JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 11, Number 1, 2004

Pp. 181–193

Transition Priors for Protein Hidden Markov Models: An Empirical Study towards Maximum Discrimination

MARKUS WISTRAND and ERIK L.L. SONNHAMMER

ABSTRACT

Insertions and deletions in a profile hidden Markov model (HMM) are modeled by transition probabilities between insert, delete and match states. These are estimated by combining observed data and prior probabilities. The transition prior probabilities can be defined either ad hoc or by maximum likelihood (ML) estimation. We show that the choice of transition prior greatly affects the HMM's ability to discriminate between true and false hits. HMM discrimination was measured using the HMMER 2.2 package applied to 373 families from Pfam. We measured the discrimination between true members and noise sequences employing various ML transition priors and also systematically scanned the parameter space of ad hoc transition priors. Our results indicate that ML priors produce far from optimal discrimination, and we present an empirically derived prior that considerably decreases the number of misclassifications compared to ML. Most of the difference stems from the probabilities for exiting a delete state. The ML prior, which is unaware of noise sequences, estimates a delete-to-delete probability that is relatively high and does not penalize noise sequences enough for optimal discrimination.

Key words: hidden Markov model, transition prior probabilities, maximum discrimination, maximum likelihood, protein classification.

INTRODUCTION

METHODS USED TO PREDICT THE RELATIONSHIP between new protein sequences and sequences of known structure or function can be divided into two large categories: those that are based on pairwise sequence comparisons and those that use shared features from many related sequences. The general challenge for all these methods is to combine high sensitivity with high specificity, i.e., to discriminate between true members and noise. Profile hidden Markov models (profile HMMs) (Krogh *et al.*, 1994; Hughey *et al.*, 1996; Eddy, 1998) use shared characteristics from several training sequences and have been reported to perform better than pairwise methods (Karplus *et al.*, 1998; Park *et al.*, 1998; Gough *et al.*, 2001).

A profile HMM can either be trained from unaligned sequences in an optimization procedure or built from a trusted multiple alignment in a much simpler way. Here we use profile HMMs (for simplicity, we often simply say "HMMs") built from prealigned sequences. Once built, the HMM can be used to search

Center for Genomics and Bioinformatics, Karolinska Institutet, S-17177 Stockholm, Sweden.

for new family members in a sequence database or to score the probability of any single sequence being a member of the family. The family membership annotation can then be inferred to sequences that score better than a given threshold. Several databases of protein families with an HMM associated with each family exist today (Schultz *et al.*, 1998; Gough *et al.*, 2001; Haft *et al.*, 2001). One of those is Pfam (Bateman *et al.*, 2002) in which profile HMMs are built and searched using the HMMER package (Eddy, 2001), which we have used for this study.

In HMMER 2.2, a profile HMM has a basic topology of three states (match, insert, and delete) connected by seven transitions (Fig. 1). There is a probability associated with each transition, as well as with the emission of any of the 20 amino acids, being in a match or insert state. A path through the model is a series of states leading from the Begin to the End state. Aligning a sequence with respect to the HMM corresponds to finding the most probable path through the model that emits the sequence. This is normally done using dynamic programming. The likelihood of the path provides a score that can be used for deciding if the sequence is a family member. In HMMER, extreme value statistics is further used to estimate the statistical significance of a score found during a database search.

Building the HMM involves estimating all probabilities in the model. This estimation process basically corresponds to estimating *posterior probabilities* by combining *prior probabilities* and *count events* from the columns of the multiple alignment. Using prior probabilities is a way to restrict the effective freedom of model parameters, in case training data does not contain enough information to be fully reliable. For instance, if few training sequences are available, there is a relatively high risk that they are not good representatives of the underlying family. An HMM built from such sequences would be overfitted to training data; that is, it would model training sequences well but perform poorly for detecting related homologues. If we have prior knowledge about what the model should look like, we can take this into account when estimating model parameters. Sjölander *et al.* (1996) exemplify this with the case of having an alignment of three sequences where all three happen to have an isoleucine at a certain position. Rather than be confident, saying that the probability of observing an isoleucine at this position is 1, we would like to have a procedure for assigning probabilities for observing other amino acids given the observation. Another example is when we observe only transitions from match to match state in the alignment. What probability would we then assign to a transition from match to delete state or from delete to delete state? Both in the case of transitions between states and in the case of amino acid emissions, we clearly need to add prior knowledge to the count events. The relative importance of the prior should diminish when the number of training sequences increases. Different approaches have been developed to deal with emission prior probabilities (Brown *et al.*, 1993; Tatusov *et al.*, 1994; Karplus *et al.*, 1995; Sjölander *et al.*, 1996) as well as the related problem of sequence weighting to avoid biases in the training sequences (Gerstein *et al.*, 1994; Henikoff *et al.*, 1994; Eddy *et al.*, 1995; Krogh *et al.*, 1995; Karchin *et al.*, 1998). The common idea behind the most popular prior methods has been to capture the underlying distribution behind the observed vector, using either Dirichlet mixtures or substitution matrices.

