BIOINFORMATICS

Vol. 21 Suppl. 1 2005, pages i251–i257
doi:10.1093/bioinformatics/bti1014

An HMM posterior decoder for sequence feature prediction that includes homology information

Lukas Käll[1], Anders Krogh[2] and Erik L. L. Sonnhammer[1,]*

[1]Center for Genomics and Bioinformatics, Karolinska Institutet, SE-17 177 Stockholm, Sweden and [2]The Bioinformatics Center, University of Copenhagen, Universitetsparken 15, DK-2100 Copenhagen, Denmark

Received on January 17, 2005; accepted on March 27, 2005

ABSTRACT

Motivation: When predicting sequence features like transmembrane topology, signal peptides, coil–coil structures, protein secondary structure or genes, extra support can be gained from homologs.

Results: We present here a general hidden Markov model (HMM) decoding algorithm that combines probabilities for sequence features of homologs by considering the average of the posterior label probability of each position in a global sequence alignment. The algorithm is an extension of the previously described 'optimal accuracy' decoder, allowing homology information to be used. It was benchmarked using an HMM for transmembrane topology and signal peptide prediction, Phobius. We found that the performance was substantially increased when incorporating information from homologs.

Availability: A prediction server for transmembrane topology and signal peptides that uses the algorithm is available at http://phobius.cgb.ki.se/poly.html. An implementation of the algorithm is available on request from the authors.

Contact: Erik.Sonnhammer@cgb.ki.se

1 INTRODUCTION

Hidden Markov models (HMMs) are successfully being used in many different areas within bioinformatics. The applications include transmembrane topology predictors (Sonnhammer *et al*., 1998; Tusnady and Simon, 1998), signal peptide predictors (Nielsen and Krogh, 1998), coil–coil protein predictors (Delorenzi and Speed, 2002), gene predictors (Krogh *et al*., 1994b; Burge and Karlin, 1997), secondary structure predictors (Bystroff *et al*., 2000), sequence alignment programs (Needleman and Wunsch, 1970) and tools for sequence homology detection (Krogh *et al*., 1994a; Eddy, 1998). In many of the applications, it makes sense to take the homologs to the query sequence into consideration, since sequence features are likely to be shared between homologs. We present here a general HMM decoding algorithm that, rather than decoding a profile, calculates the probabilities for sequence features from each homolog individually before taking the alignment into account.

1.1 Decoding a single sequence

An HMM can be used in two conceptually different ways: as in the case of detecting sequence homology, where one asks whether a query sequence fits a model, or as in the case of the other applications mentioned above, where one is interested in determining an optimal path through a model. In the former case, the score given by the forward algorithm, i.e. the sum of probabilities of all paths through the model, is considered the most accurate measure. In this study, we will mainly discuss the latter case, where the path through the model is of interest. For many such applications the Viterbi algorithm, which finds the most probable path through the model, is used. However, a drawback is that there might be many similar paths through the model with probabilities that add up to a higher probability than the single most probable path.

A way to recognize similar paths is to assign a common label to the states that represent the same kind of sequence feature (Krogh, 1994). For example when predicting protein secondary structure, states representing α-helical amino acids can be assigned one label, states that represent amino acids in β-strands a second label and states representing amino acids in loops a third label. In this setting, one could focus on the labeling a sequence is predicted to have, rather than the exact state path. We can determine the most probable labeling of a sequence, i.e. the highest sum of probabilities of all paths having the same way to label a sequence, with the 1-best algorithm (Schwartz and Chow, 1990).

Rather than looking for the overall and most likely labeling of a sequence, one is often interested in maximizing the number of positions that are correctly predicted. The posterior label probability (PLP) is the probability of a label at a certain position in the sequence, given the sequence and the model. In other words, it is the normalized sum of probabilities of all paths passing through the states with the label at a certain

*To whom correspondence should be addressed.

position of the sequence. To maximize the expected number of correct labels corresponds to selecting the label with highest PLP for each symbol in the sequence. This kind of prediction, however, is not guaranteed to be consistent with the HMM itself. For example, in the case of transmembrane topology prediction, results could be obtained where loops being predicted on the translocated side of the membrane are followed by a loop on the side that is not translocated without the presence of an interconnecting transmembrane segment. Such a prediction violates the 'grammar' of the model.

