) P_{s_7 s_5}(t_5) \qquad (3.3)$$

where s_i is the amino acid (state) at position i on the tree and π_{s0} is the prior probability of being in state s_0. However, the amino acids at interior nodes of the tree are not generally known, so the likelihood has to be generalized. The first generalization is to sum over all the possible assignments of amino acids to the splits of the trees. The resulting likelihood is

$$L = \sum_{s_0}\sum_{s_6}\sum_{s_7}\sum_{s_8} p_{s_0} P_{s_0s_6}(t_6)P_{s_6s_1}(t_1)P_{s_6s_2}(t_2)P_{s_0s_8}(t_8)\times \\ P_{s_8s_3}(t_3)P_{s_8s_7}(t_7)P_{s_7s_4}(t_4)P_{s_7s_5}(t_5). \tag{3.4}$$

In general this expression will have 2^{2n-2} terms for n species. Thus, the likelihood can be rewritten in a simpler manner. Namely,

$$L = \sum_{s_0} p_{s_0}\{\sum_{s_6} P_{s_0s_6}(t_6)[P_{s_6s_1}(t_1)][P_{s_6s_2}(t_2)]\}\{\sum_{s_8} P_{s_0s_8}(t_8)[P_{s_8s_3}(t_3)]\times \\ [\sum_{s_7} P_{s_8s_7}(t_7)(P_{s_7s_4}(t_4))(P_{s_7s_5}(t_5))]\}. \tag{3.5}$$

Since the full likelihood has been written out, it is now possible to restate everything in terms of conditional likelihoods, thus making computation feasible. Define $L_s^{(k)}$ as the likelihood based on the data at or above point k on the tree, given that point k is known to have state s for the specific site under consideration. Then the above equation (3.5) is equivalent to working down the tree from the tips to the root. For a given point k, whose immediate descendants are i and j, the conditional likelihood can be computed for all twenty values of s_k via

$$L_{s_k}^{(k)} = (\sum_{s_i} P_{s_ks_i}(t_i)L_{s_i}^{(i)})(\sum_{s_j} P_{s_ks_j}(t_j)L_{s_j}^{(j)}). \tag{3.6}$$

If point k is a tip of the tree then $L_s^{(k)}$ will be zero for all values of s except that state actually observed, for which the conditional likelihood will be one. All the terms in eq. (3.5) will have been computed once the bottom fork on the tree is reached. Thus, the overall likelihood of the tree for a specific site can be written as

$$L = \sum_{s_0} p_{s_0} L_{s_0}^{(0)}. \tag{3.7}$$

This is the theoretical approach to calculating the likelihood. However, this approach is computationally infeasible causing Felsenstein to create the “Pulley Principle” to deal with the computational issues.

3.3.1.1 Calculation of Substitution Probabilities

The quantities $P_{ij}(t)$ are the probabilities of transition from one amino acid to another over a segment of length t. These probabilities are assumed to follow a Markov process, one in which the probability of an amino acid changing may depend on the current amino acid but not on past history. Assume that in a small interval of time of length t there is some probability, u *dt*, that the current amino acid at a particular site is changed. u is the rate of amino acid substitution per unit of time. Since this model assumes that transitions and transversions are the same, for infinitesimal *dt* the substitution probabilities are

$$P_{ij}(dt) = (1 - u\,dt)d_{ij} + u\,dt\,p_{j}, \qquad (3.8)$$

where π_j is the probability of amino acid i being replaced by amino acid j, and δ_{ij} is the Kronecker delta function. From eq. (3.8) it can be shown that for an arbitrary t,

$$P_{ij}(t) = e^{-ut}d_{ij} + (1 - e^{-ut})p_{j}. \qquad (3.9)$$

Here e^{-ut} is the probability that a site remains constant over a length of time t. If that site does change over time then the probability that it ends up in state j is simply π_j. One nice property of this Markov process is that it is reversible. This means that the process of amino acid substitution will look exactly the same whether we start at the root of the tree and work toward the nodes or whether we start at the nodes of the tree and work our way toward the root. Reversibility also requires that

$$p_{i}P_{ij}(t) = P_{ji}(t)p_{j}. \qquad (3.10)$$

3.3.1.2 The Pulley Principle

Since the Markov process is reversible and there are no constraints placed on the lengths of the branches in the tree a useful property to the estimation of the phylogenetic trees can be established. This property is called the "Pulley Principle" by Felsenstein (1981). It can be easily shown that the likelihood of the entire tree is unaffected if the placement of the root is altered. For instance, consider the last two steps in the calculation of the likelihood for the above example of Figure 3.1. The likelihood of the tree at one site involves forks 0, 6, and 8 of the tree and is

$$L = \sum_{s_0} p_{s_0} (P_{s_0 s_6}(t_6) L_{s_6}^{(6)})(P_{s_0 s_8}(t_8) L_8^{(8)}). \tag{3.11}$$

The reversibility property (eq. (3.10)) can be invoked to show that

$$p_{s_0} P_{s_0 s_6}(t_6) = p_{s_0} P_{s_6 s_0}(t_6), \tag{3.12}$$

which means that the likelihood can be written as

$$L = p_{s_6} L_{s_6}^{(6)} L_{s_8}^{(8)} \sum_{s_0} P_{s_6 s_0}(t_6) P_{s_0 s_8}(t_8). \tag{3.13}$$

The Chapman-Kolmogorov equation states that the summation on the right of eq. (3.13) can be replaced by $P_{s_6 s_8}(t_6 + t_8)$. This means that the likelihood of the tree depends only on the sum of the branch lengths $t_6 + t_8$ so that the root of the tree may be placed anywhere on the segment joining node 6 to node 8. In other words, the root of the tree acts as a sort of pulley; all the parts of the tree to one side of the root can be shifted up while those on the other side of the root can be shifted down by the same amount. This process can be repeated to show that the likelihood is independent of the placement of the root on the tree.

3.3.1.3 Finding the Maximum Likelihood Tree

The Pulley Principle allows one to alter the length of any given branch of the tree in an optimal fashion. A specific case of the general EM algorithm of Dempster et al. (1977) is used to arrive at an iterative method for finding the optimal tree branch lengths. I will use an example to show how the iterative method can be obtained. Consider the right half of the tree in Figure 3.1. The root of this partial tree can be thought to lie between nodes 7 and 8. Suppose that the root lies immediately to the right of node 8. The likelihood for one particular site in the protein sequence is then given by

$$L = \sum_{s_0}\sum_{s_8}\sum_{s_7} p_{s_0}(P_{s_0 s_8}(0) L_{s_8}{}^{(8)})(P_{s_0 s_7}(t_7) L_{s_7}{}^{(7)}). \tag{3.14}$$

This then becomes

$$L = e^{-t_7}\sum_{s} p_s L_s{}^{(8)} L_s{}^{(7)} + (1 - e^{-t_7})[\sum_{s_8} p_{s_8} L_{s_8}{}^{(8)}][\sum_{s_7} p_{s_7} L_{s_7}{}^{(7)}] \tag{3.15}$$

upon substitution of eq. (3.14) into eq. (3.9) with u = 1. Eq. (3.15) is simply the likelihood corresponding to one site of the protein sequence; the full likelihood is then of the form

$$L = \prod_i (A_i q + B_i p) \tag{3.16}$$

where $q = e^{-t_7}$, $p = 1 - e^{-t_7}$, $A_i = \sum_s p_s L_s{}^{(8)} L_s{}^{(7)}$, and $B_i = (\sum_{s_8} p_{s_8} L_{s_8}{}^{(8)})(\sum_{s_7} p_{s_7} L_{s_7}{}^{(7)})$ for the ith protein site. The logarithm of eq. (3.16) can be taken to obtain an iterative equation for finding the optimal branch lengths; this equation is

$$p^{(k+1)} = \frac{1}{K}\sum_i \frac{B_i p^{(k)}}{A_i q^{(k)} + B_i p^{(k)}}, \tag{3.17}$$

where K is the total number of sites in the protein sequence of interest, and $q^{(k)} = 1\text{-}p^{(k)}$. Iteration is performed for each site until eq. (3.17) converges before moving on to the next site. The iteration is finished when a complete pass through the tree can be made without any of the p_i changing substantially.

