Maximum Entropy Weighting of Aligned Sequences of Proteins or DNA

Anders Krogh
Nordita
Blegdamsvej 17, 2100 Copenhagen
Denmark
email: krogh@nordita.dk

Graeme Mitchison
Laboratory of Molecular Biology
Hills Road, Cambridge CB2 2QH
England
email: gjm@mrc-lmb.cam.ac.uk

Abstract

In a family of proteins or other biological sequences like DNA the various subfamilies are often very unevenly represented. For this reason a scheme for assigning weights to each sequence can greatly improve performance at tasks such as database searching with profiles or other consensus models based on multiple alignments. A new weighting scheme for this type of database search is proposed. In a statistical description of the searching problem it is derived from the maximum entropy principle. It can be proved that, in a certain sense, it corrects for uneven representation. It is shown that finding the maximum entropy weights is an easy optimization problem for which standard techniques are applicable.

Introduction

Consensus models made from multiple sequence alignments have proved very useful for searching databases (Taylor 1986; Gribskov, McLachlan, & Eisenberg 1987; Barton 1990; Bairoch 1993; Henikoff & Henikoff 1994; Krogh *et al.* 1994). A common problem, however, is that some groups of sequences dominate the multiple alignment and outweigh other groups. For instance, an alignment of a random set of known globins would contain mostly vertebrate alpha and beta chains of which several hundred are known, whereas other families of globins, like leghemoglobins, would have many fewer representatives. Thus a search based on such a globin alignment would be more likely to pick out vertebrate alpha and beta chains than the less common globins. For this reason a weighting of the sequences that compensates for the differences in representation may be very important. A method for weighting sequences can be useful in other situations too, in the prediction of protein secondary structure from multiple alignments (Levin *et al.* 1993) for instance, or for use in the actual alignment procedure (Thompson, Higgins, & Gibson 1994a).

Several methods exist for weighting sequences in alignments. The methods in (Felsenstein 1973; Altschul, Carroll, & Lipman 1989) can be used for sequences that are related by a known phylogenetic tree; the distances between nodes in the tree are used for calculating the weights. Some related methods, *e.g.* those in (Gerstein, Sonnhammer, & Chothia 1994) and (Thompson, Higgins, & Gibson 1994b), use an automatic method for tree-building. A weighting method by Vingron and Argos (1989) is based on the simple rationale that sequences that are close to a lot of other sequences should have a small weight. If d_{nm} is the distance (which can be any measure) between sequence n and m, they assign the weight $w_n = \sum_m d_{nm}$ to sequence n. However, this gives the 'wrong' weights for some simple examples; see Table 1. Another method by Sibbald and Argos (1990) based on Voronoi diagrams does not have these shortcomings, and its main drawback is that it is quite computationally demanding for large alignments. A computationally very simple method has recently been suggested by Henikoff and Henikoff (1994). If k residues are present in a column of the alignment of which m are the same as the one in sequence n, then $1/mk$ is added to the weight of sequence n. After going through all the columns the weights are normalized. This scheme will also give the 'correct' set of weights for the example in Table 1.

Sequence	V.&A.	VW	ME
AAAAAAAA	0.1875	0.1667	0.1667
AAAAAAAA	0.1875	0.1667	0.1667
CCCCCCCC	0.1875	0.1667	0.1667
CCCCCCCC	0.1875	0.1667	0.1667
GGGGGGGG	0.25	0.3333	0.3333

Table 1: A toy example of a multiple alignment. The first column shows the weights assigned by the method proposed in (Vingron & Argos 1989). It is seen that the last sequence obtains a weight only slightly larger than the other four sequences. It is obvious that the last sequence should be assigned twice as large a weight as the first four, so that the three different sequences are given the same weights. This is exactly what the Voronoi weights (VW) shown in the second column do. The last column shows that the maximum entropy weights are equal to the VW. We have actually cheated a little here, because ME weights only ensure that the weights of the A sequences add up to 1/3, and not that they are individually equal to 1/6 (similarly for the C sequences).