Benchmarking of methods for detection of remote homology has most often focused on discrimination between members of different protein families (Henikoff *et al.*, 1996; Karplus *et al.*, 1998; Lindahl *et al.*, 2000; Madera *et al.*, 2002). This takes the tradeoff between sensitivity (avoiding false negatives) and specificity (avoiding false positives) into account. Tightening the model to reduce the number of false

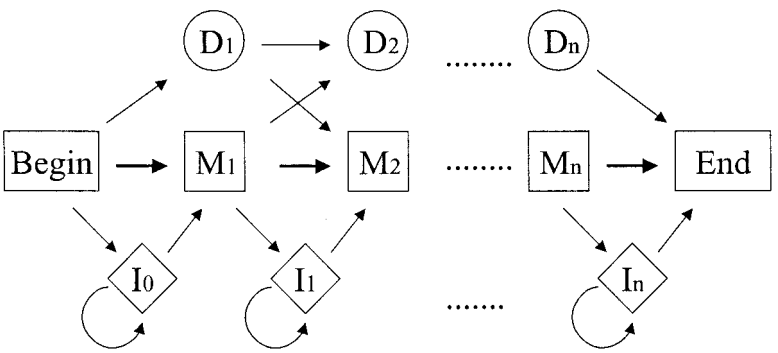

FIG. 1. Overview of the plan 7 architecture of HMMER 2.2. D stands for delete state, M for match state, and I for insert state. Arrows indicate the allowed transitions between states.

positives detected, i.e., increasing the specificity, generally decreases the model's sensitivity and vice versa. The goal is therefore to find an optimal balance that maximizes the discrimination between members of different families. However, when using sequence-based protein families, there is a risk that members of different families are distantly related; yet this may not be obvious from the sequence. In such cases, a lack of discrimination is actually biologically correct and might be a good property of the HMM. One can avoid this problem by using only proteins with a known 3D structure, but that both limits and biases the examples. To get a richer and more representative dataset, one can use, for instance, Pfam, but then it makes more sense to measure discrimination against sequences that are guaranteed to be unrelated, e.g., reversed real sequences.

The discriminatory power of an HMM will depend on the procedure for combining observations and priors into posterior probabilities. In this paper, we have focused on how transition priors influence the discrimination performance of HMMs. Little work has previously been done on transition priors. We have here changed each parameter in the transition prior systematically to explore the parameter space and measured the resulting discrimination using Pfam as test data. This allowed us to test whether transition priors estimated by maximum likelihood discriminate optimally. The results indicate that maximum likelihood transition priors discriminate poorly. Given that they were estimated entirely without taking specificity into account, this is not surprising. From our scanning experiments, we empirically extracted a transition prior that consistently optimized discrimination and recommend this prior to be employed for general HMM database searching.

Prior probability distributions

Prior knowledge can be provided in many different ways (Durbin *et al.*, 1998). One way is to add extra counts, *pseudocounts*, to the observed data counts. The pseudocounts can be added in proportion to our belief of how likely one event is compared to another, e.g., how likely a *match* → *match* transition is compared to a *match* → *delete* transition (see Fig. 1). In HMMER 2.2, the default choice for transition prior probabilities is to use Dirichlet densities, which can be seen as an advanced way of adding pseudocounts. A Dirichlet density is a multinomial distribution over probability vectors (for HMMs, probability vectors of transitions between states).

Generally, a Dirichlet density ρ over probability vectors of k parameters $\vec{p} = (p_1, \ldots, p_k)$ is defined by k parameters $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ as

$$\rho(\vec{p}) = \frac{\prod_{i=1}^{k} p_i^{\alpha_i - 1}}{Z(\vec{\alpha})} \tag{1}$$

where $Z(\vec{\alpha})$ is a normalizing factor defined by the gamma function:

$$Z(\vec{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_i \alpha_i\right)}. \tag{2}$$

It can be shown that the mean of a Dirichlet density is equal to the normalized mean of its parameters, i.e.,

$$E[p_i] = \frac{\alpha_i}{\sum_i \alpha_i}, \tag{3}$$

and that the variance of the density is inversely proportional to the sum of its parameters $\sum_i \alpha_i$. The ratio between α-values can therefore be seen as a measure of the bias of the prior probabilities for one event compared to another, and a high sum of α-values indicates a high belief in the prior.

In this paper, we will consider how to optimally set the transition prior probabilities in HMMER 2.2. There are three "types" of transitions in HMMER 2.2: from a match state, from a delete state, and from

an insert state; and an independent Dirichlet density models each of those. In all, the transition prior probabilities are thus characterized by seven α-values: α_{MM}, α_{MD}, α_{MI} for transitions from match state, α_{II}, α_{IM} for transitions from insert state, and α_{DD}, α_{DM} for transitions from delete state. For instance, the Dirichlet density for the match state transition prior is a probability distribution over probability vectors for transitions from the match state:

$$\rho(p_{MM}, p_{MI}, p_{MD}) = \frac{p_{MM}^{\alpha_{MM}-1} p_{MI}^{\alpha_{MI}-1} p_{MD}^{\alpha_{MM}-1}}{Z(\alpha_{MM}, \alpha_{MI}, \alpha_{MD})}. \quad (4)$$

It has a mean $(p_{MM}, p_{MD}, p_{MI}) = (\alpha_{MM}, \alpha_{MD}, \alpha_{MI})/(\alpha_{MM} + \alpha_{MD} + \alpha_{MI})$, and the higher the sum $\alpha_{MM} + \alpha_{MD} + \alpha_{MI}$, the more peaked it is around this mean.