3.3.1.4 Searching Among Tree Topologies

In an ideal situation, one would be able to find the maximum likelihood tree by finding the tree that maximizes the likelihood for all possible tree topologies. However, the number of possible unrooted tree topologies grows very rapidly via the expression $\frac{(2n-5)!}{(n-3)!2^{n-3}}$ for n species (Edwards and Cavalli-Sforza 1964) . Thus, a method must be devised to arrive at the maximum likelihood tree that does not take into account every single tree topology. One possible method is to build the tree by successively adding species to a tree starting with just two species. Each time a new species is added to the tree, there will be 2k-5 segments from which the kth species could arise. Each position is tried; the final position is determined by the iteration method described in section 3.3.1.3. The position of the species on the tree topology is determined by that topology which yields the highest likelihood. If more than four species are present in the tree, a local rearrangement of the branches is performed in order to see if the rearrangement improves the likelihood. If any of the rearrangements does improve the likelihood, then the new order is accepted and the next species is added in the manner just described. This process is iterated until all the species have been added and local rearrangement does not provide a higher likelihood. An important thing to note about this method is that it depends on the order of the original sequences in the alignment. The order of the sequences is the same order that the new species are added to find the maximum likelihood tree. Because of this, Felsenstein (1981) recommends trying several runs with different orderings of the input sequences. Although this is not an exhaustive method (i.e. not all tree topologies are utilized), Felsenstein (1981) has found the method to be successful in practice.

3.3.2 Method of Kishino, Miyata, and Hasegawa

In addition to the likelihood method of Felsenstein (1981), Kishino, Miyata, and Hasegawa (1990) developed a method to maximize the likelihood of a given tree topology. This is a method based on a Markov model that does not assume a constant evolutionary rate among amino acids and takes into account unequal transition probabilities among pairs of amino acids (Kishino et. al. 1990). The model also tries to incorporate insertion/deletion events during evolution. Since unequal transition probabilities are assumed, the empirical transition matrix of Dayhoff (Dayhoff et. al. 1978) is used. Dayhoff showed that an amino acid in a protein is replaced more often than expected under equal transition probabilities by a physicochemically similar amino acid.

If independence of evolution at different sites is assumed, the probability of a given set of data can be computed site by site. The product of probabilities can then be taken across all sites at the final stage of computation (Felsenstein 1981). The likelihood for a given tree topology T and sequence data $\mathbf{X}$ is written as $L=\text{Prob}(\mathbf{X}|T, \theta)$ where θ is a vector of parameters. A general expression for the likelihood of a tree is possible to arrive at, but it is more illustrative to present the case of only four taxonomic units. The formulation of the general result will be clear from the simpler example.

Rarely is an individual interested in the evolutionary history of only four species. Hence, a simplified method for dealing with more than five species is presented since the method of maximum likelihood presented here will explode computationally for any more than five species. Start by breaking the total number of species into five groups. The determination of these five groups is important. Hopefully, the groups will be apparent at the beginning and the members will be known to cluster together in advance. Label the five groups s_1, s_2, s_3, s_4, s_5, where $s = s_1+s_2+s_3+s_4+s_5$ is the total number of species one wishes to know the evolutionary history of. In other words, the sequence data can be represented as follows:

$$\text{Group 1}\qquad\left\{\begin{array}{lllll} \text{Species } (1.1) & X_{(1.1)1} & X_{(1.1)2} & \ldots & X_{(1.1)n} \\ \quad\vdots & & & & \\ \text{Species } (1.s_1) & X_{(1.s1)1} & X_{(1.s1)2} & \ldots & X_{(1.s1)n} \end{array}\right.$$

$$\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots$$

$$\text{Group 5}\qquad\left\{\begin{array}{lllll} \text{Species } (5.1) & X_{(5.1)1} & X_{(5.1)2} & \ldots & X_{(5.1)n} \\ \quad\vdots & & & & \\ \text{Species } (5.s_5) & X_{(5.s5)1} & X_{(5.s5)2} & \ldots & X_{(5.s5)n} \end{array}\right.$$

By picking one species from each group, the maximum likelihood procedure of Felsenstein (1981) can be applied to the data for m alternative tree topologies of the five groups. The probability of occupying amino acids x_1, x_2, x_3, x_4, and x_5 at a site in the selection from species s_1, s_2, s_3, s_4, and s_5 is given by

$$f(x_1, x_2, x_3, x_4, x_5 \,|\, q) = \sum_{i=1}^{20}\left[p_i P_{ix1}(t_1) P_{ix2}(t_2) \times \sum_{j=1}^{20} P_{ij}(t_6) P_{jx3}(t_3) P_{jx4}(t_4) P_{jx5}(t_5)\right]. \tag{3.18}$$

The log-likelihood is

$$l(q \mid X) = \sum_{h=1}^{n} \log f(X_h \,|\, q). \tag{3.19}$$

where n is the total number of sites in the sequence of interest. A maximum likelihood estimate of θ can be obtained via Newton's method and following the procedure described.

An alternate strategy to that mentioned above in section 3.3.1.4 for finding the maximum likelihood tree is called "star decomposition". This method is employed in the MOLPHY program developed by Adachi and Hasegawa (1996). Star decomposition is very similar to the neighbor-joining algorithm for a distance matrix (described in section 3.1.2). Just as the neighbor-joining algorithm, this procedure starts with a star-like tree (Adachi and Hasegawa 1996). From the star-like tree, a pair of species is separated from the others. Among all possible pairwise combinations of species, a pairing giving the highest likelihood is chosen. This process is continued until all the multifurcations can be resolved into bifurcations. If not, the Akaike Information Criterion (AIC) (Akaike 1974) is used to determine if the multifurcation should be resolved. The final tree obtained by the method of star decomposition is uniquely defined. Once an approximate tree topology is obtained by the star decomposition method, local rearragements of the sequences are performed to see if a better topology can be found. This is done by moving all the possible internal nodes around to see if a tree topology with a higher likelihood can be found. If such a tree does exist, then the rearragement of the sequences is carried out (Adachi and Hasegawa 1996). The local rearragement procedure is carried out through the entire tree until no further improvement of the likelihood is obtained. It is not guaranteed that this procedure will obtain the maximum likelihood tree. Therefore, the use of several alternative input tree topologies is recommended. The chosen tree should then be the tree with the highest likelihood from all the initial possibilities.