In (Eddy, Mitchison, & Durbin 1994) a strategy is developed for estimating a hidden Markov model (HMM) so as to optimize the discriminatory power of the model. Although this aspect is not emphasized in the paper, the methods called maximum discrimination (MD) and maximin can be interpreted as weighting schemes, and they turn out to be closely related to the method derived in the present paper.

All this work on weighting has shown that weighting can improve performance of tasks such as database searches quite significantly. See also (Vingron & Sibbald 1993).

In this paper a new weighting scheme is suggested. It is specifically designed for use with profiles and HMMs made for searching databases. If a certain group of sequences dominates a multiple alignment, these sequences and their close relatives will be very close to the model, *i.e.*, have a high likelihood, and thus have a much larger chance of being found in a search, than sequences with smaller representation. Our basic idea has been to design a weighting scheme that gives poorly represented sequences a larger likelihood. This idea translates into a maximum entropy principle, and the weighting scheme will be referred to as maximum entropy (ME) weighting.

One of the main advantages of our weighting scheme is that it is based on theory compatible with the basic assumptions behind profile search and HMM search, whereas most other weighting schemes are based on intuitive ideas. It can be proved that our scheme corrects in a certain sense for the uneven representation.

It is important to realize that there is no objectively 'correct' way to weight sequences. Any weighting scheme builds on some assumptions about the structure of the space of sequences and on some goals for the weighting. Often a weighting scheme has some implicit assumptions about the probability distribution over the space of sequences. In this work these assumptions are the same as those which underpin profile and HMM search.

Profiles of block alignments

For simplicity we will first discuss the case of block alignments, but all the results will carry over to the more general case discussed later. We define a block alignment to be one where gaps can occur, but these gaps are treated like additional characters, *i.e.*, no penalty is used for opening a gap.

Assume the multiple alignment consists of sequences s^n, $n = 1, \ldots, N$. Each sequence consists of characters s_i^n, $i = 1, \ldots, L$, some of which may be gap characters ('-'). Define

$$m_{ij}^n = \begin{cases} 1 \text{ if sequence } n \text{ has character } j \\ \qquad \text{at position } i,\ s_i^n = j \\ 0 \text{ otherwise.} \end{cases} \quad (1)$$

Then the frequency of character j in column i is

$$p_i(j) = p_{ij} = \frac{\sum_n m_{ij}^n}{\sum_n \sum_j m_{ij}^n} = \frac{1}{N} \sum_n m_{ij}^n. \quad (2)$$

These frequencies constitute the profile p.

In profile search a distance is defined between a profile and a sequence. Here this distance is defined in terms of the probability $P(s|p)$ of sequence s given the profile p, which we will usually call just $P(s)$. This is defined as

$$P(s) = \prod_i p_i(s_i). \quad (3)$$

The distance between the profile and the sequence could then be taken to be $-\log P(s)$, for example. It is assumed that the sequence is already aligned to the profile by some method, so s is a sequence of length L with gap characters in appropriate places. Here we do not have to specify which method is used for alignment, but to be consistent one would usually choose the alignment that maximizes the probability of the sequence given the profile, which can be found by standard dynamic programming techniques.

Maximum entropy weights

What does it really mean to say that a profile is skewed towards certain sequences? One way to express it is that some sequences in the alignment have a larger probability than others. From (2) it is obvious that subfamilies with many representatives will dominate the p_{ij}, and therefore have high probabilities whereas subfamilies with a low representation will have small probabilities. The goal of a weighting scheme is to correct for this uneven representation by assigning weights w_n to all sequences. Thus the profile is changed from the raw frequencies to weighted frequencies, *i.e.*, equation (2) changes to

$$p_i(j) = p_{ij} = \frac{\sum_n w_n m_{ij}^n}{\sum_n w_n} = \frac{1}{W} \sum_n w_n m_{ij}^n, \quad (4)$$

where $W = \sum_n w_n$. Only the relative size of the weights matters, and one can normalize the weights to sum to one, but it turns out to be convenient to use the general form in the following. It will be assumed throughout that the weights are non-negative.

