BAYESIAN EVOLUTIONARY TREE ESTIMATION

Kimmen Sjölander
Molecular Applications Group
P.O. Box 51110, Palo Alto, CA 94303-1110
kimmen@mag.com
(650)846-3589

March 30, 1998

Abstract

This work involves the inference of evolutionary trees for proteins, using relative entropy, a distance metric from information theory, in combination with Dirichlet mixture densities over amino acid distributions, to identify key structural or functional positions in the molecule, and constrain tree topologies to preserve these important positions within subtrees.

Experimental results suggest that this method, Bayesian Evolutionary Tree Estimation (Bête), provides several advantages over existing tree-estimation methods. It is robust with respect to differing evolutionary clocks among taxa, differing mutation rates at sites in the molecule, handles deletions of portions of the molecule among taxa, and produces tree topologies that agree more closely with accepted phylogenies and functional subgroups within the data. Bête is also computationally efficient in the number of taxa, so that large numbers of sequences (in the hundreds) may be used as input to the tree-estimation process.

1 Introduction

This work was originally motivated by the need to identify functional subfamilies in proteins, and the residues mediating these functional specificities, during the UCSC participation in the Second Meeting on the Critical Assessment of Techniques for Protein Structure Prediction (CASP2) [Eis97, KKB+97a, KKB+97b].[1]

In this contest, we attempted to identify the three-dimensional fold of proteins whose structures were unknown (the "target" proteins), based on discernible homology to proteins of solved structure. All target proteins had been screened to ensure none had a homolog that was recognizable by pairwise sequence comparison, making the set of target proteins a challenging one.

We found that for each target studied, we often had a rather large pool of candidate homologs which all appeared outwardly to be equally similar to the target. Filtering through these candidates for the best match required a lot of *post hoc* analysis and was error-prone. Much of this analysis involved examining multiple alignments between the target and a (sometimes large and diverse) protein family. In confirming a remote homology between the target and such a family, it became obvious that a decomposition of the larger family into functional subfamilies would be extremely helpful. This would enable us to examine the alignment of the target to each subfamily separately, and identify the closest possible match within the large family.

The method I developed to identify the functional subfamilies in a protein family has two stages. In the first stage, I estimate an evolutionary tree for the family. In the second stage, I employ minimum-description length principles to cut the tree into subtrees and determine the subfamilies in the data.

For reasons of space, this paper will focus on the method for estimating an evolutionary tree employed in the first stage. The method for cutting the tree into subtrees to identify the subfamilies is covered in UCSC Technical Report UCSC-CRL-97-14 [Sj97].[2]

The computational efficiency of this method ($O(mn^2 \log n)$, where n is the number of sequences in the alignment, and m is the number of columns in the alignment) allows it to be employed on alignments of large numbers of sequences. This makes it possible to estimate evolutionary trees for data that parsimony and maximum likelihood methods—two classes of methods that are presumed to be the most accurate for real biological sequences, but which have strong limitations in the number of taxa—cannot handle. Moreover, this method produces trees whose topologies appear to agree more closely with what is known about the functions of the proteins than the trees produced by other methods, especially when some of the mathematical assumptions made by these methods are not met in the data.

1.1 Evolutionary processes and macromolecules

Two primary processes underlie the evolution of protein molecules: *speciation*, involving the relationships between organisms, and *gene duplication*, involving the relationships between families of molecules. Both are of interest in the context of inferring evolutionary trees, but yield different types of information. For clarity, I use the term "phylogenetic" to refer only to the first type of relationship, and use the term "evolutionary" to refer to the more general type of relationship which includes all types of evolutionary processes, including the strictly phylogenetic.

[1] In fact, the method described in this paper was never applied within the CASP2 contest, due to time limitations.

[2] This paper is available by anonymous ftp from ftp.cse.ucsc.edu, in pub/protein/phylogeny, as a compressed postscript file, techrep_97-14.ps.Z.

Orthologous evolution refers to evolutionary changes in homologous molecules due to speciation; the gene present in the common ancestor has undergone changes in each of the descendents, but the lines of descent are straight.

Paralogous evolution refers to gene duplication events. Paralogous proteins can be produced by one or multiple gene duplication events, with subsequent divergence over time. The presence of two copies of a gene, only one of which needs to maintain the original function, provides a degree of freedom for one copy to diverge and obtain new functionality. This process of duplication and divergence can produce a variety of protein functions and structures. An example of paralogous evolution is the relationship among the various globins, such as the alpha, beta, gamma, delta, zeta and epsilon hemoglobin chains that bind oxygen in the blood and are expressed in various stages of development among vertebrates, their relationships to myoglobins, which bind oxygen in muscle tissues [Str95].

1.2 Trees as models of evolution

These evolutionary relationships have traditionally been represented using mathematical objects called trees. In graph theory, trees are connected, acyclic graphs, composed of *nodes* (or *vertices*) and *edges*. Nodes can be internal, or terminal, in which case they are called *leaves*. Trees generally, but not always, have a distinguished internal node called the *root*. If we orient the tree so that the root is at the top of the page, and the tree grows downward, then nodes or leaves which are attached to internal nodes higher up in the tree are called the *children* of the internal node. Each tree contains one or more *subtrees*, defined by a node and the tree structure descending from that node to the leaves.

Trees that reflect the strictly phylogenetic relationships are typically called *species trees*, while those that reflect gene duplication events are called *gene trees*. In fact, trees for supergene families will show both types of relationships: at a high level, the tree as a whole will be a gene tree, but at a lower level, subtrees for individual genes will be species trees. Because we are often interested in identifying the evolutionary relationships among proteins in supergene families, the term *taxa* is used in this paper to indicate a protein sequence per se, and is not restricted to mean a particular species.

When trees are used to describe the evolutionary relationships among a set of proteins, the protein sequences are placed at the leaves, and the root represents the (inferred) common ancestor of these taxa. An internal node represents the common ancestor among the taxa at the leaves of the subtree descending from the node. Edge lengths are used to represent the estimated evolutionary distance between taxa at the leaves and inferred ancestors, at internal nodes, with longer edges implying larger evolutionary distances.

A subtree is called *monophyletic* for a group of taxa if the subtree contains all the sequences belonging to that group, and no others. This term is often used to refer to taxa related by speciation (i.e., following the standard use of phylogenetic trees). In the framework of examining a gene tree, we call a subtree monophyletic for a particular gene if it contains all the taxa for that gene, and no others.

1.3 Current methods for estimating evolutionary trees

To take advantage of the explosion of genomic sequence data, a large number of evolutionary tree estimation methods have been developed. The first observation to make is that it is clearly not

computationally feasible to explore all possible tree topologies for large sets of taxa, since the number of tree topologies grows exponentially in the number of taxa [HMM96].

Because of this combinatorial explosion, heuristic methods are used for datasets having more than 11 or so taxa (corresponding to 34,459,425 trees); exact methods, which explore all possible tree topologies (such as Maximum Likelihood), are restricted for smaller numbers of taxa.

Maximum Likelihood methods have been proved to converge on the true tree in the limit of infinitely long sequences. [3] Unfortunately, most proteins come in sizes too small to satisfy this criterion, of at most 200–300 amino acids or so.

However, alignments of supergene families can contain a large amount of information, but in a different dimension. In this case, we may not have very *long* sequences, but we are likely to have many distinct examples from the supergene family of interest.