Chapter 4

Evolutionary Trace

4.1 Evolutionary Trace Method

Researchers believe that important residues mutate less than unimportant residues; part of the basis for the theory behind the evolutionary trace method is based on the previous observation. The evolutionary trace (ET) (Lichtarge 1996) is a method to identify active sites in a protein sequence by looking for conserved residues in the branches of an evolutionary tree. The purpose of the method is to be able to rank the functional importance of amino acids in the protein structure as well as to imply some information about the protein sequence-structure-function relationship. This will hopefully lead to better drug design for pharmacologically important proteins.

Sequence fragments used for analysis are gathered from the Basic Local Alignment Search Tool (BLAST) (Altschul et al. 1990) which match the query domain of interest. BLAST is a software package which finds high scoring local alignments between a query sequence of interest and some target database. These homologous protein sequences are aligned (via a GCG program PILEUP) to create a multiple sequence alignment. PILEUP (Higgins and Sharp 1989) follows the progressive alignment method of Feng and Doolittle (1987) described earlier. Thus, a tree constructed by similarity methods from the initial sequence data is used for constructing the proper MSA. Changing the input order of the original sequences into PILEUP can change the final MSA obtained. Currently, there is no method for varying the sequence order to obtain a consistent MSA. The structure of the protein is known for one of the sequences in the MSA.

From the alignment, pairwise similarity scores are computed (also via PILEUP) between each pair of sequences. From these similarity scores an evolutionary tree is created via the method of UPGMA. Partitions of the tree (clusters of branches) are examined to assess functional importance of sequence residues. This is indicated in the tree column of Figure 4.1.

In order to analyze the data, the tree is vertically cut into a certain number of branches. For example, the tree in Figure 4.1 is cut into four branches. For each cut made in the tree, an evolutionary trace can be calculated. In other words, the evolutionary trace can be calculated for a tree when there are five branches present, when there are six branches present, or for however many branches the researcher wishes to examine. The ideal number of branches to cut the tree into depends on the protein of interest. The exact cut-off value is determined by visualization at present and better methods are being discussed. For each branch of the tree (at a particular cut point) a consensus sequence is compiled (see the group column of Figure 4.1). To do this simply look at all the sequences in a given branch of the tree. If all the residues at a given site are the same (conserved), the consensus sequence is given the conserved residue type. When sequences are conserved within each branch, but different across branches at a given site, that site is called class specific. If all the sequences do not have the same residue at each site, then the consensus sequence is neutral and left blank at that site. This is made clear in the upper portion of the Consensus Sequence column of Figure 4.1.

Once a consensus sequence has been obtained for each branch of the cut tree, the consensus sequences are aligned to get an evolutionary trace. If all consensus sequences have the same residue at any given site, that site has a conserved residue in the trace. If the residue varies between consensus sequences, but is never a gap, then that site has a class-specific residue in the trace, as depicted in the last line of the Consensus Sequence column of Figure 4.1. If a residue in any of the consensus sequences is neutral then the trace is also neutral at that site. In other words, the trace completely ignores residues containing a gap somewhere in the alignment. The protein residues are then mapped onto the protein structure (via RASMOL (Sayle and Milner-White 1995)) and color-coded to reflect conservation, class specificity, and neutrality, as depicted in the trace column of Figure 4.1.

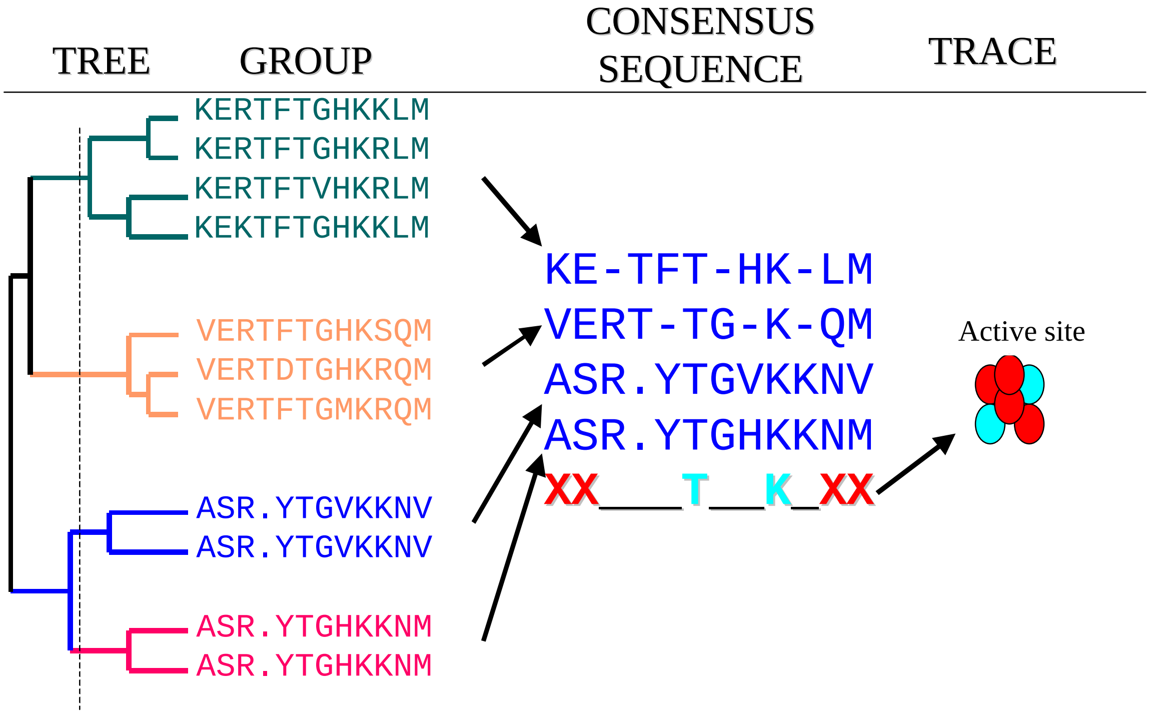

Figure 4.1: The Evolutionary Trace Method. The "Tree" column describes the tree at its branches, the "Group" column describes the sequences at each branch of the tree, the "Consensus Sequence" column shows how the trace is formed from the individual sequences, and the "Trace" column depicts which sties on the protein structure are deemed important by the trace. An "X" in the last line of the "Consensus Sequence" column means that site is not conserved, a "_" means there is a gap somewhere in the consensus sequences, and a specific letter means that site is conserved at the given cut-off value.

4.2 Trace Integral

The rank of each site is the output of the evolutionary trace procedure. The rank is simply the number of branches the tree is vertically cut into in order for a particular site to become class specific. If a site has a rank of 1 then that site is completely conserved. A site that has a rank of 0 means that site has a gap somewhere in the alignment (referred to in the above section as being neutral). A site which has a rank of anything else is one in which there is some variation but at some point along the tree the site becomes class specific.

One measure of how good the evolutionary trace performs is the trace integral, which can be obtained from the rank data. However, the data needs to be slightly transformed before an integral can be taken. The way to do this is by looking at the profile score instead of the rank. The profile can be thought of as the reverse of the rank; it is how many branches need to be subtracted from the terminal nodes of the tree to obtain class specificity at a given site. An example of a profile graph for the analysis done in Section 5.2 is Figure 4.2. To calculate the profile score subtract the rank from one more than the total number of sequences present in the alignment. However, if the rank is zero, the profile score is also zero. The integral is then the area under the profile graph. Thus, the higher the trace integral the better the evolutionary trace method performed on that particular protein. The integral value for Figure 4.2 is 4768.