Holmes and Durbin (1998) presented an algorithm for finding a path that optimizes the expected accuracy, which could, with an extension of their definition, be viewed as the sum of the PLPs of a labeling. They call this the optimal accuracy algorithm. Here we describe a similar algorithm, which finds the maximal expected accuracy labeling consistent with the grammar of the HMM.

1.2 Handling homologs

In many applications it is reasonable to assume that the prediction accuracy is increased by taking the information from the homologs into account, since the homologs often have the same features as the query sequence.

Some of the previously published decoding algorithms that take homologs into account do this by predicting features for sequence profiles rather than individual sequences. The Viterbi or 1-best algorithm is used to calculate a path through the model common to all sequences in the profile. An 'emission score' of a state is calculated as a function of the vector of individual amino acid emission probabilities and the vector of relative amino acid frequencies at the position of interest in the sequence profile. One method to calculate such a score would be to use the scalar product to combine the two vectors (Martelli *et al*., 2002; Söding, 2005). An alternative method, that keeps the score as a probability, is to use the product of the emission probabilities raised to the power of the corresponding frequencies (Viklund and Elofsson, 2004; Edgar and Sjölander, 2004).

A problem with these approaches, when dealing with sequence feature prediction, is that gaps are handled either as a symbol of its own or by assigning gap emission probabilities proportional to the emission probabilities of the other symbols in the same alignment column. This is a disadvantage since the model length may be confused by the fact that a sequence profile contains gaps and inserts. The fact that the length of an alignment usually grows with the number of included sequences has to be compensated for, usually by ignoring positions in the profile or columns in the alignment where the query sequence contains a gap. This implies that the signal from the length model of the other sequences will be partly ignored.

A different approach is taken by Tusnady and Simon (1998), who reestimate the parameters in the HMM on the query sequence with an unsupervised Baum–Welch procedure before making the final prediction with a Viterbi decoder. When homologs are given, the reestimation is done on both the query sequence and the homologs, and subsequently, the new HMM is used for decoding the query sequence alone. In this setting there is no need for alignments, but it suffers from the problem that the model may give inconsistent predictions for the homologous sequences, because predictions are done independently.

In this work we describe a way to incorporate homology information by superimposing the PLPs for homologs into an average PLP matrix, which is used as input for the optimal accuracy algorithm. The main advantage of our method is that it applies the full probability model of the HMM to each included sequence individually before the contributing signals are merged. Thus, it recognizes the inherent grammar and length modeling of the HMM.

2 ALGORITHM

2.1 Optimal accuracy decoding

In order to describe our amendment to the optimal accuracy algorithm, we start by describing the algorithm in detail. The algorithm comprises two steps: first, the PLPs of the query sequence is calculated and second, based upon the PLPs, the optimal accuracy path is determined.

Consider an HMM with $N+1$ states with names from 0 to N, where the state 0 represents the start and end state of the HMM, i.e. the set of states is $\sigma = \{0, \ldots, N\}$. The label λ of a state i is given by the mapping $\Lambda(i) = \lambda$ and the set of states that have label λ is called $\sigma_\lambda \subset \sigma$. Therefore, $i \in \sigma_\lambda \iff \Lambda(i) = \lambda$. Let the emission probabilities of the states be given by $\mathbf{e} = (e_{ik})$, where $e_{ik} = P(x_t = k | \pi_t = i)$, and π_t and x_t are the stochastic variables representing the state and the emitted symbol at position t in the sequence. The transition probabilities are given by $\mathbf{a} = \{a_{ij}\}$, where $a_{ij} = P(\pi_{t+1} = j | \pi_t = i)$. Usually, some of the transition probabilities are set to zero in advance to avoid 'illegal' transitions. The non-zero transition probabilities define the underlying graph of the model. This graph structure restricts the possible labelings of a sequence, which we refer to as the grammar.

For a query sequence $\mathbf{x} = (x_t)$ for $t = 1, \ldots, T$ we want to predict a sequence of labels $\mathbf{l} = (l_t)$. For any predicted labeling, the expected number of correctly predicted labels is the sum of the posterior probabilities for those labels. The aim of the optimal accuracy decoding is to find the labeling that maximizes this number. However, since the model has a built-in grammar (defined by the non-zero transition probabilities), it is not optimal to pick the highest probability label at each position in the sequence but rather to choose the labeling with the highest accuracy that is consistent with the grammar.