How does HMMER 2.2 get from the Dirichlet prior and observed counts to the probability parameters in the HMM? This is done by first combining the prior distribution and the counts using Bayes' theorem to form the posterior probability distribution and then taking the integral of this, called the *posterior mean estimate*. The derivation of the resulting equation actually implemented in HMMER 2.2 is beyond the scope of this paper, but details can be found elsewhere (see Durbin *et al.*, 1998; Sjölander *et al.*, 1996). Here, we present only the resulting equation (Equation 5, the second equality is nontrivial) which reveals a nice feature of Dirichlet priors: the α-values characterizing the prior can be regarded as *pseudocounts* when estimating the posterior probabilities of transitions used in the HMM.

$$\hat{t}_{ij} = \int_{\vec{t}} P(t_{ij}|\vec{t}) \cdot P(\vec{t}|\vec{\alpha}, \vec{n}) d\vec{t} = \frac{n_{ij} + \alpha_{ij}}{\sum_k n_{ik} + \alpha_{ik}} \quad (5)$$

Here, $\hat{t}_{ij}$ is the posterior mean estimate of the transition probability from state i to j, $P(t_{ij}|\vec{t})$ is the probability of transition t_{ij} given the probability distribution of transitions $\vec{t}$, $P(\vec{t}|\vec{\alpha}, \vec{n})$ is the posterior probability of the distribution $\vec{t}$ given the observed transitions $\vec{n}$ and the Dirichlet density $\vec{\alpha}$. Further, α_{ij} is the prior transition probability between the states i and j, and n_{ij} are the observed transitions between these states. The sum is over all possible transitions from state i. The estimate interpolates smoothly between prior probabilities and observed counts. If no counts are available, the estimated transition probability $\hat{t}_{ij}$ is entirely specified by the prior as the ratio between the α_{ij} and the sum of all α's. When much data is available, the estimate will instead be almost entirely determined by the counts. The sum over α-values determines the relative importance of the prior in comparison to the observed data, i.e., how much data is needed to change our prior belief.

Estimating the transition prior

The prior probabilities themselves have to be estimated. One approach is to use maximum likelihood (ML), i.e., to find the parameters $\alpha_1, \ldots, \alpha_k$ that optimally fit the Dirichlet distribution to observed count events from a large set of data. The ML approach has been explored for estimating emission prior probabilities using count vectors of amino acids from trusted multiple alignments (Sjölander *et al.*, 1996), and it has been argued that transition prior probabilities can be treated analogously (Durbin *et al.*, 1998; Baldi *et al.*, 1998). This would incorporate our prior knowledge of how likely a deletion, for instance, is in a multiple alignment of a sequence family and thus its cost when searching for homologous sequences. The disadvantage of such an approach is that most often we are not interested in scoring true hits as high as possible but rather in maximizing the discrimination between true members and noise sequences.

An alternative approach to estimate transition prior probabilities is to empirically observe which prior parameters reliably produce optimal discrimination. This is a less theoretical approach to prior estimation than the maximum likelihood approach, but seems currently to be the only way to optimize discrimination.

METHODS AND DATA

The basic benchmark strategy we used to compare different settings of the transition prior was to construct sets of training sequences accompanied by remote homologues used as test sequences. For a

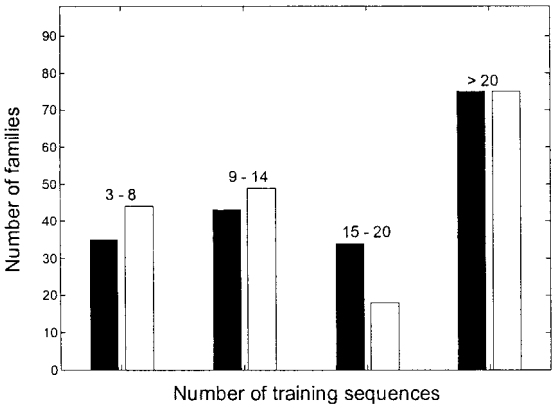

FIG. 2. Distribution of number of training sequences in Group A (black bars) and Group B (white bars).

specific choice of transition prior parameters, HMMs were built from the training sequences, and the test was to detect the test sequences buried in a huge number of noise sequences of about the same length.

For this, we used data from the 3,360 seed alignments in Pfam 7.0. Training and test sets were constructed in the following way for all families in the database:

- All sequences of length $< 70\%$ of the family mean length were discarded.
- Each family was partitioned into subclusters such that no sequences in different clusters had $> 20\%$ sequence identity.
- Families ending up with only one cluster were discarded.
- For the remaining families, the sequences in the largest cluster were chosen as training sequences. For a few families, sequences in one or more of the remaining clusters were added to the training sequences so that the training sequences always made up at least 50% of the total number of sequences. All other clusters were merged into the set of test sequences.
- To avoid redundancy among the test sequences, sequences were discarded until there were no two test sequences with $> 80\%$ sequence identity.

After this procedure, 373 families were left. These were divided into group A and group B with 187 and 186 families, respectively, and all the following experiments were done in parallel on both groups. By comparing the result for the two groups we reduce the risk for overfitting to data. The distributions of training and test sequences are shown in Figs. 2 and 3. The mean and median number of training sequences per family were 35.0 and 17 in group A and 32.3 and 15 in group B, while the mean and median number of test sequences per family were 3.7 and 2 in both groups.

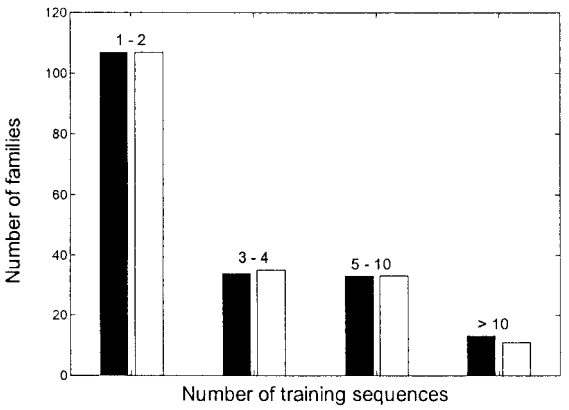

FIG. 3. Distribution of number of test sequences in Group A (black bars) and Group B (white bars).