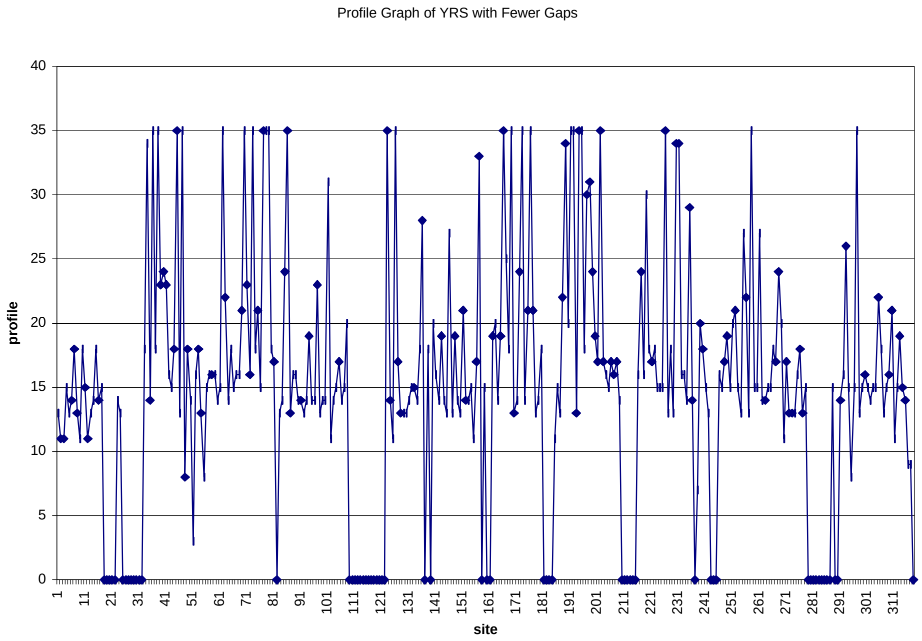

Figure 4.2 The profile graph of the YRS protein with fewer gaps

The profile graph is just one measure of how the trace performs. It is also used to help determine where the tree should be cut off to determine which sites are important in the protein and which do not perform any given function. The peaks in the profile graph indicate important sites, those that become class specific early in the phylogeny. For this particular protein, the cut-off value was determined to be 19 branches.

Chapter 5

Preliminary Work

5.1 Bootstrap

Resampling technology, like bootstrapping, can be used to determine if the clusters of an evolutionary tree are likely to be due to random coincidence. The bootstrap is a computer-based method frequently used to assess the accuracy of many statistical estimates developed by Efron (1979). The basic idea of the bootstrap involves making assumptions about the variability of an unknown distribution from which the data were drawn by resampling from the data. The justification of the resampling is that, if the original sample size n is large, each possible value of x will be represented in the same proportion as in the underlying distribution, and resampling from the data points with replacement will be the same as sampling from the underlying distribution (Felsenstein 1985). There exist both parametric and non-parametric bootstrap methods.

5.1.1 Non-parametric Bootstrap

Felsenstein (1985) describes the method of the bootstrap as follows. Start with data points x_1,x_2, …, x_n which are assumed to be drawn independently from the same distribution. From these values, one can apply some method of statistical estimation to obtain an estimate of the parameter of interest. If the exact distribution of the data was known and if our function were tractable, an estimate of the standard error could be obtained as well as confidence intervals for the estimate. However, when the distribution of the data is unknown (as is frequently the case), the bootstrap is a useful alternative. The bootstrap procedure suggests that the data be resampled to construct several fake data sets. Each of the fake (bootstrapped) data sets is constructed by resampling n points from the original data with replacement. For each fake data set, compute the estimate of interest. The resampling procedure is performed many times, usually on the order of 1000. The idea of the bootstrap is that the 1000 estimates approximate the distribution of the actual estimate of interest. The variance of the estimate of interest can be estimated from the bootstrapped estimates, as well as confidence intervals for the parameter.

For phylogenetic trees, the bootstrap can be used to resample the MSA creating new alignments. Confidence in each branch of the original evolutionary tree can be assessed from these bootstrapped alignments. The confidence estimate is based on the proportion of bootstrapped trees exhibiting the same clustering as the original tree (or partial clustering at a particular partition).

A non-parametric bootstrap sample is created from the original alignment by sampling the columns of the alignment with replacement. Thus, the new alignment may have several repeated columns and several missing columns from the original alignment. It is important to note that the sequence length of the bootstrapped alignment and the MSA are exactly the same.

5.1.2 Parametric Bootstrap

Unlike the non-parametric bootstrap, the parametric bootstrap creates new samples (replicates) by simulation involving maximum likelihood estimates. First, a model of evolution is assumed. Then the parameters of the model are estimated from the data. The parameters of phylogenetic data are the tree topology, the branch lengths of the tree, and transition/transversion rates. The substitution matrices incorporate the transition/transversion rates. New character matrices (multiple sequence alignments) are then simulated assuming the parameterized model from above. Finally, the new replicates are analyzed (Huelsenbeck and Hillis 1996).

The main advantage of the parametric bootstrap is that it generates independent stochastic replicates while the non-parametric bootstrap preserves bias present in the MSA. The main advantage of the non-parametric bootstrap is that it does not base its replicates on any assumptions about the distribution of the data. The parametric bootstrap depends on a detailed model of protein evolution (mainly substitution matrices) and more computational resources associated with obtaining the details.

5.1.3 Confidence Intervals on Phylogenies

Each bootstrap sample involves a tree topology as well as branch lengths. Confidence limits on a statistic are usually based on the percentile method: for a 95% confidence interval take the upper and lower 2.5% points of the distribution of the bootstrapped statistics (Felsenstein 1985). In the case of phylogenies, researchers are often interested in the topology of portions of the tree rather than the whole tree. This is precisely what we wish to assess in the case of the ET, not the total phylogeny but the similarity of the clusters present at a pre-determined cut point of the tree. For instance, if the cut-off value for a particular tree is 19, we wish to examine the different clusters of species present when there are 19 trees in all the bootstrapped samples. Bootstrapping provides us with a confidence interval containing not the true phylogeny, but the phylogeny that would be estimated from repeated sampling of several amino acids from the original data set (Felsenstein 1985).

5.2 Analysis of YRS Protein

The original set of sequences for the tyrosyl tRNA synthetase (YRS) protein, an essential enzyme for protein synthesis, was 699 residues long and consisted of 35 species. The first step in the analysis is to bootstrap the MSA. This is accomplished by using the computer package PHYLIP (Felsenstein 1993) created by Joe Felsenstein and widely distributed on the web. This package contains a program to bootstrap MSA's called SEQBOOT. Seqboot requires an original multiple sequence alignment as well as the length of the sequence and number of species. The result is a bootstrapped MSA. Several bootstrapped samples can be created at once so it is an easy task to create 1000 bootstrapped alignments. The evolutionary trace method is then run on the 1000 bootstrapped samples. The ET method requires the input of a msf file as well as a pileout file. The msf file is a format of the MSA. The pileout file is simply the pairwise similarity scores of the sequences in the alignment. In order to perform the analysis the original msf file is sent to the ET program along with the pileout file created from the bootstrapped sample. The original msf is included because we are just trying to create new trees at this point (which are created via the pileout file) and we do not want to destroy the sequence information present in the original alignment. The trace integral scores are calculated for each bootstrapped sample and are compared to the original integral. The histogram of these values is seen in Figure 5.1.