Just as many diverse sequences from a single-gene family can highlight positions which are conserved for functional or structural reasons, when several observations from each individual gene family are available, the multiple sequence alignment (MSA) of a supergene family can highlight the different evolutionary constraints at positions in the common structure. Some positions in an MSA will appear highly variable, while others will be perfectly conserved across large evolutionary distances. Still others will be conserved within functional subfamilies, but differ across subfamilies; changes in residues at these positions are correlated with changes in functional specificity [CSV95, OHF96]. Identifying each of these position types is crucial for correct evolutionary tree inference.

A recent talk discussed in *Science* [Bal97] highlighted the importance of discriminating informative from non-informative positions in proteins for estimating evolutionary trees. In this talk, evolutionary biologist Gavin Naylor presented sequence data for which a large body of fossil and morphological evidence supported a particular evolutionary tree topology. When alignments of these sequences were used as input to evolutionary tree estimation programs, the programs produced tree topologies that disagreed with the (presumably) correct tree. But when Naylor restricted the alignment columns to only a subset—in particular, those positions known to be important for determining the protein's three-dimensional structure—the same methods that had produced (presumably) incorrect phylogenetic trees based on the entire alignments, now produced tree topologies that agreed with the tree based on morphological and fossil evidence.

But what if there are no solved structures for the proteins being analyzed, so that we do not know *a priori* which are the important positions?

Bayesian Evolutionary Tree Estimation (Bête) is designed specifically to identify these important positions directly from the primary sequence information alone, and to use these positions to constrain the tree topology as it is being estimated. It is especially well-suited to leverage the information contained in large multiple alignments.

Many methods for estimating evolutionary trees employ simplifying mathematical assumptions—such as that taxa in a family are equally distant from the common ancestor, that positions mutate at the same rate and according to the same underlying distribution, and so on—in order to make the mathematics tractable, even though there are numerous counter-examples to these assumptions in biology. The degree to which a method's accuracy at estimating a correct evolutionary tree when the data violate the assumption has been tested using simulated data experiments [Yan96, YNM94, MHN91, Yan94, RW97, CSMT95, KF94, Fel96, Fel96]. Very few experiments, however, have been done to validate the tree topologies produced on real biological data, especially in the case of protein alignments of diverse sequences. For reasons of space, this paper presents results on a single protein family (see Section 4), and summarizes extensive experimental validation

[3] In this case, "truth" is relative to the mathematical model implicit to the method.

on a variety of protein families, a selection of which is given in the Technical Report mentioned earlier [Sj97].

2 Mathematical Foundations

Bayesian Evolutionary Tree Estimation employs tools developed in two areas: Information Theory (primarily entropy measures) and Dirichlet mixture priors over amino acid probabilities. This section gives the basics needed to understand the method.

2.1 Entropy

For an excellent and general overview of information theory, see [CT91].

The *entropy* H of a random variable X which takes on different values $i = 1 \ldots n$ with probability p_i is defined to be

$$H(X) = -\sum_i p_i \log p_i \tag{1}$$

H(X) is the average encoding cost, and measures the uncertainty in the random variable X. If X takes on exactly one value with probability 1, then $H(X)$ is zero. This value is maximized when each possible value X can take is equally likely, which agrees with our intuition that such a random variable has a maximum degree of uncertainty.

Relative entropy, also known as *Kullback-Leibler divergence*, is a measure of the average additional cost to encode a random variable with distribution p using distribution q instead of distribution p. The relative entropy between two probability distributions p and q is defined to be

$$D(p||q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{2}$$

Although not a true distance metric (being neither symmetric, nor obeying the triangle inequality), it is a useful metric for examining the similarity between two distributions. When the relative entropy between the means of all pairs of components in the nine-component Dirichlet density given in Figure 1 was examined, this metric correlated directly with the physico-chemical description of the components; small relative entropies were associated with components preferring amino acids with similar physico-chemical attributes, and large relative entropies were associated with components preferring very different amino acids. This makes this metric appealing for examining the similarity between profiles constructed for homologous proteins.

2.2 Dirichlet mixture priors

We originally developed the use of Dirichlet mixture densities for proteins to assist in identifying remote homologies, especially in the case where available training data for a statistical model was limited, or highly correlated.[4]

We address this problem by incorporating prior information about amino acid distributions that typically occur in columns of multiple alignments into the process of building a statistical model. These *Dirichlet* [BS94, Ber85, SD89] densities over amino acid distributions condense the

[4]This section contains a very high-level overview of Dirichlet mixture densities, and the use of these densities to estimate amino acid distributions in profiles or hidden Markov models. For a complete exposition of this method, including full derivations and formulas, see [SKB+96].

information in databases of multiple alignments, and are combined with the observed amino acids at a position in a multiple alignment to form a more effective estimate of the expected distribution at that position among homologous proteins. Often, these densities capture some prototypical distributions. Taken as an ensemble, they explain the observed distributions in columns of multiple alignments.

A *Dirichlet mixture density* over amino acid distributions is a mixture of individual Dirichlet densities, each of which prefers particular amino acid distributions, and which function jointly to assign probabilities to all amino acid distributions.

The formula employed to estimate the posterior probability of amino acid i ($\hat{p}_i$) at a position in an alignment, given the observed amino acid counts $\vec{n} = n_1 \ldots n_{20}$, and the Dirichlet mixture density θ, with parameters $\vec{\alpha}_1, \ldots, \vec{\alpha}_l$, $q_1 \ldots, q_l$ (where $\vec{\alpha} = \alpha_1 \ldots \alpha_{20}$ are the parameters of the densities, and the q are the *mixture coefficients*), is

$$\hat{p}_i := \sum_{j=1}^{l} \text{Prob}\left(\vec{\alpha}_j \;\middle|\; \vec{n}, \Theta\right) \frac{n_i + \alpha_{j,i}}{|\vec{n}| + |\vec{\alpha}_j|} \; . \tag{3}$$

In this formula, Prob $(\vec{\alpha}_j \mid \vec{n}, \Theta)$ represents the posterior probability of the j^{th} component of the mixture, given the observed amino acids $\vec{n}$, and $|\vec{n}|$ and $|\vec{\alpha}_j|$ represent the total number of amino acids observed in the column and the sum of the parameters of the j^{th} component in the mixture, respectively.

Thus, instead of identifying one single component of the mixture that accounts for the observed data, we determine how likely each individual component is to have produced the data. Each component then contributes pseudocounts proportional to the posterior probability that it produced the observed counts.

With accurate prior information about which kinds of amino acid distributions are reasonable in columns of alignments, it is possible with only a few sequences to identify which prototypical distribution may have generated the amino acids observed in a particular column. Using this informed guess, we adjust the expected amino acid probabilities to include the possibility of amino acids that may not have been seen but are consistent with observed amino acid distributions. This method interpolates smoothly between reliance on prior beliefs concerning likely amino acid distributions, in the absence of data, and confidence in the amino acid frequencies observed at each position, given sufficient data (see Table 3). In comparison with other methods for estimating amino acid distributions from multiple alignment columns, the statistical models produced are more effective at generalizing to previously unseen data, and are often superior at database search and discrimination experiments [HK96, Kar95, BE95, TAK94, BHK$^+$93].

Table 1 shows the preferred amino acids for each component in the Dirichlet mixture prior Blocks9.