The entropy of a profile

In information theory the entropy of a discrete probability distribution $P(z)$ is given by $-\sum_z P(z) \log P(z)$, see for instance (Cover & Thomas 1991). For the probability distribution over sequences (3) the entropy is

$$S(w) = -\sum_s P(s) \log P(s), \quad (5)$$

where the sum extends over all possible sequences with length L. Since the profile depends on the weights, so will the entropy, which is indicated explicitly. Using

the definition (3) of $P(s)$ this can be rewritten as an explicit sum over all combinations of L characters

$$\begin{aligned} S(w) &= -\sum_{j_1}\sum_{j_2}\cdots\sum_{j_L}\prod_k p_{kj_k}\sum_i \log p_{ij_i} \\ &= -\sum_{ij} p_{ij}\log p_{ij}. \end{aligned} \quad (6)$$

This is the sum of the entropies of each column in the profile, $S = \sum_i S_i$, where $S_i = -\sum_j p_{ij}\log p_{ij}$.

Among all probability distributions the uniform distribution has the largest entropy. Thus by maximizing the entropy, the probability distributions p_{ij} will be as close to uniform as possible given the sequences in the alignment; this was exactly the goal formulated above. Thus the maximum entropy weights w_n^{ME} are simply defined as

$$w^{ME} = \underset{w}{\operatorname{argmax}} S(w). \quad (7)$$

Maximum entropy weights will usually make most of the probabilities $P(s^n)$ equal to the minimum probability, a claim that will be formalized later. One may argue that this is not a desirable situation if all the probabilities are very small. Indeed there are other criteria one could use for optimizing the weights, but as we shall see later, some of the obvious criteria turn out to be equivalent or closely related to the ME criterion.

Maximum discrimination and maximin

In (Eddy, Mitchison, & Durbin 1994) hidden Markov models for maximum discrimination are discussed, and two optimization criteria are put forward. It is argued that a model should be optimized in such a way that the sequence with the minimum probability has the highest possible probability; *i.e.*, one would like to find a set of weights such that $\min_n P(s^n)$ is as large as possible,

$$w^{mm} = \underset{w}{\operatorname{argmax}} \min_n P(s^n). \quad (8)$$

This criterion is called *maximin*.

The other criterion is called maximum discrimination (MD), and this is in fact the primary criterion in (Eddy, Mitchison, & Durbin 1994). It is argued that models should be optimized to discriminate between sequences in the given family and sequences in other families. This is done by comparing the performance of the model to a very simple model, in which it is assumed that the characters in the sequences are randomly drawn from a fixed distribution, for instance the overall distribution of amino acids in a big database. If the probability of a sequence s according to the simple model is $Q(s)$, it is shown that one should minimize the quantity

$$D = -\sum_n \log \frac{P(s^n)}{P(s^n)+Q(s^n)} \quad (9)$$

to develop a model that optimally separates the sequences in the alignment from the simple model. In terms of weights, the maximum discrimination criterion means

$$w^{MD} = \underset{w}{\operatorname{argmin}} D. \quad (10)$$

If the model ends up being a good one, $Q(s)/P(s)$ will generally be extremely small, so at the minimum the first order expansion of the logarithm will be a very good approximation,

$$D \simeq \sum_n \frac{Q(s^n)}{P(s^n)}. \quad (11)$$

This sum will be dominated by the contribution from the sequence with the largest term, *i.e.*, the sequence with the lowest log-odds score, $\log \frac{P(s^n)}{Q(s^n)}$. Therefore one can approximate the maximum discrimination weights by a maximin solution,

$$w^{MD} \simeq \underset{w}{\operatorname{argmax}} \min_n \frac{P(s^n)}{Q(s^n)} \quad (12)$$

Here both MD and maximin are interpreted as weighting schemes, although the optimization in (Eddy, Mitchison, & Durbin 1994) is not explicitly over sequence weights. Surprisingly, simulations suggest that maximin and ME are equivalent; furthermore, a relative entropy version of ME appears to be equivalent to maximin as defined by (12), and hence approximately the same as MD. We indicate later why these results might hold true.