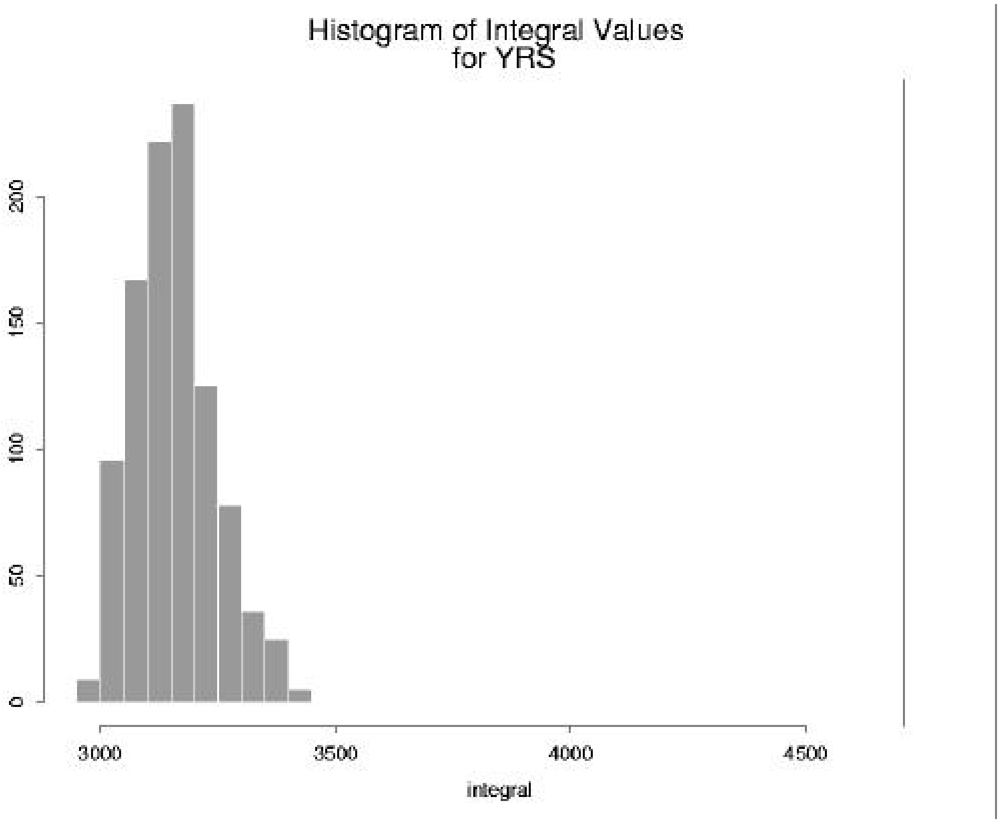

Figure 5.1: Histogram of integral values for the bootstrapped samples of the YRS protein. The vertical line is the integral value (4701) of the original MSA.

As is clearly seen, the value of the original integral is a great deal higher than for any of the bootstrapped samples. It was conjectured that since the ET method ignores those sites with a gap, the original integral would be much higher than the bootstrapped. This is because the bootstrapped alignment might favor a site with a gap and resample it more frequently than another. This could alter the pairwise similarity scores significantly and thusly create a distorted tree. Thus, the decision was made to look at a new MSA of the YRS protein: one with fewer gaps. All non-important gaps were removed from the original alignment. An unimportant gap is one that is not fundamental in the structure of the protein. New bootstrapped samples were created using this shorter alignment which contained only 318 residues. The ET program was then re-run using these shorter bootstrapped alignments, and new integrals were calculated. The resulting histogram of integral values is Figure 5.2. This time the value for the original integral actually falls on the histogram, not far off to the right of it. In this case, there are only twelve bootstrapped samples that have an integral equal to or better than the original trace.

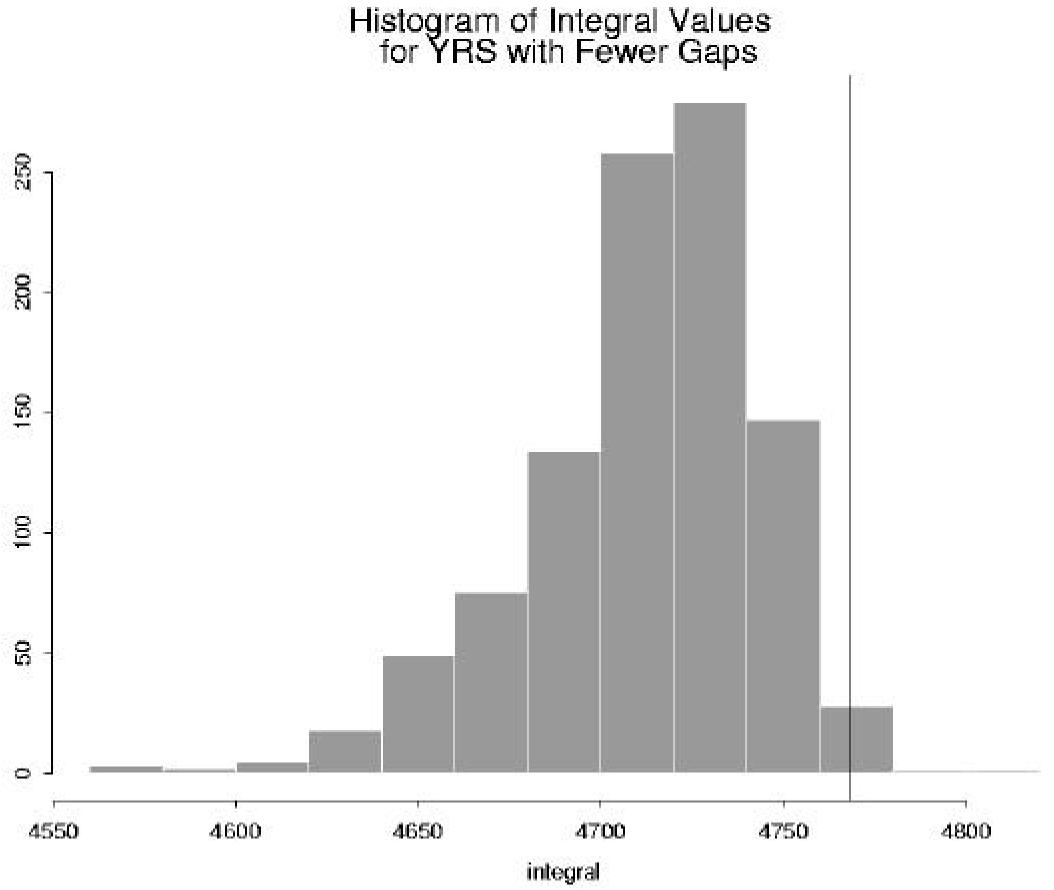

Figure 5.2: Histogram of integral values for the bootstrapped samples of the YRS protein with fewer gaps. The vertical line is the integral value (4768) of the original MSA with fewer gaps.