2.3 Methods for estimating amino acid distributions

This section shows results obtained when estimating the expected amino acids using two methods: Dirichlet mixture priors and substitution matrices. Tables 2 and 3 give examples of the different results produced by these methods given a varying number of isoleucines observed (and no other amino acids).

Analysis of 9-Component Dirichlet Mixture Prior Blocks9								
Comp.	Ratio (r) of amino acid frequency relative to background frequency							
	$8 \leq r$	$4 \leq r \leq 8$	$2 \leq r \leq 4$	$1 \leq r \leq 2$	$1/2 \leq r < 1$	$1/4 \leq r < 1/2$	$1/8 \leq r < 1/4$	$r < 1/8$
1			SAT	CGP	NVM	QHRIKFLDW	EY	
2	Y	FW	H		LM	NQICVSR	TPAKDGE	
3			QE	KNRSHDTA	MPYG	VLIWCF		
4		KR	Q	H	NETMS	PWYALGVCI	DF	
5		LM	I	FV		WYCTQ	APHR	KSENDG
6		IV		LM	CTA	F	YSPWN	EQKRDGH
7		D	EN	QHS	KGPTA	RY	MVLFWIC	
8			M	IVLFTYCA	WSHQRNK	PEG	D	
9			PGW	CHRDE	NQKFYTLAM	SVI		

Table 1: Preferred amino acids of Dirichlet Mixture Prior Blocks9

The function used to compute the ratio of the frequency of amino acid i in component j relative to the background frequency predicted by the mixture as a whole is $\frac{\alpha_{j,i}/|\vec{\alpha}_j|}{\sum_k q_k \alpha_{k,i}/|\vec{\alpha}_k|}$. When an amino acid has a frequency ratio (r) larger than 1, it is more likely than the background distribution. Conversely, amino acids having small frequency ratios are less likely than the background distribution.
An analysis of the amino acids favored by each component reveals the following:

Component 1 favors small neutral residues.

Component 2 favors the aromatics.

Component 3 gives high probability to most of the polar residues (except for C, Y, and W).

Component 4 gives high probability to positively charged amino acids and residues with NH_2 groups.

Component 5 gives high probability to residues that are aliphatic or large and non-polar.

Component 6 prefers I and V (aliphatic residues commonly found in Beta sheets), and allows substitutions with L and M.

Component 7 gives high probability to negatively charged residues, allowing substitutions with certain of the hydrophilic polar residues.

Component 8 gives high probability to uncharged hydrophobics, with the exception of G.

Component 9 gives high probability to distributions peaked around individual amino acids (especially P, G, W, and C).

2.3.1 Substitution matrices

The need for incorporating prior information about amino acid distributions into protein alignment motivated the development of amino acid substitution matrices. These have been used effectively in database search and discrimination tasks [HH93, JTT92, GBH90, HH92, Alt91, Cla94, RJ94], and are also used to compute the evolutionary distances between taxa in evolutionary tree estimation methods.

There are two drawbacks associated with the use of substitution matrices. First, each amino acid has a fixed substitution probability with respect to every other amino acid. In any particular substitution matrix, to paraphrase Gertrude Stein, a serine is a serine is a serine. However, a serine seen in one context, for instance, an active site position requiring serine for interaction with a particular substrate [Bre88], will have different substitution probabilities than a serine seen in a context requiring any polar residue. Second, only the relative frequency of amino acids is considered, while the actual number observed is ignored. Thus, in substitution-matrix-based methods, the expected amino acid probabilities are identical for any pure serine column, whether it contains 1, 3, or 100 serines. All three situations are treated identically, and the estimates produced are indistinguishable (see Table 2 for amino acid probabilities produced using the popular substitution

	Expected amino acids using Substitution Matrix Blosum62		
	Given 1 Isoleucine	Given 3 Isoleucines	Given 10 Isoleucines
A			
C			
D			
E			
F			
G			
H			
I			
K			
L			
M			
N			
P			
Q			
R			
S			
T			
V			
W			
Y			

Table 2: Posterior estimates of the amino acids given 1, 3 and 10 isoleucines observed in a column (with no other amino acids), using substitution matrix Blosum62. Note that the posterior estimate of the amino acids does not change as the number of isoleucines increases, but generalizes the amino acids observed to reflect substitution probabilities for isoleucine in general.

matrix Blosum62 [HH92]).

2.3.2 Dirichlet mixture priors

Dirichlet densities address the inability of substitution matrices to represent more than one context for the amino acids by the multiple components of Dirichlet mixtures. These components enable a mixture to represent a variety of contexts for each amino acid. For example, the mixture density shown in Table 1 presents several contexts for isoleucine. A pure isoleucine distribution would be given high probability by component 9, which gives high probability to all conserved distributions. Components 5, 6, and 8 prefer isoleucine found in combination with other amino acids.

Dirichlet mixtures also address the second drawback associated with substitution matrices—the importance of the actual number of residues observed—in the formula used to compute the expected amino acids (Equation 3). In this formula, given no observations, the estimated amino acids probabilities approximate the background distribution. But as more data becomes available, the estimate for a column becomes increasingly peaked around the maximum likelihood estimate for that column (i.e., $\hat{p}_i$ approaches $n_i / |\vec{n}|$ as $|\vec{n}|$ increases). Importantly, when the data indicate a residue is conserved at a particular position, the expected amino acid probabilities produced by this method will remain focused on that residue, instead of being modified to include all the residues that substitute on average for the conserved residue. This is demonstrated in Table 3.

	Expected amino acids using Dirichlet mixture prior Blocks9		
	Given 1 Isoleucine	Given 3 Isoleucines	Given 10 Isoleucines
A			
C			
D			
E			
F			
G			
H			
I			
K			
L			
M			
N			
P			
Q			
R			
S			
T			
V			
W			
Y			

Table 3: Posterior estimates of the amino acids given 1, 3 and 10 isoleucines observed in a column (with no other amino acids), using Dirichlet mixture prior Blocks9. Note that the posterior estimate of the amino acids given a single isoleucine shows fairly significant probability for valine and leucine—amino acids very similar to isoleucine—as well as all other amino acids. But as the number of isoleucines observed increases, the posterior estimate becomes more peaked around isoleucine, until at 10 isoleucines, the posterior estimate appears fairly conserved.

3 Bayesian Evolutionary Tree Estimation (Bête)

The method employed in this work to construct an evolutionary tree falls within a hierarchical clustering paradigm known as *agglomerative clustering* using *nearest neighbor* heuristics. Initially, each sequence is in its own class, and forms a leaf of the tree. At each iteration of the algorithm, the two closest classes are merged (agglomerated), until at termination all sequences are in a single class, forming the root of the tree.

The outer aspects of the algorithm to construct the tree are certainly not new, and have some outward similarities to other methods used to infer evolutionary trees, particularly neighbor-joining or other distance methods, such as that used by ClustalW. What gives this tree-construction approach a new twist is how the classes are represented at each step, and the distance metric used to determine the nearest neighbors at each step.

A *partition* on the sequences in an alignment is a division of the sequences into separate non-overlapping subsets, or equivalence classes. Each stage of the agglomerative algorithm defines a distinct partition of the tree into subtrees, which defines a partition of the sequences into subfamilies.

Each such partition induces a different set of multiple alignments, one for each subfamily. As classes are merged, the observed amino acid counts at each position are combined, and a profile of expected amino acids for each position in the multiple alignment for the class is created. These profiles are computed by combining the total observed amino acids in the class at each position, weighted appropriately (see Section 3.3), with a Dirichlet mixture prior (Equation 3) over

amino acid distributions. I employ a symmetrized form of relative entropy (a distance metric from Information Theory) to determine the two classes to merge at each step.