Properties of ME weights

The main property of ME weights is that all the sequence probabilities are either the same and equal to the minimum, $P(s^n) = P_{min}$, or they are larger and have zero weight, $w^n = 0$. More formally, if $P_{min} = \min_n P(s^n)$, then

$$w^* \text{ is a ME set of weights} \quad (13)$$

$$\Updownarrow$$

$$w_n^* = 0 \text{ for all sequences with } P(s^n) > P_{min}. \quad (14)$$

This will be proved below.

Another nice property of the entropy is that it is a concave function of the weights if the weights are normalized. When the weights are not normalized, the entropy is flat in directions where the relative sizes of the weights are constant. The proof of this 'semi-concavity' goes as follows.

The entropy $-\sum_k p_k \log p_k$ is a concave function in p_k. A concave function of a linear function is concave, and so is a sum of concave functions. The probabilities p_{ij} are linear functions of the weights, see (4), for a fixed sum W of the weights. Thus the total entropy (6) is a concave function of the weights for fixed W. If W is not fixed the entropy is concave except in directions $w_n = \text{const} \cdot a_n$, where a_n are arbitrary positive numbers.

Proof that (13) ⇒ (14)

First, the derivative of $S(w)$ with respect to w_n is needed. We start by finding the derivative of p_{ij} (4):

$$\frac{\partial p_{ij}}{\partial w_n} = W^{-1}(m_{ij}^n - p_{ij}). \tag{15}$$

Differentiation of S (6) then gives

$$\begin{aligned}\frac{\partial S}{\partial w_n} &= -\sum_{ij}(\log p_{ij} - 1)\frac{\partial p_{ij}}{\partial w_n} \\ &= -W^{-1}\left(\sum_{ij} m_{ij}^n \log p_{ij} - \sum_{ij} p_{ij}\log p_{ij}\right)\end{aligned} \tag{16}$$

The last term is the entropy (6), and from (3) it follows that

$$P(s^n) = \prod_{ij} p_{ij}^{m_{ij}^n}, \tag{17}$$

so the derivative of the entropy becomes

$$\frac{\partial S}{\partial w_n} = -W^{-1}(\log P(s^n) + S). \tag{18}$$

At the maximum of the entropy either the derivative (18) is zero, or the weight is zero and the derivative is non-positive, *i.e.*

$$\begin{gathered}\text{either} \\ -W^{-1}(\log P(s^n) + S(w^*)) = 0 \\ \text{or} \\ -W^{-1}(\log P(s^n) + S(w^*)) \le 0 \quad \text{and } w_n^* = 0,\end{gathered} \tag{19}$$

from which (14) easily follows.

Proof that (14) ⇒ (13)

Rewrite the entropy (6) using (4) and (17)

$$\begin{aligned}S &= -\frac{1}{W}\sum_{ij}\sum_n w_n m_{ij}^n \log p_{ij} \\ &= -\frac{1}{W}\sum_n w_n \log \prod_{ij} p_{ij}^{m_{ij}^n} \\ &= -\frac{1}{W}\sum_n w_n \log P(s^n).\end{aligned} \tag{20}$$

From this and (14) it follows that $S(w^*) = -\log P_{min}$. Therefore, the derivative (18) of the entropy is zero if $w_n^* > 0$, and negative for n with $P(s^n) > P_{min}$ and $w_n^* = 0$. From the 'semi-concavity' of S (discussed above) it then follows that w^* is a maximum of S. This completes the proof.

As a little corollary, we can mention that $P_{min} = \exp(-S(w^*))$, which is evident from (19).