The next step is to look at each of the bootstrapped samples with a better integral value to see if the important residues are altered very much. It was deemed that for the YRS protein a cut-off value of 19 was sufficient. This means that when the tree has 19 branches one can determine the functionally important residues by finding those residues which are conserved. By looking at a linear correlation of each of the better bootstrap samples to the original it can be seen that all trees are very highly correlated (see the first column of Table 5.1). This holds true for the bootstrapped samples with the worst trees (lowest integral scores), bootstrapped trees in the middle of the integral distribution, and randomly selected bootstrapped trees. All correlation values were above 0.90 which indicates that all MSA's are very highly correlated with each other as well as with the original MSA. The same holds true for Spearman's rank based correlation coefficient as well. When looking at just those sites which are deemed to be important in the original MSA at a cut-off of 19, the correlation drops off slightly (see the second column of Table 5.1). However, there is no significant difference between the better group and the worse group. In both instances the correlation is never worse than 0.84 for all bootstrapped samples.

	Linear Correlation	Linear Correlation with rank of 19 or lower
305	0.9813560	0.9450938
494	0.9840113	0.9487326
559	0.9822103	0.9499063
772	0.9866349	0.9646932
279	0.9875581	0.9563955
491	0.9801784	0.9387111
712	0.9860109	0.9612516
798	0.9862114	0.9500121
528	0.9871333	0.9512679
987	0.9845549	0.9559030
694	0.9885343	0.9673210
939	0.9850674	0.9470137

Table 5.1: Linear correlation of bootstrapped samples to the original.
The rows represent the correlation of different bootstrapped samples to the original.

The parametric bootstrap was also utilized to examine the trace integral in the case of the YRS protein. A program called Pseq-Gen developed by Grassly, Adachi, and Rambaut (1997) implements the parametric bootstrap procedure. This program takes as input a phylogenetic tree with branch lengths and outputs a multiple sequence alignment created from the input tree. Ideally, the input tree would be created via the maximum likelihood method. For the initial analysis, the input tree was created via the method of UPGMA using the NEIGHBOR program from PHYLIP. Since the ET ignores those positions in the MSA where a gap is present, I used the YRS MSA with fewer gaps for the parametric bootstrap analysis. Pseq-Gen created 1000 sequences of length 318 using the UPGMA tree created from PHYLIP and the PAM001 substitution matrix. The ET analysis was run on the parametric bootstrapped samples using the pairwise similarity scores calculated from the parametric bootstrapped MSA and the original MSA. Integral values were then calculated for all 1000 bootstrapped samples. Figure 5.3 depicts the histogram of integral values.   As can clearly be seen from Figure 5.3, the value of the original integral is much larger than that of the parametric bootstrapped samples.

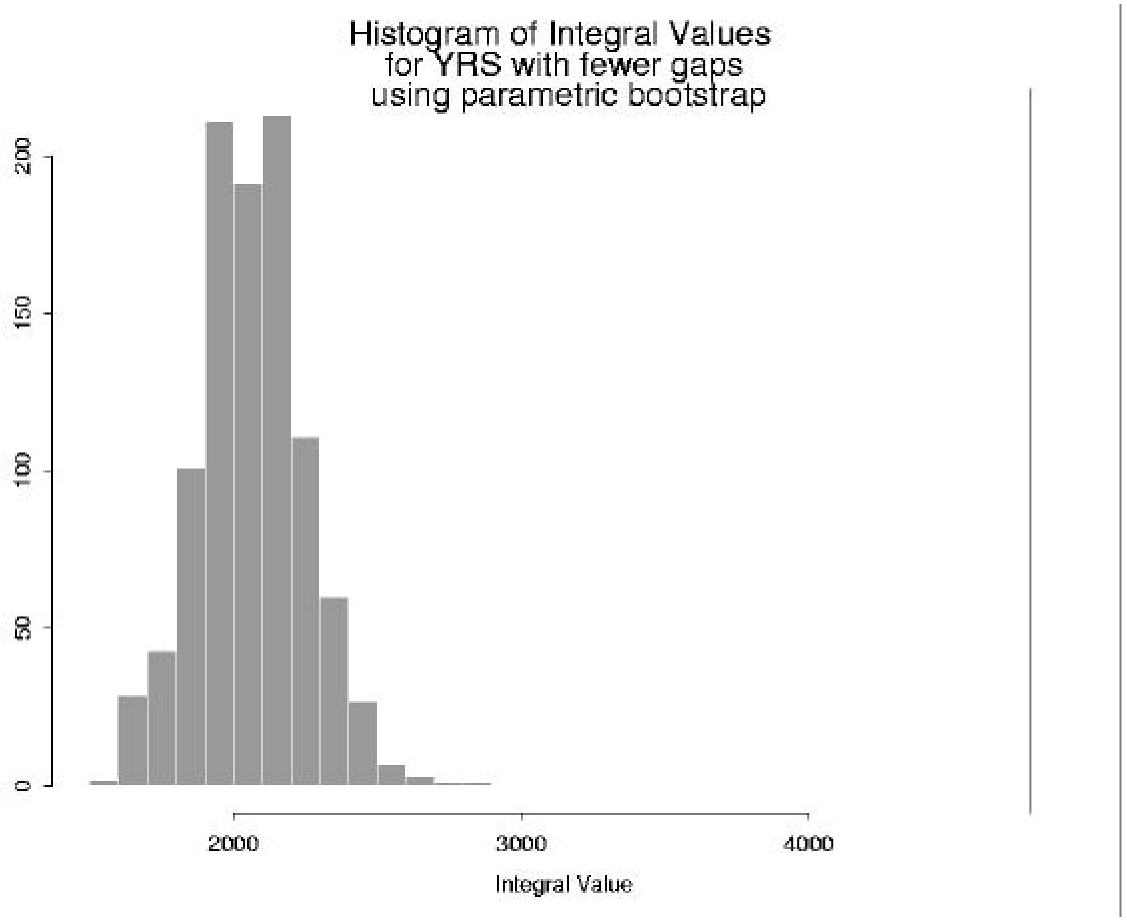

Figure 5.3: Histogram of integral values for the parametric bootstrapped samples of the YRS protein with fewer gaps. The vertical line is the integral value (4768) of the original MSA with fewer gaps.

5.3 Analysis of SRP Protein

The original set of sequences for the signal recognition particle receptor (SRP) protein was 707 residues long and consisted of 36 species. As above, the first step in the analysis is to use PHYLIP to bootstrap the MSA. The evolutionary trace method is then run on the 1000 bootstrapped samples. The trace integral scores are then calculated for each bootstrapped sample and are compared to the original integral. The histogram of these values is Figure 5.4.