This metric, the Total Relative Entropy (TRE), is defined to be

$$TRE = \sum_c D(i_c \,\|\, j_c) + D(j_c \,\|\, i_c) \tag{4}$$

where i_c and j_c are the probability distributions at position c in the profile for the i^{th} and j^{th} classes respectively, and the relative entropy between distributions i_c and j_c is defined as in Equation 2.

3.1 The algorithm to estimate an evolutionary tree

Input: A multiple alignment[5]

1. Initially, let each sequence form a separate class, corresponding to a leaf in the tree. Create a profile for each row in the alignment, using Dirichlet mixture densities to form a posterior estimate for the expected amino acids at each position given the single amino acid (or indel character[6]) observed, as shown in Equation 3.

2. While the number of classes in the partition is greater than 1, do:

   (a) Compute the total relative entropy (TRE) (Equation 4) between every pair of profiles.

   (b) Find the pair giving the lowest TRE.

   (c) Replace these two classes with a single class that combines the counts at each position. Form an internal node to represent this new class, adding edges to the nodes (or leaves) representing the classes joined. This reduces the number of classes in the partition by 1.

   (d) Estimate the number of independent observations in the new class, and weight the sequences accordingly (see Section 3.3).

   (e) Create a profile of the expected amino acids at each position for the new class using the weighted counts, in combination with a Dirichlet mixture prior, as shown in Equation 3.

By only computing the TRE as needed (that is, between the newly formed class and remaining classes in step 2(a), and employing a priority queue to pick the lowest TRE score during step 2(b)) we obtain an algorithm that runs in $O(MN^2 \log N)$, where M is the number of columns and N is the number of *taxa*, or sequences, in the alignment.

[5]The program will tolerate a degree of noise in the alignment. Obviously, the better the alignment, the more accurate the results. The alignment should omit regions that cannot be reliably multiply aligned, such as loop regions on the exposed surface of the protein. Because of this possible omission of residues in some proteins in an input alignment, the term "sequence" used below may refer not to the entire protein sequence, but to only a segment (or segments) of a protein.

[6]In the case of an indel, where no amino acid is aligned at a position, $n_i = 0$ for all i, and the posterior estimate of the expected amino acids approximates the background distribution in the data used to estimate the Dirichlet mixture density.

3.2 Identifying key positions to guide tree topology estimation

Evolutionary tree inference methods have been shown to be erroneous when all positions in an alignment are treated as equally informative [Bal97]. This section shows how Bayesian Evolutionary Tree Estimation infers the key functional and structural positions automatically, and uses these positions to guide the construction of an evolutionary tree for sequences in a multiple alignment.

Perfectly conserved amino acids are generally important for functional or structural reasons, but are not helpful for distinguishing between alternative branching orders in trees. The positions that are the most informative are those that mediate the functional or structural specificity of the subfamilies in the data. These positions show up as conserved within the subfamilies, but may change from one subfamily to another [CSV95]. Mutations of these positions are generally not well tolerated specifically because they change the function or structure of the protein, while mutations at other positions may be easily tolerated, because the overall function or structure of the protein is not affected.

As Table 4 shows, the method employed here to estimate an evolutionary tree works as an implicit weighting scheme on the columns in the multiple alignment which favors joining two classes if they are similar (or identical) at positions showing high conservation (low tolerance for mutation), and favors keeping two classes separate if they are dissimilar at such positions. During the agglomeration process, as increasingly divergent sequences are added to the classes being formed, conserved positions start to become evident. These conserved positions have a large impact on the TRE between two profiles; positions showing higher tolerance for mutation (the more mixed distributions) have less impact on the TRE between two profiles. This helps constrain the tree topologies produced to maintain conserved distributions within subtrees corresponding to functional subfamilies, and construct tree topologies that reflect the functional hierarchy in the data.

3.3 The role of sequence weighting in tree construction

The importance of sequence weighting in estimating profiles or hidden Markov models for remote homolog detection is well known [KKB$^+$97a]. It has proved to be important in this work, as well.

Almost any set of homologous proteins will contain some highly populated subfamilies and some less populated subfamilies, and a model constructed from it will favor the most highly represented sequences. To reduce training-set bias, sequence-weighting schemes assign relative weights to training sequences [Cla94, SA90, THG94b, THG94a, HH94, ACL89, VS93]. This assignment of relative weights, without regard to the magnitude of the total weights, is appropriate with substitution-matrix-based methods for estimating amino acid distributions, since these methods take into account only the relative frequencies of the amino acids at each position.

However, in Bayesian methods (such as our use of Dirichlet mixtures, as shown in Equation 3), the total weight assigned to the set of training sequences has a large impact on the posterior amino acid distributions produced. When no data is available (for example, when a column shows only indel characters), the posterior estimate is close to the background frequencies of amino acids. Given few observations (or low total weight), the prior dominates the equation, and the posterior amino acid distributions will be generalized to include amino acids similar to those observed. As the total weight allotted the sequences increases, the posterior amino acid estimate will reflect the frequencies in the data. By adjusting the total weight, one can smoothly interpolate between the background frequencies and observed data frequencies, with intermediate weights giving very natural generalizations.

When we estimate a profile for a subfamily during the tree-construction process, we need to

Relative entropy between distributions in profiles during Bête tree estimation		
	Similar Residue Types	Dissimilar Residue Types
Conserved Distribution	Low (or zero)	Large
Mixed Distribution	Low (or zero)	Moderate

Table 4: Interaction between relative entropy and amino acid distributions in profiles in Bête tree estimation.

Relative entropy and Bayesian estimation of amino acid probabilities in profiles work together to guide evolutionary tree topologies to preserve functionally or structurally conserved positions.

This table shows the relative entropy between two distributions for four types of cases: conserved distributions *agreeing* on the same amino acid, conserved distributions *disagreeing* on the amino acid conserved, mixed distributions preferring the same type of amino acids, and mixed distributions preferring different types of amino acids. The symmetrized relative entropy (Equation 4, fixed for a single column c, instead of summing over all columns) is largest when two distributions are conserved for different amino acids, especially when the amino acids are of different types. This value is smallest when the distributions are conserved for the same amino acid. In the case where the two distributions are mixed (not showing a strong preference for a particular amino acid) there are two possibilities. If both mixed distributions prefer similar types of amino acids (polar, for example), the relative entropy is small, but if the mixed distributions are of different types (e.g., one prefers polar amino acids, while the other prefers non-polar amino acids), the relative entropy will be larger, but still of moderate size. Because the Dirichlet mixture densities tend to generalize the posterior estimates toward the background distribution in the case where mixed amino acids are observed, the relative entropy between two different mixed distributions is larger than when the mixed distributions prefer similar amino acids, but is still not very large. When a conserved distribution disagrees with a mixed distribution, the relative entropy is larger than when two mixed distributions disagree, but not as large as when two conserved distributions disagree.