Relation to maximin and MD

In all the numerical tests we have done, we obtained essentially identical weights from ME and maximin (*e.g.* Table 2). The following argument suggests why this might be:

It is easy to imagine that, for maximin, $P(s^n) = P_{min}$ will hold for many of the sequences. If one can show that $w_n^{mm} = 0$ for all sequences with $P(s^n) > P_{min}$, then ME and maximin would be equivalent by (13) and (14). Assume that it is *not* the case that $w_n = 0$ for all sequences with $P(s^n) > P_{min}$, and that there are $K < N$ sequences with $P(s^n) = P_{min}$. If the K gradient vectors in w-space of the corresponding probabilities $P(s^n)$ are linearly independent, there must be a direction in w-space in which all the K probabilities *increase*, and thus we cannot be at a maximin solution. More precisely, we should restrict to the subspace R of w-space defined by the K sequences with $P(s^n) = P_{min}$ and those sequences for which $P(s^n) > P_{min}$ and $w_n > 0$. By assumption, there is at least one of the latter, and so R has dimension $\ge K+1$. Since the weights are normalized in the definition (4) of $P(s^n)$, one degree of freedom is lost, and we can further restrict to the affine subspace where $\sum w_n = 1$. We therefore require that the K gradient vectors be independent in a subspace of dimension $\ge K$. We have not been able to prove this independence yet except by explicit calculation in the cases $N \le 3$, but we conjecture that a proof for all N will be possible.

Turning now to the relation between ME and MD, the relative entropy between P and the simple model Q in (9) is defined as

$$\begin{aligned}R(w) &= -\sum_n P(s^n)\log\frac{P(s^n)}{Q(s^n)} \\ &= -\sum_{ij} p_{ij}\log\frac{p_{ij}}{q_j},\end{aligned} \tag{21}$$

where we have assumed that the simple model just assigns the probability $\prod_i q(s_i)$ to the sequence. Note that the sign of this relative entropy is unconventional. Using this, one can show that what holds for $P(s^n)$ in ME holds for the odds ratio $O(s^n) = P(s^n)/Q(s^n)$ with *maximum relative entropy* (MRE). That is, either $O(s^n) = O_{min}$ or $O(s^n) > O_{min}$ and $w_n = 0$, where $O_{min} = \min_n O(s^n)$. Therefore, to the extent that (12) is a good approximation to MD, we expect MRE to be a good approximation to MD. This is supported by numerical simulations (see Table 2).

If the distribution q is uniform $R(w) = S(w) + \text{const}$, so ME and MRE are equivalent. Therefore ME is also an MRE method, and MRE should perhaps be considered the basic method, even if one sometimes chooses the uniform model for comparison.

Use of ME weights

One of the satisfying features of ME weights is that the entropy is 'semi-concave' as described earlier. This means that finding the weights is in principle a very simple optimization problem.

The straightforward way to optimize the entropy is to do gradient ascent using (18), which has worked well

Swissprot identifier	Weights					Log odds					
	GSC	H&H	MRE	MM	MD	UW	GSC	H&H	MRE	MM	MD
UROK_HUMAN	0.0365	0.0324	0.0	0.0001	0.0035	61.0	56.8	55.0	50.1	50.1	50.9
UROK_RAT	0.0365	0.0358	0.0	0.0001	0.0065	58.3	54.1	52.8	49.9	49.9	50.2
HGF_RAT	0.0445	0.0465	0.0206	0.0206	0.0318	53.0	50.4	50.1	47.8	47.8	48.7
UROK_PIG	0.0568	0.0417	0.0325	0.0325	0.0358	57.2	54.0	51.7	47.8	47.8	48.5
THRB_BOVIN	0.0516	0.0469	0.0336	0.0336	0.0366	53.7	51.7	50.3	47.8	47.8	48.5
UROT_DESRO	0.0608	0.0513	0.0337	0.0337	0.0376	54.4	52.7	51.2	47.8	47.8	48.5
UROT_RAT	0.0391	0.0449	0.0372	0.0372	0.0399	55.2	51.5	51.1	47.8	47.8	48.4
THRB_HUMAN	0.0516	0.0464	0.0390	0.0390	0.0414	53.3	51.1	49.7	47.8	47.8	48.4
UROK_PAPCY	0.0365	0.0360	0.0404	0.0403	0.0397	58.5	54.0	52.3	47.8	47.8	48.4
UROT_HUMAN	0.0742	0.0557	0.0405	0.0405	0.0424	53.2	53.8	51.3	47.8	47.8	48.4
UROT_MOUSE	0.0391	0.0453	0.0487	0.0487	0.0458	54.8	51.2	50.8	47.8	47.8	48.3
THRB_RAT	0.0516	0.0513	0.0502	0.0502	0.0525	50.7	49.2	48.2	47.8	47.8	48.2
FA12_HUMAN	0.0742	0.0584	0.0624	0.0624	0.0616	50.3	51.5	49.0	47.8	47.8	48.0
THRB_MOUSE	0.0516	0.0515	0.0715	0.0715	0.0672	49.9	48.5	47.4	47.8	47.8	47.9
HGF_HUMAN	0.0445	0.0534	0.0777	0.0777	0.0680	50.8	48.2	48.3	47.8	47.8	47.9
UROK_MOUSE	0.0365	0.0430	0.0784	0.0783	0.0698	54.9	50.4	49.4	47.8	47.8	47.9
PLMN_MOUSE	0.0848	0.0805	0.0944	0.0944	0.0912	45.3	48.3	47.0	47.8	47.8	47.6
UROK_CHICK	0.1295	0.1789	0.2393	0.2393	0.2284	13.6	33.5	41.0	47.8	47.8	46.7