In Pfam it is not straightforward to define negative examples, because distantly related homologues may be found in different families. We therefore believe that the best solution for Pfam is to employ reversed real sequences as negative examples. These are, for sure, not family members and should, overall, have a realistic amino acid composition. Noise files were constructed by selecting fragments of appropriate length randomly from SWISS-PROT and reversing them C to N. In each file, all noise sequences were of the same length, from noise100 (100 amino acids) to noise1000 in increments of 100. Each file contained 5,000 noise sequences except for noise1000 where there were only 2,874 sequences. Each family was associated with the noise file with the smallest sequence length above the median length of the family's test sequences.

The benchmark strategy was the following. For each transition prior, HMMs were built from the training sets (i.e., 187 HMMs for group A and 186 HMMs for group B). All other HMM building parameters were kept as default, which means that the models were configured for searches that are global with respect to the HMM and local with respect to the sequence. All test sequences, as well as the noise sequences, were then aligned to the corresponding HMMs and ranked in a list from best to worst based on the raw log-odds score. A cutoff score was chosen for each family so as to get the minimum error rate (MER) defined as the minimum number of false positives plus false negatives. For instance, if there are two positive test sequences corresponding to a certain family and the rank list of the five highest scoring sequences are *member*, *false*, *member*, *false*, *false*, the cutoff score would be set to the score of the second highest scoring positive sequence. This gives an MER score of one (one false positive, no false negatives), which is optimal given the rank list. The number of test sequences for a family sets an upper limit to the MER. A family with many test sequences can therefore contribute with a much higher MER than a family with few test sequences. We noted that this could introduce an undesired bias to our benchmark, as the test sequences for a certain family often are relatively similar. Therefore, we divide the MER with the number of test sequences, which gives a value between 0 and 1. Summing these values for all families in the group gave us the minimum error rate sum (MERS) for a specific prior. The MERS is a global measure of the discriminatory power of the specific transition prior employed, and we like it to be minimized.

Prior information was changed systematically in three sets of runs, one for each of the three Dirichlet densities of the transition prior, and the MERS was calculated for each prior setting. In the M-runs, α_{MI} and α_{MD} were changed; in the I-runs, α_{II} and α_{IM} were changed; and in the D-runs, α_{DD} and α_{DM} were changed. All parameters not changed in one particular set of runs were kept at their default values, as were all other settings in HMMER 2.2. Parameter α_{MM} is arguably the prior parameter with the least importance as the *match* $\rightarrow$ *match* transition is by far the most common transition observed in the alignments; hence, this parameter was never changed.

The ratio between α-values will determine their relative importance, and the sum of α-values will determine the importance of the prior in comparison to observed counts (see Equation 5 and earlier discussion). For the two α-values changed in each set of runs, we therefore varied the ratio between them and their sum, and for each setting of parameters, the MERS value was computed. This gave us a matrix, showing MERS-values as a function of the ratio and the sum of the α-values changed.

Prior estimation from maximum-likelihood

We used data from the seed alignments in Pfam 7.2 to perform a maximum-likelihood estimate of each of the Dirichlet densities specifying the transition prior; i.e., we sought the parameters that optimize the probability of the observed transitions in the alignments. This is equivalent to finding the parameters $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ that minimize the negative log likelihood (NLL) of the observed data, i.e.,

$$\begin{aligned} f(\vec{\alpha}) &= -\sum_{t=1}^{m} \log P(\vec{n}_t|\vec{\alpha}, |\vec{n}_t|) \\ &= -\left(\sum_{t=mm,mi,md} \log P(n_t|\alpha_t, 3) + \sum_{t=im,ii} \log P(n_t|\alpha_t, 2) + \sum_{t=dm,dd} \log P(n_t|\alpha_t, 2) \right) \end{aligned} \tag{6}$$

where $\vec{n}$ is a count vector, $|\vec{n}|$ is the number of observed transitions in the count vector, and m is the number of count vectors. Here, Sjölander *et al.* (1996) is also an essential reference for the implementation. Specific

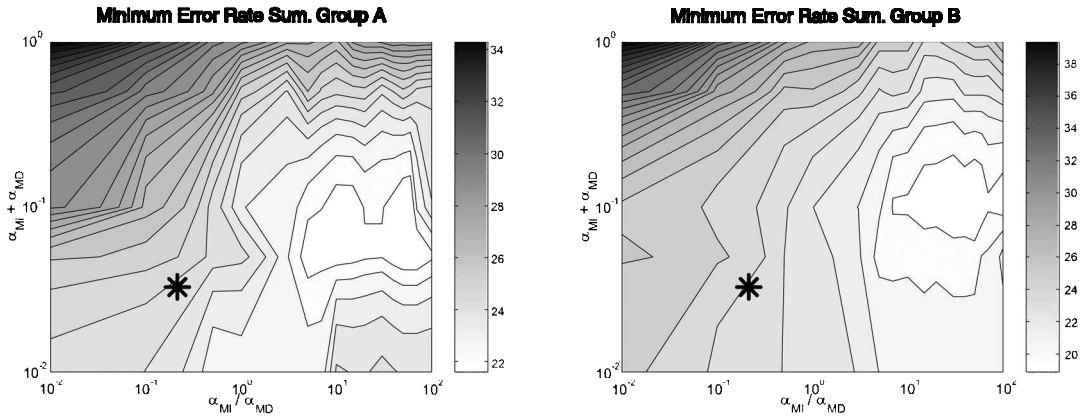

FIG. 4. Discrimination as a function of α_{MD} and α_{MM} keeping all other α-values at default values. White is the best region (lowest number of errors), and the asterisk indicates the ML prior.

for the transition prior is that we are estimating three distributions (match state, delete state, and insert state) rather than one (e.g., emission distribution).