Given the definitions above, we can now find a way to calculate the PLPs of $\mathbf{x}$. We begin by calculating the posterior

state probability of state i at position t in the sequence:

$$\begin{aligned} P(\pi_t = i|\mathbf{x},\mathbf{a},\mathbf{e}) &= \frac{P(\pi_t = i,\mathbf{x}|\mathbf{a},\mathbf{e})}{P(\mathbf{x}|\mathbf{a},\mathbf{e})} \\ &= \frac{P(\pi_t = i, x_1 \dots x_t|\mathbf{a},\mathbf{e})P(\pi_t = i, x_{t+1} \dots x_T|\mathbf{a},\mathbf{e})}{P(\mathbf{x}|\mathbf{a},\mathbf{e})} \\ &= \frac{f_{i,t}b_{i,t}}{P(\mathbf{x}|\mathbf{a},\mathbf{e})}. \end{aligned} \quad (1)$$

Here the forward variables, $f_{i,t}$, is given by the recursion

$$f_{i,t} = \begin{cases} \delta_{i0}, & t = 0 \\ e_{ix_t}\sum_{j\in\sigma} f_{j,t-1}a_{ji}, & t = 1,\dots,T \\ \delta_{i0}\sum_{j\in\sigma} f_{j,T}a_{j0}, & t = T+1 \end{cases} \quad (2)$$

and the backward variables, $b_{i,t}$, by

$$b_{i,t} = \begin{cases} \delta_{i0}, & t = T+1 \\ a_{i0}, & t = T \\ \sum_{j\in\sigma} a_{ij}e_{jx_{t+1}}b_{j,t+1}, & t = T-1,\dots,1 \\ \delta_{i0}\sum_{j\in\sigma} a_{0j}e_{jx_1}b_{j,1}, & t = 0. \end{cases} \quad (3)$$

Here we have used the Kronecker's δ_{ij} defined as

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j. \end{cases}$$

We know that

$$P(\mathbf{x}|\mathbf{a},\mathbf{e}) = f_{0,T+1} = b_{0,0}. \quad (4)$$

We can now calculate the PLP for the label λ at position t as

$$\begin{aligned} g_{\lambda,t} \equiv P(l_t = \lambda|\mathbf{x},\mathbf{a},\mathbf{e}) &= \sum_{i\in\sigma_\lambda} P(\pi_t = i|\mathbf{x},\mathbf{a},\mathbf{e}) \\ &= \sum_{i\in\sigma_\lambda} \frac{f_{i,t}b_{i,t}}{f_{0,T+1}}. \end{aligned} \quad (5)$$

Given these PLPs, we now want to find an optimal path through the model. However, we should do so under the constraint that the path should be a possible path through the model. Since we already applied the transition probabilities when calculating the PLPs it makes no sense to apply the full Markov model once again, when searching the best path. Instead, we have to rely on the graph structure (or grammar) of the HMM.

Our goal is to maximize the expected accuracy of a labeling of a sequence, i.e. the expected number of correctly predicted labels, as

$$A(\mathbf{l}) = \sum_{t=0}^{T+1} g_{l_t,t}. \quad (6)$$

This is an extension of the definition for pairwise sequence alignments made by Holmes and Durbin (1998). They define the expected accuracy of an alignment as the sum of the posterior probabilities for all aligned positions.

We define the transition possibilities $\mathbf{d} = (d_{ij})$ as

$$d_{ij} = \begin{cases} 0, & \text{if } a_{ij} = 0 \\ 1, & \text{if } a_{ij} > 0. \end{cases} \quad (7)$$

Now we can use a Viterbi inspired recursion to calculate the optimal accuracy, $\hat{A}_{j,t}$, to a position t in the sequence for state j.

$$\hat{A}_{j,0} = \begin{cases} 0, & \text{if } j = 0 \\ -\infty, & \text{if } j \neq 0 \end{cases} \quad (8)$$

$$\hat{A}_{j,t} = g_{\Lambda(j),t} + \max_{i\in\sigma} \hat{A}_{i,t-1}d_{ij}, \quad t = 1,\dots,T+1. \quad (9)$$

In analogy with the Viterbi algorithm we can use backpointers to track the path through $\hat{A}$ ending in $\hat{A}_{0,T}$ rendering the optimal accuracy labeling. Note that degeneracy in the best path between states with the same label is of no consequence for the resulting labeling.