Table 2: Weights and log odds scores for the kringle domains in Table 3. The relative entropy versions of ME (MRE) and maximin (MM) were used, as defined by (21) and (12), respectively. The MD weights were obtained by maximizing D in (9) (Eddy, Mitchison, & Durbin 1994), where the model Q was defined as in (21). The probabilities q_j were the frequencies of occurrence of amino acids in the Blocks database. The log odds scores for sequence n is defined to be $\log[P(s^n)/Q(s^n)]$. The MM weights were computed by incrementally increasing the weight of the sequence with the current minimal score; this seemed as effective as the incremental update of probability parameters used in (Eddy, Mitchison, & Durbin 1994). We also computed the log odds for uniform weights (UW) as well as the weights (labeled GSC) by the method in (Gerstein, Sonnhammer, & Chothia 1994) and the weights (labeled H&H) by the method in (Henikoff & Henikoff 1994).

in our numerical test. In gradient ascent, the weights are changed iteratively in the direction of the gradient, *i.e.*, $w_n^{new} = w_n^{old} + \epsilon \frac{\partial S}{\partial w_n}$, where ϵ is some small constant. It is also possible to use more sophisticated techniques like conjugate gradient ascent (Press *et al.* 1986).

In Table 2, weights and log odds scores are derived for sequences consisting of a conserved region in a set of kringle domain proteins shown in Table 3. Note the almost identical weights and log odds scores of MRE and MM, and the similarity between MD and MRE. Here two sequences, UROK_HUMAN and UROK_RAT, receive zero weights by MRE and maximin; as expected, they have higher log odds scores than the rest. UROK_HUMAN differs from UROK_PAPCY by only two residues, and UROK_RAT differs from UROK_MOUSE by only two residues. The residues that differ are found in other sequences, and therefore UROK_HUMAN and UROK_RAT have a high probability, even though they do not contribute to the profile.

In the table weights and log odds scores are shown for two other weighting schemes for comparison. In the columns labeled GSC the results of the weighting scheme by Gerstein, Sonnhammer, & Chothia (1994) are shown and in the columns labeled H&H the results of the scheme by Henikoff & Henikoff (1994) are shown. For this example, the GSC weights are the least radical, *i.e.*, closest to uniform, the H&H are further from uniform, but not as radical as the MRE and MM weighting schemes.

General alignments and hidden Markov models

Because of space limitations, we cannot describe details of HMMs, so this section probably requires basic knowledge about this field. For an introduction, see (Rabiner 1989) and (Krogh *et al.* 1994).

It is not completely evident how to extend our treatment of block alignments to the more general case of alignments with gaps, because the entropy can be defined in several ways. It turns out that using the 'sequence-space entropy' defined in (5) makes the theory intractable, although good approximations may be found. Instead one should use the expression (20) as the starting point.