HMMER 2.2 provides the possibility of saving count vectors from model building, which we did for all alignments without using any weighting scheme. In all, this produced about 858,000 count vectors. The minimum of the objective function was estimated using a conjugate gradient descent algorithm starting with initial values picked at random. To compensate for the risk of getting trapped in local minima, we reran the optimization several times. Each time, the algorithm reproduced the ratio between individual parameters, while the sum of the parameters varied in a narrow range.

RESULTS

The exploration of transition prior parameters was done independently for group A and group B families, in order to see if the results agree and can be considered generalizable. In HMMER 2.2, the transition prior consists of three separate Dirichlet densities: transitions from match, insert, and delete states. We varied the parameters for each density separately while keeping the other two constant. In 21 of the families in both group A and group B, none of the true positives were detected above the MER cutoff no matter the setting of the transition prior. We chose not to include these families in the analysis, which basically just removes a constant factor from the MER calculations. The results are shown as six contour plots (Figs. 4–6), one for each Dirichlet density and group. The MERS values are presented as a function of the ratio and sum

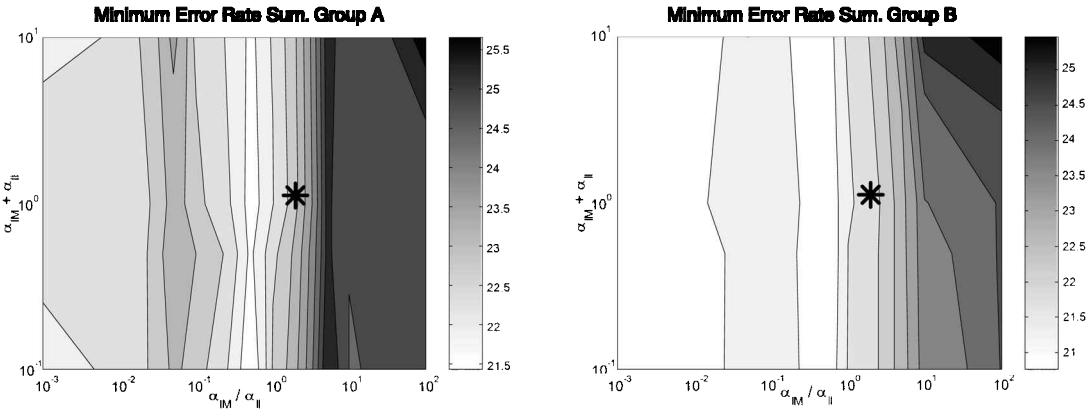

FIG. 5. Discrimination as a function of α_{IM} and α_{II} keeping all other α-values at default values.

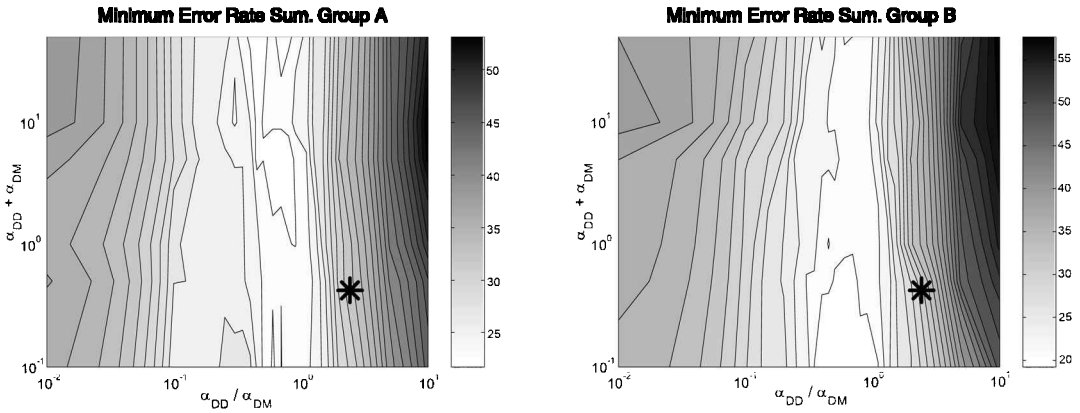

FIG. 6. Discrimination as a function of α_{DD} and α_{DM} keeping all other α-values at default values.

of α-values. A dark area in the contour plots represents a poorly discriminating region (high MERS) while a light area is discriminating well (low MERS), as indicated by the shading scale bar next to each plot. In each of the figures, the maximum likelihood estimated set of α-values is marked with an asterisk.

Match transitions

Here, only α_{MI} and α_{MD} were varied, while α_{MM} was kept fixed as it is the overwhelmingly dominating transition from match states. The optimal white area in Fig. 4 is to the right of the asterisk, indicating that a 15-fold higher ratio α_{MI}/α_{MD} would increase the performance of HMMER 2.2, compared to the ML prior. The sum $\alpha_{MI} + \alpha_{MD}$ does influence the performance, but this could partly be due to that the relative importance of α_{MM} shifts as the sum $\alpha_{MI} + \alpha_{MD}$ is changed.

Insert transitions

The values α_{II} and α_{IM} were varied in this set of runs (Fig. 5). In the parameter region explored, the importance of the parameter change is less than was the case for the match transitions (see the scale of the shading bar). A threshold exists in the α_{IM}/α_{II} plot. For $\alpha_{IM}/\alpha_{II} < 1$, there is a vast area in which performance is almost unchanged, but when the ratio exceeds 1, the performance deteriorates. The ML prior is not optimal and would gain from a decreased ratio α_{IM}/α_{II}, although not much. The sum $\alpha_{IM} + \alpha_{II}$ seems to be of little importance for the performance.