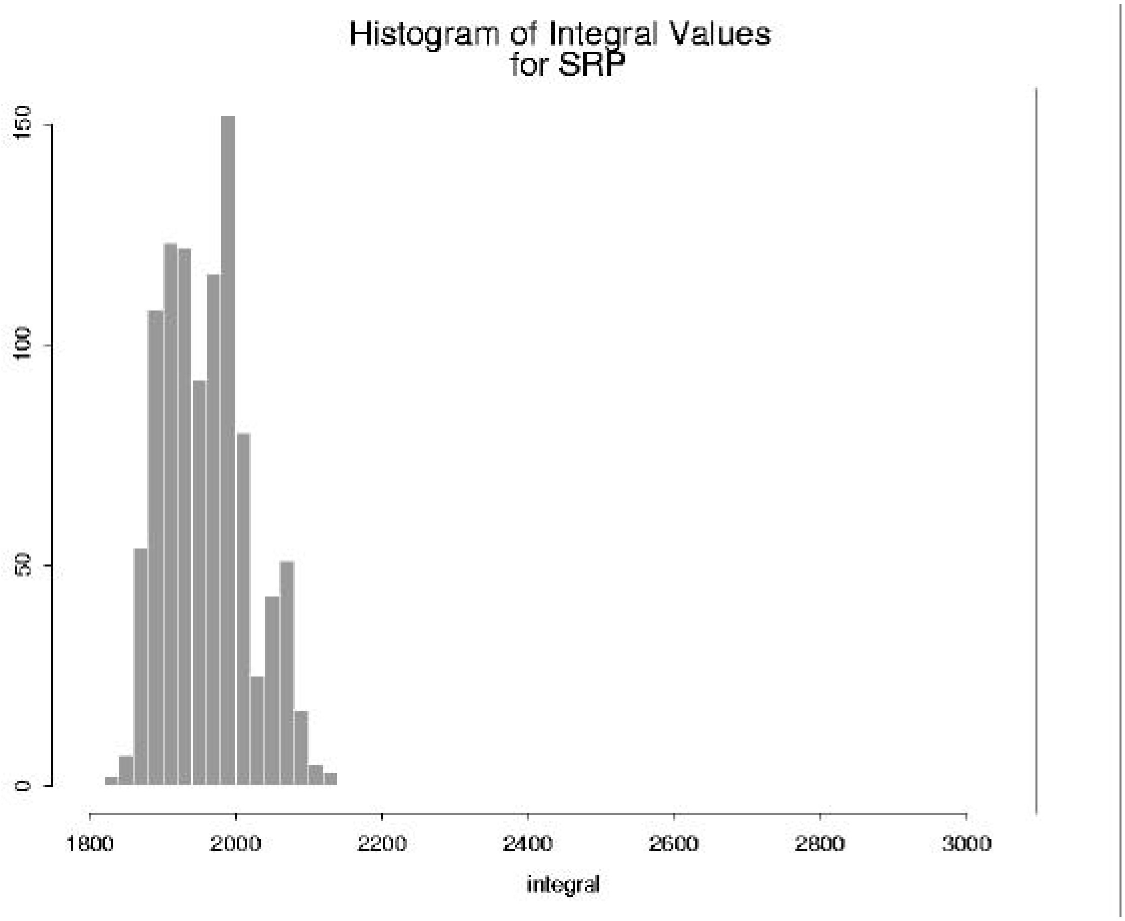

Figure 5.4: Histogram of integral values for the bootstrapped samples of the SRP protein. The vertical line indicates the value, 3095, of the integral for the original MSA.

As in the case of the YRS protein, the value of the original integral is a great deal higher than for any of the bootstrapped samples. Thus, the decision was made to look at a new MSA of the SRP protein: one with fewer gaps. All non-important gaps were removed from the original alignment. New bootstrapped samples were created using this shorter alignment which contained only 294 residues. The trace was then re-run using these shorter bootstrapped alignments, and new integrals were calculated. The resulting histogram of integral values is Figure 5.5. This time the value for the original integral actually falls on the histogram, not way off to the right of it. In this case, there are only 21 bootstrapped samples that have an integral equal to or better than the original trace.

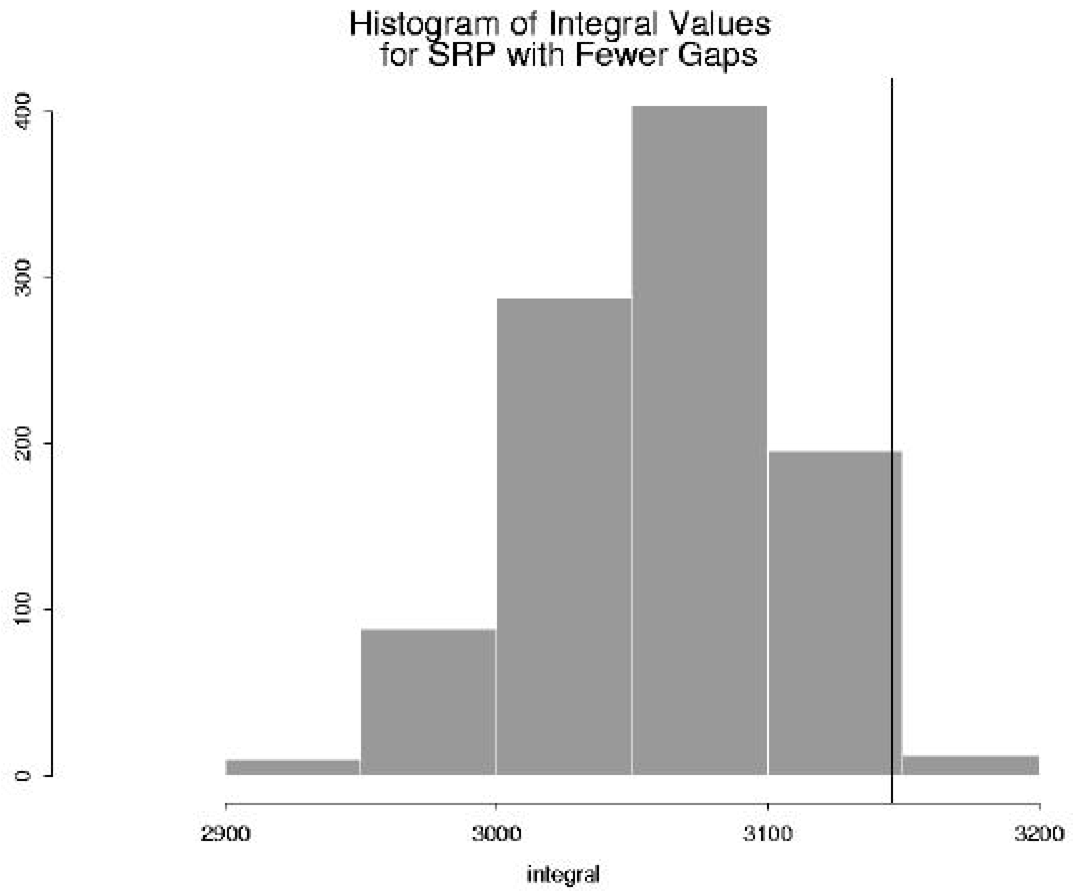

Figure 5.5: Histogram of the integral values for the bootstrapped samples of the SRP protein with fewer gaps. The vertical line indicates the integral value, 3146, of the original MSA with fewer gaps.

The next step is to look at each of the bootstrapped samples with a better integral and see if the important residues are altered very much. It was deemed that for the SRP protein a cut-off value of 19 was sufficient. This means that when the tree has 19 branches all the functionally important residues will be present. It is pure coincidence that the YRS protein and the SRP protein have the same cut-off value. Again, the correlation analysis shows that all the bootstrapped samples are highly correlated with the original sample. Thus, the important residues do not appear to change much from one bootstrapped sample to another or vary much from the original.

Chapter 6

Future Work

Two areas of interest regarding the Evolutionary Trace procedure will be examined. The first is based on incorporating the maximum likelihood tree into the procedure. The second deals with figuring out where the noise exists in the trace signal.

6.1 Maximum Likelihood Tree for Protein Sequences

In an effort to improve the phylogenetic tree used in the evolutionary trace calculation, I would like to be able to use the tree created by the method of maximum likelihood instead of that created by the UPGMA method. The maximum likelihood tree is thought to be a better overall tree than that of the UPGMA tree. Also, the likelihood of a tree is a statistical measure of its compatibility with the data.