A related algorithm to the optimal accuracy decoder, the 'posterior-Viterbi' decoder and its application to prediction of topology of β-barrel proteins is described in a recent paper by Fariselli *et al*. (2005). The main algorithmic difference in their approach lies in that they optimize the product of the PLPs, instead of the sum as we have done according to Equation (6).

2.2 Homolog handling extension

How can we incorporate information from homologs to the query sequence into the optimal accuracy algorithm? Our solution is to calculate the PLPs for each sequence individually, then take the average PLP for each label at each position of the alignment, and thereafter optimize the expected accuracy based on this average PLP.

Let us say that we have an alignment of the sequences $\mathbf{x}^1,\dots,\mathbf{x}^M$, where the mapping between positions in the original sequence (the absolute positions), $t^1,\dots,t^M$, and the positions in the alignment (the relative positions) are given by the functions $k^1(t^1),\dots,k^M(t^M)$. We have also assigned sequence weights, $w^1,\dots,w^M$ to the sequences. Let our query sequence be $\mathbf{x}^1$. We first calculate the PLPs, $\mathbf{g}^m$ of each sequence $\mathbf{x}^m$ by using Equation (5). We define the gapped PLP for $\mathbf{x}^m$ with respect to the alignment as

$$\tilde{g}^m_{\lambda,\tau} = \begin{cases} g^m_{\lambda,t}, & \text{if } \exists t : \tau = k^m(t) \\ 0, & \text{if } \nexists t : \tau = k^m(t). \end{cases} \quad (10)$$

So if there is a gap in the sequence at position τ, the gapped PLP is set to 0 for all labels at that position. The average PLP

for the alignment can be calculated as

$$\tilde{g}^*_{\lambda,\tau} = \frac{\sum_{m=1}^{M} w^m \tilde{g}^m_{\lambda,\tau}}{\sum_{\lambda}\sum_{m=1}^{M} w^m \tilde{g}^m_{\lambda,\tau}}. \quad (11)$$

We can reformulate Equation (9) as

$$\hat{A}_{j,t} = \tilde{g}^*_{\Lambda(j),k^1(t)} + \max_{i\in\sigma} \hat{A}_{i,t-1} d_{ij} \quad (12)$$

and, as in the single sequence case, it is possible to find the optimal labeling by the use of backpointers.

3 RESULTS

In order to measure only the performance of the decoding algorithm, we chose to use a pretrained HMM, and not to retrain the HMM during the comparison. We used a recently published HMM, Phobius (Käll *et al*., 2004), which is a combined transmembrane topology and signal peptide predictor. The combination of these features is logical as transmembrane helices often get falsely predicted as signal peptides and vice versa, since both features contain a long hydrophobic stretch.

We have made use of the different cross-validation models and their corresponding test data from the Phobius 10-fold cross validation to measure performance of different HMM decoding algorithms. The default decoder of Phobius is the 1-best algorithm (without homologs).

We compared the 1-best algorithm without (1) and with (2) homologs, a Viterbi decoder without (3) and with (4) homologs, a decoder preceded by parameter reestimation based on the query sequence without (5) and with (6) homologs, and optimal accuracy decoding without (7) and with (8) homologs. The 1-best decoder and the Viterbi decoder with information from homologs that we used are described by (Viklund and Elofsson, 2004). The decoder preceded by parameter reestimation was inspired by Tusnady and Simon (1998), although we used the 1-best algorithm instead of a Viterbi algorithm for the final prediction since it gave better performance (data not shown).

The measurements of the decoder's ability to predict transmembrane topology is shown in Table 1 and signal peptides in Table 2.

We can conclude that optimal accuracy decoding makes significantly better predictions than the other methods when predicting transmembrane topology of the test set containing transmembrane proteins. When considering the sequences with erroneous predictions by the optimal accuracy decoding with homologs, we noted that their alignments contain fewer sequences (on average ∼60) compared with those correctly predicted (on average ∼80). However the parameter reestimation algorithm seems to be the better choice for weeding out soluble proteins.

When predicting signal peptides, the optimal accuracy decoder with homologs shows better Matthew correlation than the other methods, thus indicating that it is the most suitable for the task. We can also see that the 1-best and Viterbi decoders are not helped by information from homologs.