Delete transitions

Here, the α-values governing transitions from delete states were varied (Fig. 6), and we observed the biggest impact on the MERS. The contour plots show optimal performance for values of the ratio α_{DD}/α_{DM} in the region 0.3–1. The ML prior has a much higher value of this ratio (about four times) and an MERS considerably higher than the best choice of transition prior possible. Once again, the sum $\alpha_{DD} + \alpha_{DM}$ seems to be of little importance.

Overall, we find a reasonably good agreement between families in group A and B, suggesting that the optimal areas in the parameter space are not specific to a set of families but are generally valid. One should, of course, ignore small deviations and draw conclusions only from the larger trends.

An empirical prior for maximum discrimination

Having extracted the optimal parameters for the three individual densities in the prior, we then combined these to form a new prior that should be close to the point of optimal discrimination. Because each density was optimized using suboptimal settings in the other settings, however, the new prior did not discriminate as well as the optimal points in the previous runs. We therefore performed additional optimization of the delete transitions using the new parameters for the match and insert priors to generate a prior that is very close to the achievable optimum. We call this prior the empirical maximum discrimination (EMD) prior.

Although there is some arbitrariness in defining the maximum, we consider EMD an appropriate name as it highlights the way the prior was estimated. We tried to optimize the prior further by the Downhill Simplex Method iterative optimizing procedure with an initial guess based on the runs we had performed, but this did not yield significantly better discrimination.

Using the EMD prior increases the HMM's discrimination considerably compared to the ML prior. Could it be that the ML prior got stuck in a local minimum when it was estimated? This is not the case. The NLL (Equation 6), which is minimized in the estimation process, is significantly higher (worse) for the EMD prior than for the ML prior (Table 1). Another possibility we considered was whether using a weighting scheme when generating the count vectors would affect the estimation of the ML prior. For instance, using the Blosum62 weighting scheme in *hmmbuild* could perhaps give an ML prior very different from the ML prior we get using unweighted counts. However, this possibility was also ruled out (Table 1).

Our test was based on setting a cutoff for each family to minimize the number of false positives and false negatives, and this number is the minimum error rate for that family using a specific prior. This is an approach in which all misclassifications are valued equally, and perhaps this is not always what we want. In case we are mainly focusing on sensitivity, we would, for instance, choose a lower cutoff level. Would the EMD prior in such a case still do better than the ML prior? We have analyzed this by counting the number of false positives as a function of false negatives for all families in group A and group B. In practice, we went through the ranked list of log-odds scores for each family and counted the number of false positives for each level of false negatives and then summed them for each group. The results show that for all levels of false positives accepted we get a lower number of false negatives using the EMD prior than when using the ML prior (Fig. 7). The EMD prior is thus more sensitive than the ML prior for all levels of specificity.

Prior versus counts

When an HMM is built from an alignment of few sequences, the importance of having a good prior is high. If there are many sequences in the alignment, the prior will be less important. To see how well these statements hold, we split our results for delete state transitions into two sets: one set of all families with < 20 training sequences and a second set of all families with $>= 20$ training sequences. Fixing $\alpha_{DD}+\alpha_{DM}$ to 1, we plotted the MERS against α_{DD}/α_{DM} for both sets. The results confirmed the statements (Fig. 8). The set with few training sequences is much more dependent on the choice of transition prior. A poor choice of transition prior is thus especially detrimental when having few training sequences.

DISCUSSION

We have evaluated the settings of transition priors in HMMER 2.2 using data from Pfam 7.0. Dirichlet densities are the default way of modeling transition priors and are considered to perform well, which is why we have used them here. Intuitively, a maximum likelihood estimate of the parameters describing the density from a huge set of data can be thought to give good results because it models the background rate of transitions. However, we have shown that using a reasonably extensive set of data divided into two groups to avoid overfitting, this is not the case based on a criterion of optimizing the discrimination between false positives and false negatives. The setting of transition priors that we suggest increases the discrimination of HMMs considerably compared to the ML prior.

Why does the ML prior perform so poorly? One reason could be that the HMM architecture may have inherent biases as it is based on assumptions of proteins. Using regularization would then counteract this. This aside, we think the reason is that there are actually two distributions of transitions, one positive and one negative, that need to be considered. The positive distribution describes transitions observed in multiple alignments of related sequences and also the distribution of transitions one would expect to see in homologous sequences aligned to the alignment. This is our ML distribution. The negative distribution, on the other hand, describes the distribution one would expect to see in noise sequences aligned to the multiple alignment. Although the likelihood for deletions and insertions would probably be higher in the negative distribution, the two distributions would surely intersect to a certain degree. Some noise sequences

TABLE 1. α-VALUES CHARACTERIZING THREE TRANSITION PRIORS[a]

Prior	α_{MM}	α_{MI}	α_{MD}	α_{IM}	α_{II}	α_{DM}	α_{DD}	α_{MI}/α_{MD}	α_{DD}/α_{DM}	*MERS Gr. A*	MERS *Gr. B*	*NLL*
ML unweighted	0.924	0.006	0.027	0.744	0.375	0.125	0.302	0.22	2.4	36.4	38.6	$7.3*10^5$
ML weighted	1.064	0.007	0.033	0.755	0.407	0.147	0.363	0.21	2.5	36.4	38.4	$6.8*10^5$
MD	0.794	0.095	0.005	0.333	0.667	0.278	0.222	20	0.8	20.6	18.9	$10*10^5$

[a]The ML priors were estimated using a maximum likelihood approach and data from Pfam 7.2 seed alignments, without weighting and with "wblosum" weighting in *hmmbuild*. As can be seen, weighting does not make a significant difference. The EMD prior maximized the discrimination in our tests, yet its NLL with respect to Pfam 7.2 seed counts (unweighted) is significantly less optimal than for the ML priors. The main differences between ML and EMD priors are in the ratios α_{MI}/α_{MD} and α_{DD}/α_{DM}; the ratio α_{DD}/α_{DM} has the greatest influence on the MERS (see Fig. 6).