At the present time, a program, called MOLPHY (Adachi and Hasegawa 1996), does exist to calculate the maximum likelihood tree. However, MOLPHY works only when a small number of taxa (species) are involved. Most of the proteins that we will be working with contain at least twenty different taxa. I would like to be able to make some minor changes to MOLPHY to be able to use twenty or more sequences to create a maximum likelihood tree. The method of maximum likelihood should be a better tree building method to use with the ET method and will hopefully produce the best results.

6.2 Noise-level Detection

No method currently exists to define the appropriate cut-off level of the phylogenetic tree from one protein to the next. The method in use presently is eyeball detection. I wish to give some statistical reasoning for choosing an appropriate cut-off level. One way to accomplish this is through the use of a statistic called "mutual information". Mutual information (Applebaum 1996), which comes from information theory, measures the amount of association between two sites of a sequence beyond that expected to occur by random chance (Wollenberg and Atchley 2000). The mutual information, denoted MI, between two sites X and Y is calculated as

$$MI_{XY} = \sum_{i=1}^{20} \sum_{j=1}^{20} P(X_i, Y_j) \log_{20} \frac{P(X_i, Y_j)}{P(X_i) P(Y_j)} \qquad (6.1)$$

where $P(X_I)$ is the probability of amino acid i at site X, $P(Y_j)$ is the probability of amino acid j at site Y, and $P(X_i, Y_j)$ is the joint probability of amino acid i at site X and amino acid j at site Y (for $X \neq Y$). When sites are independent, the MI is zero. Smaller MI values result from sites with less variation. On the other hand, the maximum MI value occurs when the variation of two different sites is perfectly correlated (Wollenberg and Atchley 2000). The maximum possible MI value is one, which will occur when all the residues are uniformly distributed.

In order to achieve the proper value of functional resolution that is not subject to inspector error, tail distributions of MI values are calculated. This is a distribution which decreases in value as the independent variable increases (Wollenberg and Atchley 2000). The tail distributions are obtained by subtracting the cumulative frequency within a given range of MI values from one. Setting threshold values will provide an objetive method of detecting those sites which are functionally important.

Bibliography

Adachi, J. and Hasegawa, M. (1996) MOLPHY Version 2.3: Programs for Molecular Phylogenetics Based on Maximum Likelihood. *Computer Science Monographs, Number 27.* Institute of Statistical Mathematics, Tokyo.

Akaike, H. (1974) A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control.* AC-19: 716-723.

Applebaum, D. (1996) *Probability and Information: an Integrated Approach*, Cambridge University Press (New York, NY).

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.

Baldi, Pierre and Brunak, Soren. (1998) *Bioinformatics: The Machine Learning Approach,* Massachusetts Institute of Technology Press (Cambridge, MA).

Blackstock, J.C. (1998) *Biochemistry,* Butterworth-Heinemann (Boston, MA).

Curtis, H. and Barnes, N.S. (1989) *Biology: Part 1 Biology of Cells,* Worth Publishers (New York).

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. (1978) A Model of Evolutionary Change in Proteins. In Dayhoff, Mo.O., ed., *Atlas of Protein Sequence and Structure*, volume 5, supplement 3. National Biomedical Research Foundation, Washington D.C., pp. 345-352.

Dempster, A.P., Laird, M.N., and Rubin, D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B*, 39:1-38.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998) *Biological Sequence Analysis: Probabilistic Models of Protein and Nucleic Acids*, Cambridge University Press (New York).

Edwards, A.W.F. and Cavalli-Sforza, L. (1964) Reconstruction of Evolutionary Trees. In Heywood, V.H. and McNeill, J., eds., *Phenetic and Phylogenetic Classification.* Systematics Association Publication No. 6. pp. 67-76.

Efron, B. (1979) Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics,* 7:1-26.

Felsenstein, J. (1978) Cases in which Parsimony and Compatibility Methods Will Be Positively Misleading. *Systematic Zoology*, 27:401-410.

Felsenstein, J. (1981) Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach. *Journal of Molecular Evolution,* 17:368-376.

Felsenstein, J. (1985) Confidence Limits on Phylogenies: An Approach using the Bootstrap. *Evolution*, 39(4): 783-791.

Felsenstein, J. (1993) PHYLIP (Phylogeny Inference Package) version 3.5c. Distributed by the author. Department of Genetics, University of Washington, Seattle.

Felsenstein, J. (1996) Inferring Phylogenies from Protein Sequence by Parsimony, Distance, and Likelihood Methods. *Methods in Enzymology,* 24: 418-427.

Feng, D. and Doolittle, R. (1987) Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees. *Journal of Molecular Evolution,* 25:351-360.

Grassly, N.C., Adachi, J., and Rambaut, A. (1997) Pseq-Gen: an Application for the Monte Carlo simulation of protein sequence evolution along phylogenetic trees. *Computer Applications in the Biosciences,* 13(5): 559-560.

Higgins, D.G. and Sharp, P.M. (1989) Fast and Sensitive Multiple Sequence Alignments on a Microcomputer. *Computer Applications in the Biosciences,* 5:151-153.

Hillis, D.M., Mable, B.K., Larson, A., Davis, S.K., and Zimmer, E.A. (1996) Nucleic Acids IV: Sequencing and Cloning. In *Molecular Systematics,* eds. Hillis, D.M., Moritz, C. and Mable, B.K. (Sinauer, Sunderland, MA), 2nd Ed., pp.321-381.

Hulesenbeck, J.P., Hillis, D. M., and Jones, R. (1996) Parametric Boostrapping in Molecular Phylogenetics: Application and Performance. In *Molecular Zoology: Advances, Strategies, and Protocols,* eds. Ferraris, J.D. and Palumbi, S.R. (Wiley-Liss, New York), pp.19-45.

Jukes, T.H., and Cantor, C. (1969) Evolution of Protein Molecules. In *Mammalian Protein Metabolism,* ed. Munro, H.N. (Academic Press, New York). pp. 21-132.

Kishino, H., Miyata, T., and Hasegawa, M. (1990) Maximum Likelihood Inference of Protein Phylogeny and the Origin of Chloroplasts. *Journal of Molecular Evolution,* 31:151-160.

Li, Wen-Hsiung. (1997) *Molecular Evolution,* Sinauer Associates, Inc. (Sunderland, Massachusetts).

Lichtarge, O., Bourne, H.R., and Cohen, F.E. (1996) An Evolutionary Trace Method Defines Binding Surfaces Common to Protein Families. *Journal of Molecular Biology,* 257:342-358.

Lio, Pietro and Goldman, Nick. (1998) Models of Molecular Evolution and Phylogeny. *Genome Research,* 8:1233-1244.

Matsuda, H. (1996) Protein Phylogenetic Inference Using Maximum Likelihood with a Genetic Algorithm. *Pacific Symposium on Biocomputing*. pp. 512-523.

Needleman, S.B. and Wunsch, C.D. (1970) A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology,* 48:443-453.

Sayle, Roger and Milner-White, W. James. (1995) *RasMol: Biolmolcular Graphics for All. Trends in Biochemical Sciences,* 20:374.

Sokal, R.R. and Michener, C.D. (1958) A Statistical Method for Evaluating Systematic Relationships. *University of Kansas Scientific Bulletin,* 28:1409-1438.

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

Wollenberg, Kurt R. and Atchley, William R. (2000) Separation of Phylogenetic and Functional Associations in Biological Sequences by using the Parametric Bootstrap. *Proceedings of the National Academy of Science,* 97 (7):3288-3291.