As Table 3 shows, when only a few amino acids are observed in a position, the posterior distribution produced using Dirichlet mixture priors tends toward the background distribution of the amino acids, with some additional probability for similar amino acids. However, as the number of observations increase, the posterior estimate using Dirichlet mixture priors converges on the frequencies in the data. As the number of observations increase, if the same amino acid is observed, the distribution will become increasingly conserved using Dirichlet mixture priors. (In contrast, substitution matrices do not produce conserved distributions, regardless of the number of amino acids observed in a position.) By controlling the total weight associated with the observed amino acids in a position, we can control the degree to which we generalize the posterior amino acid distributions produced using Dirichlet mixture priors. (Section 3.3 gives the method employed in this work to set the sequence weights in this work.)

In this way, conserved positions tend to dominate the measure of the total relative entropy (TRE) between two profiles. This favors joining two classes that have similar amino acids at conserved positions, and separating classes that have different (especially if conflicting) amino acids at conserved positions.

ensure that any redundancy in the data is compensated for by decreasing the total weight allotted the sequences. Otherwise, the posterior estimate of the amino acids at each position can become prematurely focused on the frequencies in the data, and unable to generalize to similar subfamilies when choosing which two subfamilies to join.

The approach I have taken in this work involves trying to estimate the number of independent observations in the data when setting the total weight, and then distributing the number of estimated independent observations among the sequences. For details on the method, see [Sj97].

4 Results on Bacteriorhodopsin and homologs

Bacteriorhodopsins are retinal-containing proteins found in Halobacterium halobium, a branch of archaebacteria. Like rhodopsin, bacteriorhodopsins are composed of 7 transmembrane helices, and mediate the transfer of ions and protons across the cell membrane [pro]. Bacteriorhodopsins are

related to other retinal proteins in archae, including halorhodopsins, archaerhodopsin precursors, and sensory rhodopsins.[7]

This data proved to be the most problematic for the evolutionary tree methods tested. With the exception of Bête, which produced monophyletic subtrees for all the functional classes identified in the literature (see Section 4.1), none of the other methods tested were able to correctly produce monophyletic subtrees for two of the functional subfamilies in the data (the archaerhodopsin precursors, and the bacteriorhodopsins themselves).

Two aspects of the data appear to be at the root of the problem. First, fairly long branch lengths (known to cause problems with phylogenetic tree estimation methods) are needed in a tree for this data, due to pairwise residue identities as low as 18% between species. Second, it appears that a deletion at the N and C termini of one of the proteins (bacr_halhm) threw off each of the other methods, and caused this protein to be incorrectly joined within the trees produced.

The alignment used in these experiments was obtained by reestimating an HSSP alignment [SS91] for the corresponding PDB structure 1brd, and identified homologs, using HMM methods.[8] This HMM was then used to search SwissProt for additional homologs, which were aligned to the model to produce the multiple alignment used here.

4.1 Functional subfamilies

Bête's decomposition of the data into subfamilies (using the method given in [Sj97]) corresponds directly with the three primary functional classes identified in the literature [JLY+95, GLRJ96, RBM+95, NPWR81] for this family:

1. Light-driven proton pumps

   This group include two subfamilies: bacteriorhodopsin (BACR proteins), and archaerhodopsin precursors (BAC1/BAC2).

2. Sensory rhodopsin II (BACT)

3. Light-driven chloride pumps, AKA halorhodopsins (BACH)

Two positions, 62 and 73, are involved in the function of all the proteins aligned. Seidel et al [RBM+95] noted that proton pumps (BACR, BAC1, BAC2) are characterized by aspartic acid at both positions; chloride pumps (BACH) have threonine at 62 followed by alanine at 73; sensors have aspartic acid at 62, followed by tyrosine or phenylalanine at 73. Other positions noted as involved in the function of these proteins include position 28-29, 127 and 161 noted as binding sites for the retinal chromophore in [NPWR81].[9]

[7] An interesting overview of the structure and function of these proteins, with useful links, is found on the World Wide Web at http://monera.ncl.ac.uk/energy/brd.html [War95].

[8] The software used in this work to refine the HSSP alignment and identify additional homologs comes from the Sequence Alignment and Modeling (SAM) suite maintained at UCSC [HK95]. See the UCSC website at http://www.cse.ucsc.edu/research/compbio/papers/sam_doc/sam_doc.html.

[9] Note that position numbering is different in these references, and it is necessary to refer to the sequence to obtain the mapping between alternative numberings. The numbering used here is with respect to the columns in the original HSSP aligment 1brd.hssp.

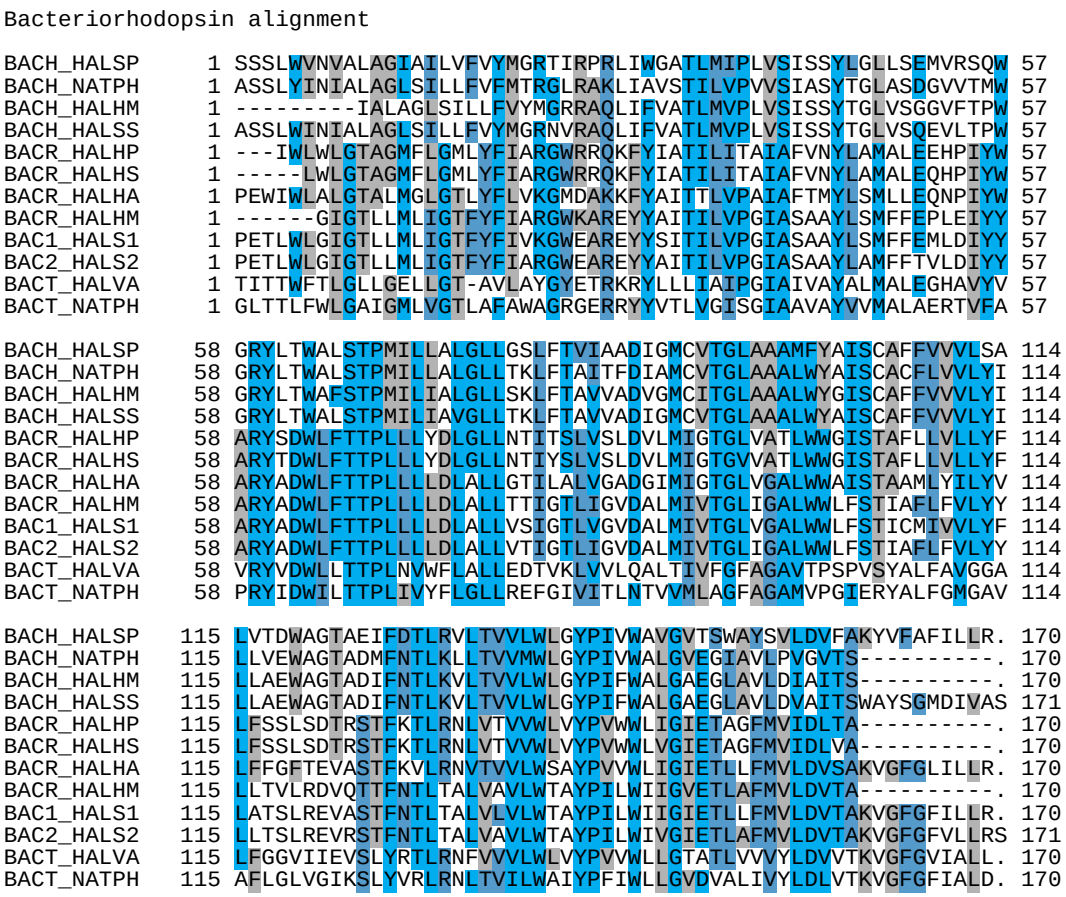

Figure 1: A permuted version of the bacteriorhodopsin alignment used as input to tree estimation methods (outgroup sequence not shown). This alignment is a black-and-white version of an alignment coloured by Belvu [Son97]: uncoloured positions reflect no conservation signal, dark grey positions reflect moderate conservation signal (typically showing a preference for a particular physico-chemical type, such as polar, rather than a specific amino acid), and light grey positions are those that reflect strong conservation of individual amino acids.
The sequences have been re-ordered, to show the decomposition into the three subfamilies produced by Bête: Bach, Bacr/Bac1/Bac2, and Bact. Bête placed the outgroup sequence into a separate subfamily, apart from the three subfamilies (see Figure 2).