Signal peptide cleavage site predictions made by the alignment based methods, i.e. optimal accuracy with homologs (51% correct), 1-best with homologs (35% correct) and Viterbi with homologs (33% correct) show severely worse results than the other algorithms (all of which have just >70% correct predictions). This is probably because it is harder to pin-point an exact location of a feature when the prediction is based on an average over an alignment.

Table 1. Correct transmembrane topology predictions measured on sets with (TM) and without (non-TM) transmembrane domains by different HMM decoding algorithms with and without homologs numbered (1)–(8) according to the text

No.	Algorithm	Homologs	TM (%)	Non-TM (%)
1	1-best	No	67.8*	97.0
2		Yes	66.1*	97.8**
3	Viterbi	No	59.2*	95.7
4		Yes	57.9*	96.7
5	Parameter re-estimation	No	68.2*	97.2
6		Yes	68.8*	97.8**
7	Optimal accuracy	No	67.1*	95.3*
8		Yes	74.7	97.1

A prediction was counted as correct when all the transmembrane helices overlap the annotated transmembrane helices of the protein over a stretch of at least five residues and the location of the loops were correct. For the proteins not containing transmembrane helices a correct transmembrane topology prediction corresponds to a prediction that does not contain any transmembrane helices.
Figures where the differences to the optimal accuracy decoding with homologs was significant at 99% confidence level were marked with *if they were lower than optimal accuracy decoding and with **if they were higher.

Table 2. Errors in signal peptide prediction on sets with (SP) and without (non-SP) signal peptide by different HMM decoding algorithms numbered (1)–(8) according to the text

No.	Algorithm	Homologs	SP (%)	Non-SP (%)	ρ[a]
1	1-best	No	3.48	3.30	0.901
2		Yes	35.5*	0.67**	0.677
3	Viterbi	No	5.98*	2.77	0.887
4		Yes	40.3*	0.60**	0.641
5	Parameter re-estimation	No	3.56	3.22	0.902
6		Yes	4.39	2.70	0.904
7	Optimal accuracy	No	2.73	5.25*	0.872
8		Yes	3.41	2.32	0.921

The position of cleavage site was not taken in account.
[a]The Matthews correlation coefficient is defined as $\rho = (N_{tp}N_{tn} - N_{fp}N_{fn})/\sqrt{(N_{tp}+N_{fp})(N_{tp}+N_{fn})(N_{tn}+N_{fp})(N_{tn}+N_{fn})}$, where $N_{\{t,f\}\{p,n\}}$ denotes the number of {true,false} {positive,negative} signal peptide predictions.
Figures where the differences to the optimal accuracy decoding with homologs was significant at 99% confidence level were marked with *if they were higher than optimal accuracy decoding and a **if they were lower.

4 DISCUSSION

We have described a new way to incorporate information from homologs when decoding HMMs. For the Phobius model, the decoder increases the transmembrane topology prediction performance as well as the ability to predict signal peptides. It is reasonable to expect performance increases in other application areas as well. A strength of the algorithm is that it enables the signals from feature lengths of each individual homolog to have an impact on the prediction. The method works well with HMMs trained for single sequence usage.

Our evaluation shows that even though the best decoders are helped by information from homologs, not all show an increase in performance. It makes a difference how homologs are incorporated in the decoding process. Other approaches for incorporating homologs in feature prediction with HMMs (Viklund and Elofsson, 2004), use homologs in their training procedure and not just in the decoding. This is probably the major origin of the performance increase they report. It is likely that Phobius as well would gain substantially in performance from incorporating homologs in the training procedure.

Here we have tried to isolate the effect of choice of decoding procedure by using the same HMM and test data for all decoding methods. However, the choice of architecture and parameter estimation procedure could have an effect on the performance of the decoder. For instance, if the architecture contains structures where many paths result in the same labeling (as Phobius does) the Viterbi algorithm is less suitable. One could also argue that if the Viterbi training procedure had been used when estimating the parameters of Phobius we would have obtained better results for the Viterbi decoder. However, if we were to retrain the model for each decoding principle it would be even harder to tell if a difference in performance stems from training or from the choice of decoding algorithm. Other benchmarks (Viklund and Elofsson, 2004; Chen *et al*., 2002) have found an increase in performance when using the parameter reestimation procedure with homologs as opposed to using it without homologs for the HMMTOP (Tusnady and Simon, 1998) architecture. We do not find such a difference for the Phobius architecture. This could be because the Phobius model has more free parameters than the HMMTOP model.