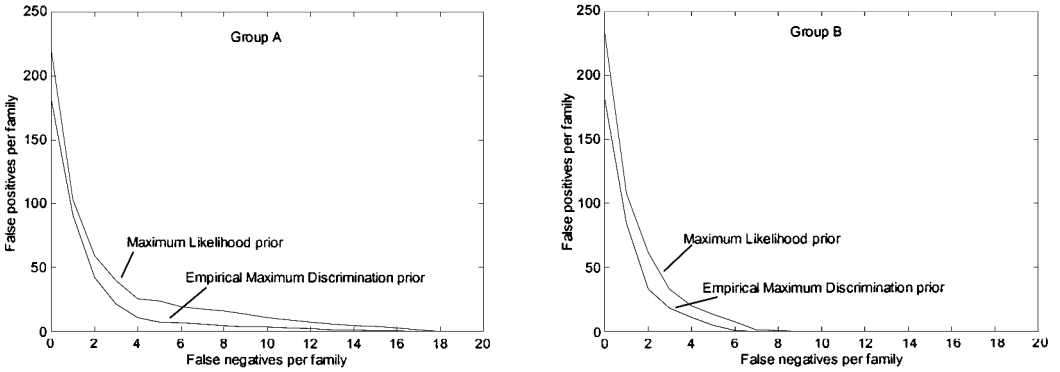

FIG. 7. Number of false positives for each level of accepted false negatives. The numbers are summed up for all included families in the respective group.

will therefore get good scores also using the ML prior. A discriminative prior distribution should be one that well describes the positive distribution but at the same time is different from the negative distribution. The EMD prior that we have presented here is an approximation of such a distribution.

Judging from our systematic parameter searches, this seems reasonable. The high ratio α_{DD}/α_{DM} of the ML prior compared to our EMD prior is most crucial for the difference in performance. This high ratio makes HMMs relatively prone to deletions, which allows them to model noise sequences using long deletions without reducing the score much. Using a prior that restricts allowance for deletions, we get a much better separation between true members and noise. This effect of restricting parameter space is most important when few training sequences are available (Fig. 8) and therefore little information about the protein family being modeled. When the number of training sequences increases, the importance of the transition prior decreases, but the EMD prior is still the best choice.

The default transition prior in HMMER 2.2 is supposed to have been estimated using maximum likelihood. Nevertheless, the default prior performs very well in our test: much better than the ML priors that we estimated and not much worse than the EMD prior. We tried to reproduce the default transition prior with the maximum likelihood estimation method applied to both seed and full alignments from several versions of Pfam ranging from 1.0 to 7.2, but failed. The details of how the default transition prior in HMMER was estimated could not be produced (G. Mitchison, personal comm.). We noticed that in all our ML estimates $\alpha_{MI} < \alpha_{MD}$ and $\alpha_{DD} > \alpha_{DM}$, but in the default prior and in the EMD prior these relationships are inverted. Our guess is therefore that the default prior comes from an ML estimate using an early unreleased version of Pfam and that the parameter pairs $(\alpha_{MI}, \alpha_{MD})$ and $(\alpha_{DD}, \alpha_{DM})$ might have

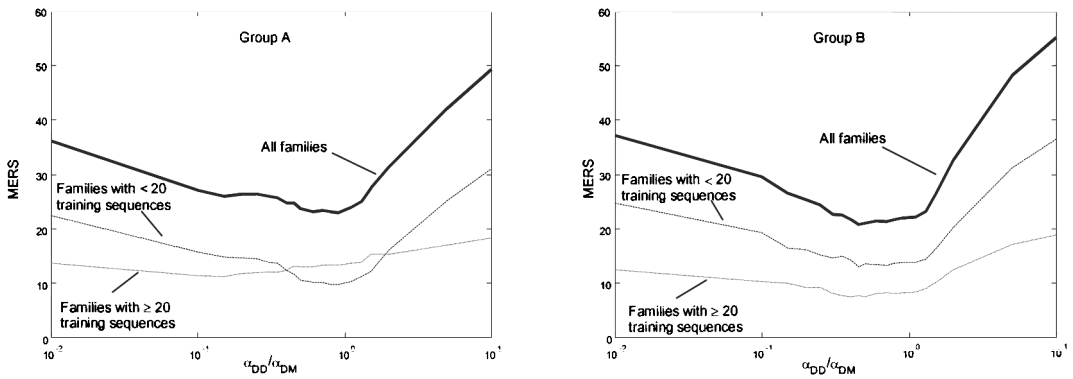

FIG. 8. Good prior information is most important when the number of training sequences is low. Keeping $\alpha_{DD} + \alpha_{DM} = 1$, we changed the α_{DD}/α_{DM} and calculated the MERS. All other parameters were kept as default. The result for all families is shown by the bold line and is split into two sets based on the number of training sequences: > 20 and ≤ 20.

been accidentally inverted. Luckily, this adds considerably to the discrimination of profile HMMs built by HMMER. We noted that the inverses of the α_{MI}/α_{MD} and α_{DD}/α_{DM} ratios produced by ML estimates using data generated with the *–fast* option in *hmmbuild* came very close to these ratios in the default prior. The *–fast* option makes *hmmbuild* employ simple heuristics when choosing model architecture, given the alignment. By default, a *maximum a posteriori* algorithm with a special architecture prior is used which produces more match states than *–fast* does. It therefore seems likely that some simple early method was used for choosing model architecture and that the (α_{MI}, α_{MD}) and (α_{DD}, α_{DM}) values were swapped.