Here we are considering only HMMs made from fixed alignments as in (Eddy, Mitchison, & Durbin 1994). The distributions over characters are estimated in the same manner as before, except that one has to normalize differently from before, because the columns can have different numbers of characters. Therefore (4) changes to

$$p_{ij} = \frac{\sum_n w_n m_{ij}^n}{\sum_{j'} \sum_n w_n m_{ij'}^n}, \qquad (22)$$

UROK_HUMAN	CYEGNGHFYRGKASTDTMGRPCLPWNS
UROK_RAT	CYHGNGQSYRGKANTDTKGRPCLAWNS
HGF_RAT	CIIGKGGSYKGTVSITKSGIKCQPWNS
UROK_PIG	CFEGNGHSYRGKANTNTGGRPCLPWNS
THRB_BOVIN	CAEGVGMNYRGNVSVTRSGIECQLWRS
UROT_DESRO	CYKDQGVTYRGTWSTSESGAQCINWNS
UROT_RAT	CFEGQGITYRGTWSTAENGAECINWNS
THRB_HUMAN	CAEGLGTNYRGHVNITRSGIECQLWRS
UROK_PAPCY	CYEGNGHFYRGKASTDTMGRSCLAWNS
UROT_HUMAN	CYFGNGSAYRGTHSLTESGASCLPWNS
UROT_MOUSE	CFEEQGITYRGTWSTAESGAECINWNS
THRB_RAT	CAMDLGLNYHGNVSVTHTGIECQLWRS
FA12_HUMAN	CYDGRGLSYRGLARTTLSGAPCQPWAS
THRB_MOUSE	CAMDLGVNYLGTVNVTHTGIQCQLWRS
HGF_HUMAN	CIIGKGRSYKGTVSITKSGIKCQPWSS
UROK_MOUSE	CYHGNGDSYRGKANTDTKGRPCLAWNA
PLMN_MOUSE	CYQSDGQSYRGTSSTTITGKKCQSWAA
UROK_CHICK	TNSICYSGNGEDYRGMAEDPGCLYWDH

Table 3: The conserved region in a set of kringle domain proteins which constitutes block BL00021A in Version 7.01 (1993) of the Blocks database, (Henikoff & Henikoff 1991). The Swissprot identifiers are shown to the left.

where the gaps are *not* counted as letters. In an HMM the transition probabilities, that determine the gap penalties, are also estimated from the alignment. If the number of times the transition from state i to j is used in the alignment is called M_{ij}, the probability of this transition is set to

$$t_{ij} = \frac{\sum_n w_n M_{ij}^n}{\sum_{j'} \sum_n w_n M_{ij'}^n}, \qquad (23)$$

which is of exactly the same form as (22). The probability of a sequence $P(s)$ is now defined as a product of probabilities as in (3), except that the new type of probabilities P_{ij} will appear in the product. Having defined $P(s)$, we can now use the entropy definition (20),

$$S(w) = -\frac{1}{W} \sum_n w_n \log P(s^n). \qquad (24)$$

Inserting the expression for $P(s)$, one can show that the entropy can be written as

$$S = S_c + S_t, \qquad (25)$$

where

$$S_c = \sum_i \frac{\sum_{j'} \sum_n w_n m_{ij'}}{\sum_n w_n} \sum_j p_{ij} \log p_{ij}. \qquad (26)$$

The transition entropy S_t has exactly the same form, except that m and p are replaced by M and t, respectively. The entropy is still a sum over all the local entropies, but they are now weighted according to their 'usage'.

Since the formalism really changes very little, it is no surprise that all the results derived earlier carry through to HMMs.

In an HMM, a type of regularization derived from a Dirichlet prior can be used (Krogh *et al.* 1994; Eddy, Mitchison, & Durbin 1994), which is particularly useful if the number of sequences in the alignment is small. It amounts to adding regularization parameters in the equations for the probability parameters, so (22) changes to

$$p_{ij} = \frac{\sum_n w_n (m_{ij}^n + \frac{1}{N}\alpha_{ij})}{\sum_{j'} \sum_n w_n (m_{ij'}^n + \frac{1}{N}\alpha_{ij'})}, \qquad (27)$$

and similarly for the transition probabilities. The parameters α_{ij} are positive numbers derived from the prior distribution. All we need to do in the previous formulas to incorporate the regularization is to replace m_{ij}^n by $m_{ij}^n + \frac{1}{N}\alpha_{ij}$, and everything holds for this case as well.