Our method is dependent on a high quality global multiple sequence alignment. At first glance one might think that one would be better-off using a local alignment, instead of a global. After all, the features we are looking for are local in their nature. However, global alignment programs generally perform better than local methods, except in the presence of large N-terminal/C-terminal extensions or large internal insertions (Thompson *et al*., 1999; Lassmann and Sonnhammer, 2002), which we hopefully remove with our requirements on the lengths of blast hits.

The approach of using a multiple sequence alignment is better for predicting features spanning over many amino acids rather than single amino acid features. In our benchmark, we noticed a decrease in accuracy of predictions of the exact location of signal peptide cleavage sites when using homologs in the optimal accuracy decoder compared with those not including them. This is easy to understand as the information of an exact position of a feature spanning a single amino acid will be diluted through a multiple sequence alignment. It could as well be questioned if the exact location of a feature is conserved throughout evolution.

A commonly stressed fact is that the training data of a machine learning method are of crucial importance. Without wanting to diminish that fact, we would like to add that in the case of HMMs it is of high importance to choose a good architecture of the HMM, a good training methods and as we have shown here, a good methodology to decode sequences.

5 MATERIALS AND METHODS

5.1 Test sets

For our measurements of prediction accuracy we used the four datasets described in Käll *et al*. (2004). In brief, they consist of two sets of transmembrane proteins with known topology with (45 sequences) and without (247 sequences) signal peptides, and two sets of soluble proteins with (1275 sequences) and without (1087 sequences) signal peptides. We merged these sets in two different ways for testing accuracy of transmembrane topology prediction and signal peptides separately. We consequently obtained four different test sets:

- 292 sequences from transmembrane proteins in a 'TM' set.
- 2362 sequences from soluble proteins in a 'non-TM' set.
- 1320 sequences with signal peptides in a 'SP' set.
- 1334 sequences without signal peptide in a 'non-SP' set.

The original division into different cross-validation sets was maintained.

5.2 Homology searches and multiple sequence alignments

For each sequence in the test sets, we searched for homologs in Uniprot/TrEMBL with blast, using an E-value cutoff of 10^{-5}. To reduce the number of fragments as well, we included only hits covering at least 75% of the length of the query sequence and 75% of the subject sequence. We re-retrieved the full length sequences of the hits and thereafter aligned them with a global multiple sequence alignment method, Kalign (T. Lassman and E.L. Sonnhammer, Submitted for publication). Sequence weighting was done according to Henikoff and Durbin (1998). However, since the method does not account for gaps, we made some changes: gaps were ignored during the weight calculations, and the weights were divided by the sequence length.

5.3 Other decoders

The implementation of 1-best with homologs follows the description in Viklund and Elofsson (2004). However, we made some modifications to the original algorithm. We weighted each sequence according to the sequence scheme mentioned above. So if in a column of an alignment from sequences with weights $w^1, \ldots, w^M$, we observe the amino acids $x^1, \ldots, x^{M-G}$ and gaps in sequence $(M - G + 1), \ldots, M$ the 'emission score' from state i will be $\left(\prod_{m=1}^{M-G} (e_{ix^m})^{w^m}\right)^{1/(\sum_{m=1}^{M-G} w^m)}$. Instead of building a pseudomultiple alignment from blast hits, a global multiple alignment was used as well. Both these modifications gave small improvements in prediction accuracies (data not shown). We used the same 'emission score' in the implementation of the Viterbi algorithm with homologs.

The parameter reestimation algorithm we implemented follows the main idea in Tusnady and Simon (1998). In the Baum–Welch parameter reestimation procedure, we obtained regularizers by multiplying the emission probabilities in the initial model by 10 000 (in accordance with HMMTOP 2.1) and the transition probabilities by 1000 (arbitrary selected value). Instead of using the Viterbi algorithm for the final prediction we used the 1-best algorithm, since it showed better performance (data not shown).