Bacteriorhodopsin: Positions involved in protein function						
Subfamily	Positions in alignment					
	28	29	62	73	127	161
BACH	R,K,Q	L	T	A	D,N	W,K,-
BACR	R,K,Q	E,K	D	D	K,N	K,-
BAC1/BAC2	R	E	D	D	N	K
BACT	R	K,R	D	F	R,V	K

Table 5: Table of important positions in bacteriorhodopsin alignment. Positions 62 (Bacr D85) and 73 (Bacr D96) are noted as particularly crucial for protein function. Mutations at position 62 change light-driven proton pumps to light-driven chloride pumps [JLY$^+$95, GLRJ96]. Dashes indicate a deletion at a position.

Comparison of trees for bacteriorhodopsin alignment			
Tree Estimation Method	Monophyletic subtrees produced?		
	BAC1/BAC2	BACR	BAC1/BAC2/BACR
Bête	Yes	Yes	Yes
Parsimony–Tree 1	No	No	Yes
Parsimony–Tree 2	No	No	Yes
Maximum Likelihood	No	No	No
Neighbor-Joining	No	No	No
Star Decomposition	No	No	Yes
Puzzle	No	No	No

Table 6: Comparison of trees for bacteriorhodopsin alignment. BAC1/BAC2 are archaerhodopsin precursors, and, along with BACR, are light-driven proton pumps. All methods produce monophyletic subtrees for the halorhodopsins (BACH) and sensory rhodopsins (BACT).

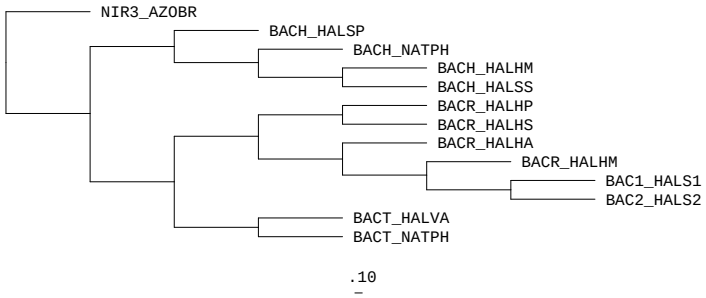

Figure 2: Bête tree for bacteriorhodopsin and homologs.
Note that all the functional subfamilies identified in the literature are monophyletic in this tree. A monophyletic subtree for the archaerhodopsin precursors (BAC1/BAC2), is joined with the a monophyletic subtree for the bacteriorhodopsins to form a monophyletic subtree for all light-driven proton pumps (BAC1/BAC2/BACR). In this tree, the light-driven proton pumps are joined more closely to the sensory rhodopsins (BACT) than to the halorhodopsins (BACH). Nir3_azobr is the outgroup sequence.
The subfamily decomposition produced by Bête separated the data into four groups: (1) the light-driven proton pumps (BAC1/BAC2/BACR), (2) the sensory rhodopsins (BACT), (3) the halorhodopsins (BACH) and (4) Nir3_azobr (the outgroup sequence).
Edge lengths drawn are all unit length, and do not correspond to the distance measurement computed to infer the tree.

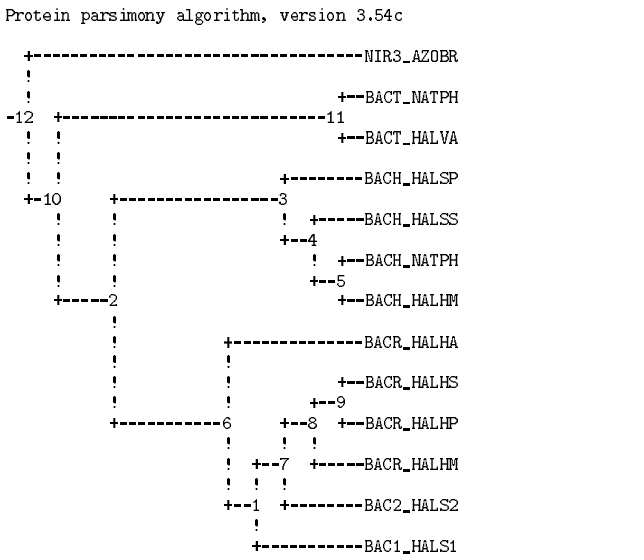

Figure 3: Parsimony tree for bacteriorhodopsin and homologs, computed using Felsenstein's *protpars* software in the PHYLIP suite This software is available through http://evolution.genetics.washington.edu/phylip.html [Fel96]. Note that no choice of a root produces monophyletic subtrees for both the archaerhodopsin precursors (BAC1/BAC2), and the bacteriorhodopsins (BACR). However, the light-driven proton pumps (BAC1/BAC2/BACR) are joined into a monophyletic subtree. In this tree, the light-driven proton pumps are joined more closely to the halorhodopsins (BACH) than to the sensory rhodopsins (BACT). Protpars also produced a second parsimony tree which was identical in the branching order among the light-driven proton pumps (BAC1/BAC2/BACR), but placed the light-driven proton pumps more closely to the sensory rhodopsins (BACT) than to the halorhodopsins (BACH). Nir3_azobr is the outgroup sequence.

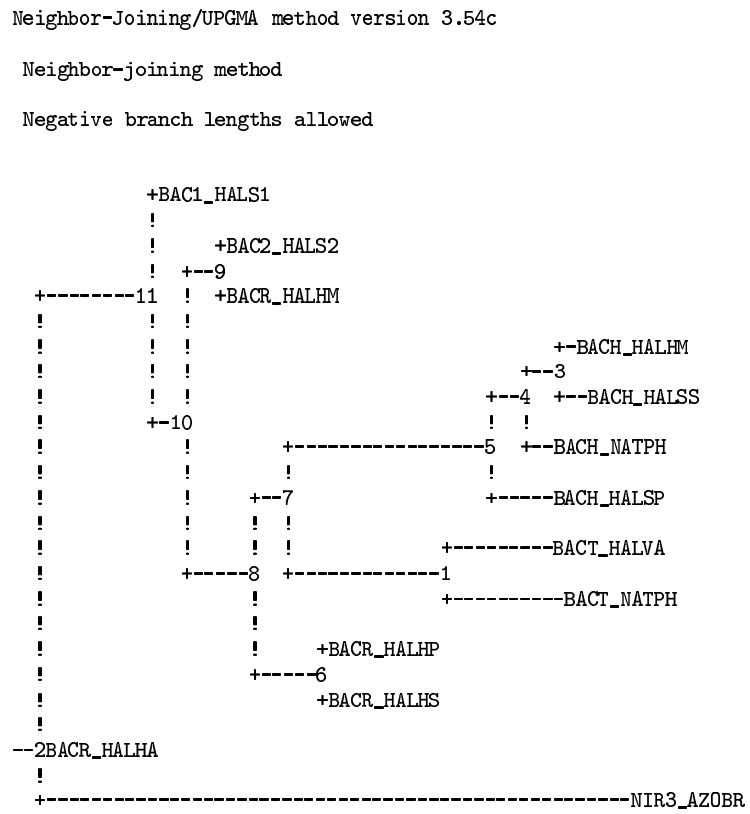

Figure 4: Neighbor-Joining tree for bacteriorhodopsin and homologs, estimated using *protdist* and *neighbor* software from the PHYLIP package. (See the caption to Figure 3 for additional information on this software.)
Note that no choice of a root produces monophyletic subtrees for the archaerhodopsin precursors (BAC1/BAC2), the bacteriorhodopsins (BACR), and the light-driven proton pumps (BAC1/BAC2/BACR) as a whole. Nir3_azobr is the outgroup sequence.