Further details on the ME weighting scheme for HMMs will be given in a forthcoming publication.

Conclusion

The maximum entropy weighting scheme applies to database search with block alignments or hidden Markov models. The standard method of profile search can essentially be considered a special case of hidden Markov models, and thus falls within this framework. We have not, however, considered the use of substitution matrices for profile search, but we believe that ME weights can also be derived for this case.

Ideally one would like to find the weights at the same time as making the multiple alignment, but in some sense these are competing objectives. Loosely speaking, when aligning sequences one tries to maximize consensus in each column of the alignment, which corresponds to minimizing the entropy, whereas the objective of weighting is to maximize entropy. These competing objectives make the combined weighting and alignment problem difficult.

The ME weighting has an inherent danger, which it shares with other weighting schemes (*e.g.* Eddy, Mitchison, & Durbin 1994). If there is a sequence in the multiple alignment that is either wrongly aligned or assigned to the sequence family by mistake, it will automatically receive a very large weight compared to the rest of the sequences. In this case weighting will cause further damage, and it is therefore important to check the sequences with large weights, to see if there is a possibility of error.

References

Altschul, S.; Carroll, R.; and Lipman, D. 1989. Weights for data related by a tree. *Journal of Molecular Biology* 207(4):647–653.

Bairoch, A. 1993. The PROSITE dictionary of sites and patterns in proteins, its current status. *Nucleic Acids Research* 21(13):3097–3103.

Barton, G. 1990. Protein multiple sequence alignment and flexible pattern matching. *Methods in Enzymology* 183:403–428.

Cover, T., and Thomas, J. 1991. *Elements of Information Theory.* New York: John Wiley & Sons.

Eddy, S.; Mitchison, G.; and Durbin, R. 1994. Maximum discrimination hidden Markov models of sequence consensus. *J. Comput. Biol.* In press.

Felsenstein, J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Amer. J. Human Genet.* 25:471–492.

Gerstein, M.; Sonnhammer, E.; and Chothia, C. 1994. Volume changes in protein evolution. *Journal of Molecular Biology* 236(4):1067–1078.

Gribskov, M.; McLachlan, A. D.; and Eisenberg, D. 1987. Profile analysis: Detection of distantly related proteins. *Proceedings of the National Academy of Sciences of the United States of America* 84:4355–4358.

Henikoff, S., and Henikoff, J. 1991. Automated assembly of protein blocks for database searching. *Nucleic Acids Research* 19(23):6565–6572.

Henikoff, S., and Henikoff, J. 1994. Position-based sequence weights. *Journal of Molecular Biology* 243(4):574–578.

Krogh, A.; Brown, M.; Mian, I. S.; Sjölander, K.; and Haussler, D. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology* 235:1501–1531.

Levin, J.; Pascarella, S.; Argos, P.; and Garnier, J. 1993. Quantification of secondary structure prediction improvement using multiple alignments. *Protein Engineering* 6.

Press, W.; Flannery, B.; Teukolsky, S.; and Vetterling, W. 1986. *Numerical Recipes.* Cambridge: Cambridge University Press.

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77(2):257–286.

Sibbald, P., and Argos, P. 1990. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *Journal of Molecular Biology* 216(4):813–818.

Taylor, W. 1986. Identification of protein sequence homology by consensus template alignment. *Journal of Molecular Biology* 188:233–258.

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

Thompson, J.; Higgins, D.; and Gibson, T. 1994b. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Computer Applications in the Biosciences* 10(1):19–29.

Vingron, M., and Argos, P. 1989. A fast and sensitive multiple sequence alignment algorithm. *Computer Applications in the Biosciences* 5(2):115–121.

Vingron, M., and Sibbald, P. 1993. Weighting in sequence space: a comparison of methods in terms of generalized sequences. *Proceedings of the National Academy of Sciences of the United States of America* 90(19):8777–8781.