5.4 Significance tests

To determine whether a difference between two decoding algorithms were significant or not, we used a paired Student's t-test (Michel, 1997). The differences Δ_k in the number of errors made by the algorithms were measured for each of the 10 cross-validation sets separately. The mean difference in errors $\bar{\Delta}$ was calculated. Under the assumption that the binomial distributions of the number of errors made can be approximated with a normal distribution (which is a good approximation for cross-validation sets of >30 samples approximately) we can calculate a Z% confidence interval of the difference in error rate

$$\Delta = \bar{\Delta} \pm t_{Z,K-1} \sqrt{\frac{1}{K(K-1)} \sum_{k=1}^{K} (\bar{\Delta} - \Delta_k)}, \qquad (13)$$

where $t_{Z,K-1}$ is the distribution function of a t-distribution with $K - 1$ degrees of freedom and the number of cross-validation sets $K = 10$ in our setting. The two methods were considered to perform significantly different if zero was not included in the 99% confidence interval.

Significance level of differences between Matthews correlation figures was not considered.

5.5 Implementation

The algorithm was implemented in Java using the BioJava package.

ACKNOWLEDGEMENTS

This work was supported by grants from Pfizer Corporation and the Swedish Knowledge Foundation. We wish to thank Håkan Viklund and Timo Lassmann for the helpful discussions.

REFERENCES

Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.

Bystroff,C., Thorsson,V. and Baker,D. (2000) HMMSTR: a hidden Markov model for local sequence–structure correlations in proteins. *J. Mol. Biol.*, **301**, 173–190.

Chen,C.P., Kernytsky,A. and Rost,B. (2002) Transmembrane helix predictions revisited. *Protein Sci.*, **11**(12), 2774–2791.

Delorenzi,M. and Speed,T. (2002) An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics*, **18**(4), 617–625.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**(9), 755–763.

Edgar,R.C. and Sjölander,K. (2004) COACH: profile-profile alignment of protein families using hidden Markov models. *Bioinformatics*, **20**(8), 1309–1318.

Fariselli,P., Martelli,P.L. and Casadio,R. (2005) The posterior-viterbi: a new decoding algorithm for hidden markov models. Quant. Biol. http://arxiv.org/abs/q-bio.BM/0501006.

Henikoff,S. and Henikoff,J.G. (1994) Position-based sequence weights. *J. Mol. Biol.*, **243**(4), 574–578.

Holmes,I. and Durbin,R. (1998) Dynamic programming alignment accuracy. *J. Comput. Biol.*, **5**(3), 493–504.

Käll,L., Krogh,A. and Sonnhammer,E.L. (2004) A combined transmembrane topology and signal peptide prediction method. *J. Mol. Biol.*, **338**(5), 1027–1036.

Krogh,A. (1994) Hidden Markov models for labeled sequences. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition.* Los Alamitos, CA, October 1994. IEEE Computer Society Press, pp. 140–144.

Krogh,A., Brown,M., Mian,I.S., Sjölander,K. and Haussler,D. (1994a) Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol.*, **235**(5), 1501–1531.

Krogh,A., Mian,I.S. and Haussler,D. (1994b) A hidden markov model that finds genes in *E.coli* dna. *Nucleic. Acids Res.*, **22**(22), 4768–4778.

Lassmann,T. and Sonnhammer,E.L. (2002) Quality assessment of multiple alignment programs. *FEBS Lett.*, **529**(1), 126–130.

Martelli,P.L., Fariselli,P., Krogh,A. and Casadio,R. (2002) A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins. *Bioinformatics*, **18** (Suppl 1), S46–S53.

Michel,T.M. (1997) *Machine Learning.* McGraw-Hill Book Co., Singapore.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**(3), 443–453.

Nielsen,H. and Krogh,A. (1998) Prediction of signal peptides and signal anchors by a hidden Markov model. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **6**, 122–130.

Schwartz,R. and Chow,Y. (1990) The N-best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proceedings of ICASSP 1990*, Albuquerque, NM, IEEE, pp. 81–84.

Söding,J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **21**, 951–960.

Sonnhammer,E.L., von Heijne,G. and Krogh,A. (1998) A hidden markov model for predicting transmembrane helices in protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **6**, 175–182.

Thompson,J.D., Plewniak,F. and Poch,O. (1999) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, **27**(13), 2682–2690.

Tusnady,G.E. and Simon,I. (1998) Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *J. Mol. Biol.*, **283**(2), 489–506.

Viklund,H. and Elofsson,A. (2004) Best alpha-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information. *Protein Sci.*, **3**(7), 1908–1917.