Many other benchmarking tests have used the SCOP database rather than Pfam and have defined positive and negative examples based on the superfamily classification. This has the advantage that also the negative sequences are real protein sequences. However, the SCOP database is biased to globular proteins, and as we wanted a more general test we decided to use Pfam. We realize that our choice to use randomly picked and reversed sequences as negative sequences could be criticized. However, preliminary results using the SCOP database and defining positive and negative sequences based on the superfamily classification are in agreement with what we have presented here.

We have shown that estimating a transition prior in the form of a Dirichlet density estimated by maximum likelihood does not give good discrimination in HMM searches. The empirical procedure we have used instead is time consuming and not theoretically well founded. Today, it is not clear to us how this could be done more effectively but ad hoc methods like ours are often unavoidable. One way perhaps worth trying would be to still use maximum likelihood but a more sophisticated approach. To this end, there are at least two alternatives. One could try to model the transition prior using a Dirichlet mixture of a certain number of densities. A mixture can account for the fact that different columns in the alignment perform best with different prior information, while a simple density uses the same prior information for all columns. Another approach would be to estimate Dirichlet densities for different structural environments (e.g., helix-to-helix, buried-to-buried etc.) using structurally marked-up alignments. When building the profile HMM, either a mixture of the structurally based densities could be used or a single one, then again based on the mark-up. As the frequency of transitions varies a lot in different structural regions of alignments, this could perhaps augment the performance.

ACKNOWLEDGMENTS

We thank Michael Åsman for initial analyses, Sean Eddy for providing prior estimation and benchmarking tools, and Kevin Karplus for helpful discussions. This work was supported by grants from Pharmacia Corporation and the Swedish Knowledge Foundation.

REFERENCES

Baldi, P., and Brunak, S. 1998. *Bioinformatics: The Machine Learning Approach*, MIT Press, Cambridge, MA.

Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S.R., Griffiths-Jones, S., Howe, K.L., Marshall, M., and Sonnhammer, E.L. 2002. The Pfam Protein Family Database. *Nucl. Acids Res.* 30, 276–280.

Brown, M., Hughey, R., Krogh, A., Mian, I. S., Sjölander, K., and Haussler, D. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 1, 47–55.

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

Eddy, S.R. 1998. Profile hidden Markov models. *Bioinformatics* 14, 755–763.

Eddy, S.R. 2001. HMMER: Profile hidden Markov models for biological sequence analysis. (*hmmer.wustl.edu/*).

Eddy, S.R., Mitchison, G., and Durbin, R. 1995. Maximum discrimination hidden Markov models of sequence consensus. *J. Comp. Biol.* 2, 9–23.

Gerstein, M., Sonnhammer, E.L.L., and Chothia, C. 1994. Volume changes in protein evolution. *J. Mol. Biol.* 236, 1067–1078.

Gough, J., Karplus, K., Hughey, R., and Chothia, C. 2001. Assignment of homology to genome sequences of hidden markov models that represents all proteins of known structure. *J. Mol. Biol.* 313, 903–919.

Haft, D.H., Loftus, B.J., Richardson, D.L., Yang, F., Eisen, J.A., Paulsen, I.T., White, O. 2001. TIGRFAMs: A protein family resource for the functional identification of proteins. *Nucl. Acids. Res.* 29, 41–43.

Henikoff, S., and Henikoff, J.G. 1994. Position-based sequence weights. *J. Mol. Biol.* 243, 574–578.

Henikoff, J.G., and Henikoff, S. 1996. Using substitution probabilities to improve position-specific scoring matrices. *Comput. Appl. Biosci.* 12, 135–143.

Hughey, R., and Krogh, A. 1996. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS* 12, 95–107.

Karchin, R., and Hughey, R. 1998. Weighting hidden Markov models for maximum discrimination. *Bioinformatics* 14, 772–782.

Karplus, K., Barrett, C., and Hughey, R. 1998. Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 14, 846–856.

Krogh, A., Brown, M., Mian, I.S., Sjölander, K., and Haussler, D. 1994. Hidden Markov models in computational biology: Applications to protein modelling. *J. Mol. Biol.* 235, 1501–1531.

Krogh, A., and Mitchison, G. 1995. Maximum entropy weighting of aligned sequences of proteins or DNA. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 3, 215–221.

Lindahl, E., and Elofsson, A. 2000. Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.* 295, 613–625.

Madera, M., and Gough, J. 2002. A comparison of profile hidden Markov model procedures for remote homology detection. *Nucl. Acids. Res.* 19, 4321–4328.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T., and Chothia, C. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.* 284, 1201–1210.

Schultz, J., Milpetz, F., Bork, P., and Ponting, C.P. 1998. SMART, a simple modular architecture research tool: Identification of signalling domains. *Proc. Natl. Acad. Sci. USA* 95, 5857–5864.

Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I.S., and Haussler, D. 1996. Dirichlet mixtures: A method for improved detection of weak but significant protein sequence homology. *CABIOS* 12, 327–345.

Tatusov, R.L., Altschul, S.T., and Koonin, E.V. 1994. Detection of conserved segments in proteins: Iterative scanning of sequence databases with alignment blocks. *Proc. Natl. Acad. Sci. USA* 91, 12091–12095.

Address correspondence to:
Erik L.L. Sonnhammer
Center for Geonomics and Bioinformatics
Karolinska Institutet
S-17177 Stockholm, Sweden

E-mail: Erik.Sonnhammer@cgb.ki.se