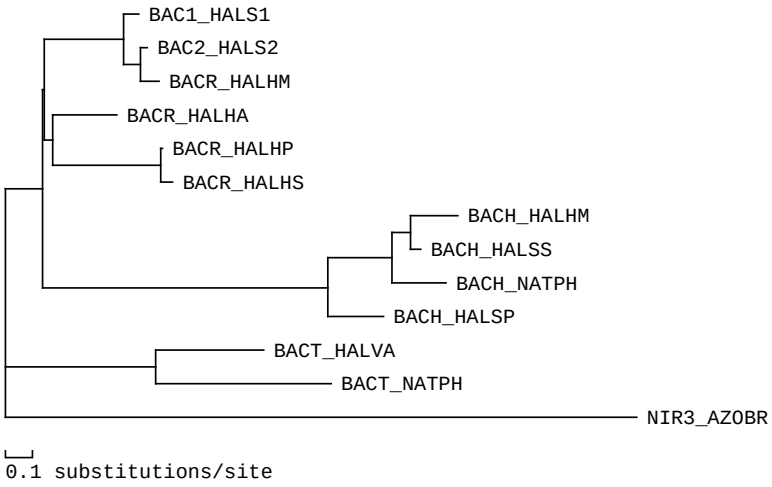

Figure 5: Star Decomposition tree for bacteriorhodopsin and homologs, estimated using the *protml* software of Adachi and Hasegawa's MOLPHY suite, with the "-s" option. This software is available by ftp from sunmh.ism.ac.jp.
Note that no choice of a root produces monophyletic subtrees for both the archaerhodopsin precursors (BAC1/BAC2), and the bacteriorhodopsins (BACR). However, the light-driven proton pumps (BAC1/BAC2/BACR) are joined into a monophyletic subtree. Nir3_azobr is the outgroup sequence.

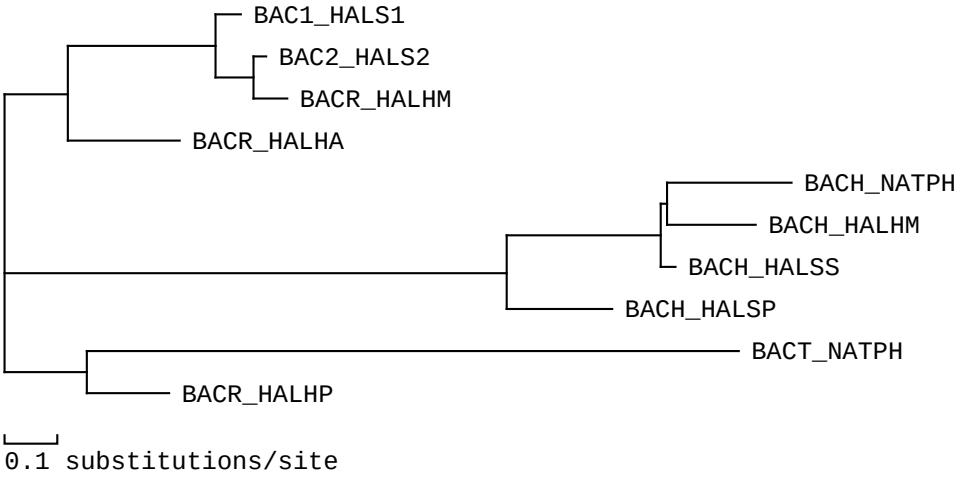

Figure 6: Maximum Likelihood tree for bacteriorhodopsin and homologs, estimated using the *protml* software of Adachi and Hasegawa's MOLPHY suite, with the "-e" option. Because of the computational complexity of ML tree estimation, a reduced set of taxa were used as the input to the program. Note that no choice of a root produces monophyletic subtrees for the archaerhodopsin precursors (BAC1/BAC2), the bacteriorhodopsins (BACR), and the light-driven proton pumps (BAC1/BAC2/BACR) as a whole.

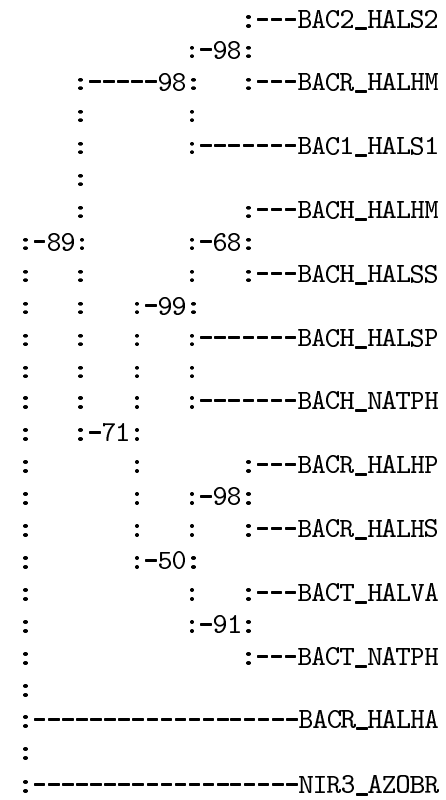

Figure 7: Puzzle tree for bacteriorhodopsin and homologs, estimated using the quartet method Puzzle from Von Haeseler and Strimmer. This software is available from http://www.zi.biologie.uni-muenchen.de/ strimmer/puzzle.html.
Note that no choice of a root produces monophyletic subtrees for the archaerhodopsin precursors (BAC1/BAC2), the bacteriorhodopsins (BACR), and the light-driven proton pumps (BAC1/BAC2/BACR) as a whole. Nir3_azobr is the outgroup sequence.

9 Literature Cited

[ACL89] Stephen F. Altschul, Raymond J. Carroll, and David J. Lipman. Weights for data related by a tree. *JMB*, 207:647–653, 1989.

[Alt91] Stephen F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *JMB*, 219:555–565, 1991.

[Bal97] Michael Balter. Morphologists learn to live with molecular upstarts. *Science*, 276:1032–1034, May 1997.

[BE95] Timothy L. Bailey and Charles Elkan. The value of prior knowledge in discovering motifs with MEME. In *ISMB-95*, pages 21–29, Menlo Park, CA, July 1995. AAAI/MIT Press.

[Ber85] J. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[BHK+93] M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, K. Sjölander, and D. Haussler. Using Dirichlet mixture priors to derive hidden Markov models for protein families. In L. Hunter, D. Searls, and J. Shavlik, editors, *ISMB-93*, pages 47–55, Menlo Park, CA, July 1993. AAAI/MIT Press.

[Bre88] S. Brenner. The molecular evolution of genes and proteins: a tale of two serines. *Nature*, 334:528–530, 1988.

[BS94] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley and Sons, first edition, 1994.

[Cla94] Jean-Michael Claverie. Some useful statistical properties of position-weight matrices. *Computers and Chemistry*, 18(3):287–294, 1994.

[CSMT95] Benham C, Kannan S, Paterson M, and Warnow T. Hen's teeth and whale's feet: generalized characters and their compatibility. *J Comput Biol*, 2:515–525, 1995.

[CSV95] G. Casari, C. Sander, and A. Valencia. A method to predict functional residues in proteins. *Structural Biology*, 2:171–178, 1995.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, first edition, 1991.

[Eis97] David Eisenberg. Into the black of night. *Nature Structural Biology*, 4:95–97, Feb 1997.

[Fel96] Joe Felsenstein. Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Methods in Enzymology*, 266:418–427, 1996.

[GBH90] David G. George, Winona C. Barker, and Lois T. Hunt. Mutation data matrix and its uses. *Methods in Enzymology*, 183:333–351, 1990.

[GLRJ96] Varo G, Brown LS, Needleman R, and Lanyi JK. Proton transport by halorhodopsin. *Biochemistry*, 35:6604–6611, May 1996.

[HH92] Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *PNAS*, 89:10915–10919, November 1992.

[HH93] Steven Henikoff and Jorja G. Henikoff. Performance evaluation of amino acid substitution matrices. *Proteins: Structure, Function, and Genetics*, 17:49–61, 1993.

[HH94] Steven Henikoff and Jorja G. Henikoff. Position-based sequence weights. *JMB*, 243(4):574–578, November 1994.

[HK95] R. Hughey and A. Krogh. SAM: Sequence alignment and modeling software system. Technical Report UCSC-CRL-95-7, University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064, 1995.

[HK96] Richard Hughey and Anders Krogh. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, 12(2):95–107, 1996.

[HMM96] David M. Hillis, Craig Moritz, and Barbara K. Mable. *Molecular systematics*. Sinauer Associates, second edition, 1996.

[JLY+95] Sasaki J, Brown LS, Chon YS, Kandori H, Maeda A, Needleman R, and Lanyi JK. Conversion of bacteriorhodopsin into a chloride ion pump. *Science*, 269:73–75, Jul 1995.

[JTT92] David T. Jones, William R. Taylor, and Janet M. Thornton. The rapid generation of mutation data matrices from protein sequences. *CABIOS*, 8(3):275–282, 1992.

[Kar95] Kevin Karplus. Regularizers for estimating distributions of amino acids from small samples. In *ISMB-95*, Menlo Park, CA, July 1995. AAAI/MIT Press.

[KF94] M.J. Kuhner and Joe Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol Biol Evol*, 11:449–468, 1994.

[KKB+97a] Kevin Karplus, Kimmen Sjölander, Christian Barrett, Melissa Cline, David Haussler, Richard Hughey, Liisa Holm, and Chris Sander. Predicting protein structure using hidden Markov models. *Proteins: Structure, Function, and Genetics*, 1997.

[KKB+97b] Kevin Karplus, Kimmen Sjölander, Christian Barrett, Melissa Cline, David Haussler, Richard Hughey, Liisa Holm, and Chris Sander. Predicting protein structure using hidden Markov models, the CASP2 contest. Technical Report UCSC-CRL-97-13, University of California, Santa Cruz, Computer Science, UC Santa Cruz, CA 95064, 1997.

[MHN91] Hasegawa M, Kishino H, and Saitou N. On the maximum likelihood method in molecular phylogenetics. *J. Mol. Evol.*, 32:443–445, May 1991.

[NPWR81] Katre NV, Wolber PK, Stoeckenius W, and Stroud RM. Attachment site(s) of retinal in bacteriorhodopsin. *PNAS*, 78:4068–4072, Jul 1981.

[OHF96] Lichtarge O, Bourne H.R., and Cohen F.E. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.*., 257:342–358, Mar 1996.

[pro] URL:http://expasy.hcuge.ch/cgi-bin/get-prodoc-entry?PDOC00291.

[RBM+95] Seidel R, Scharf B, Gautel M, Kleine K, Oesterhelt D, and Engelhard M. The primary structure of sensory rhodopsin II: a member of an additional retinal protein subgroup is coexpressed with its transducer, the halobacterial transducer of rhodopsin II. *PNAS*, 92:3036–3040, Mar 1995.

[RJ94] Michael A. Rodionov and Mark S. Johnson. Residue-residue contact substitution probabilities derived from aligned three-dimensional structures and the identification of common folds. *Protein Sci.*, 3:2366–2377, 1994.

[RW97] Kenneth Rice and Tandy Warnow. Parsimony is hard to beat. *submitted*, May 1997.

[SA90] P. Sibbald and P. Argos. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *JMB*, 216:813–818, 1990.

[SD89] T. J. Santner and D. E. Duffy. *The Statistical Analysis of Discrete Data*. Springer Verlag, New York, 1989.

[Sj97] K. Sjölander. A Bayesian-Information theoretic method for evolutionary inference in proteins. Technical Report UCSC-CRL-97-14, University of California at Santa Cruz, Computer and Information Sciences Dept., Santa Cruz, CA 95064, 1997.

[SKB+96] K. Sjölander, K. Karplus, M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *CABIOS*, 12(4):327–345, 1996.

[Son97] Erik Sonnhammer. http://www.sanger.ac.uk/ esr/belvu.html. World Wide Web, 1997.

[SS91] C. Sander and R. Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9(1):56–68, 1991.

[Str95] Lubert Stryer. *Biochemistry*. W.H.Freeman and Company, fourth edition, 1995.

[TAK94] Roman L. Tatusov, Stephen F. Altschul, and Eugen V. Koonin. Detection of conserved segments in proteins: Iterative scanning of sequence databases with alignment blocks. *PNAS*, 91:12091–12095, December 1994.

[THG94a] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. *NAR*, 22(22):4673–4680, 1994.

[THG94b] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *CABIOS*, 10(1):19–29, 1994.

[VS93] Martin Vingron and Peter R. Sibbald. Weighting in sequence space: a comparison of methods in terms of generalized sequences. *PNAS*, 90:8777–8781, October 1993.

[War95] Alan Ward. http://monera.ncl.ac.uk/energy/brd.html. World Wide Web, October 1995.

[Yan94] Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J. Mol. Evol.*, 39:306–314, Sep 1994.

[Yan96] Z. Yang. Phylogenetic analysis using parsimony and likelihood methods. *J. Mol. Evol.*, 42:294–307, Feb 1996.

[YNM94] Tateno Y, Takezaki N, and Nei M. Relative efficiencies of the maximum-likelihood, neighbor-joining, and maximum-parsimony methods when substitution rate varies with site. *Mol Biol Evol*, 11:449–468, 1994.

Acknowledgments

This work was supported by a National Science Foundation Graduate Research Fellowship, and by a fellowship from the Program in Mathematics and Molecular Biology. Inspiration for this work comes from various sources, especially from the work of Chris Sander, Georg Casari and Alfonso Valencia at EMBL-Heidelberg, and conversations with David Haussler, Graeme Mitchison, Sean Eddy, Tandy Warnow and Gary Churchill.