BMC Bioinformatics

BioMed Central

Research article

Open Access

A new decoding algorithm for hidden Markov models improves the prediction of the topology of all-beta membrane proteins

Piero Fariselli*, Pier Luigi Martelli and Rita Casadio

Address: Department of Biology, University of Bologna via Irnerio 42, 40126 Bologna, Italy

Email: Piero Fariselli* - piero.fariselli@unibo.it; Pier Luigi Martelli - gigi@biocomp.unibo.it; Rita Casadio - casadio@alma.unibo.it

* Corresponding author

from Italian Society of Bioinformatics (BITS): Annual Meeting 2005
Milan, Italy, 17–19 March 2005

Published: 1 December 2005

BMC Bioinformatics 2005, **6**(Suppl 4):S12 doi:10.1186/1471-2105-6-S4-S12

Abstract

Background: Structure prediction of membrane proteins is still a challenging computational problem. Hidden Markov models (HMM) have been successfully applied to the problem of predicting membrane protein topology. In a predictive task, the HMM is endowed with a *decoding algorithm* in order to assign the most probable state path, and in turn the labels, to an unknown sequence. The Viterbi and the posterior decoding algorithms are the most common. The former is very efficient when one path dominates, while the latter, even though does not guarantee to preserve the HMM grammar, is more effective when several concurring paths have similar probabilities. A third good alternative is 1-best, which was shown to perform equal or better than Viterbi.

Results: In this paper we introduce the posterior-Viterbi (PV) a new decoding which combines the posterior and Viterbi algorithms. PV is a two step process: first the posterior probability of each state is computed and then the best posterior allowed path through the model is evaluated by a Viterbi algorithm.

Conclusion: We show that PV decoding performs better than other algorithms when tested on the problem of the prediction of the topology of beta-barrel membrane proteins.

Background

All-beta membrane proteins constitute a well structurally conserved class of proteins, that span the outer membrane of Gram-negative bacteria with a barrel-like structure. In all cases known so far with atomic resolution, the barrel consists of an even number of anti-parallel beta strands, whose number ranges from 8 to 22 strands, depending on the protein and/or its functional role [1,2]. In eukaryotes, it is known that similar architectures are present in the outer membrane of chloroplasts and mitochondria, although so far none of the so-called "porins", mainly acting as Voltage Dependent Anion Channels (VDAC), have been solved with atomic resolution ([3] and references therein). It is therefore urgent to devise methods for the prediction of the topology of this class of membrane proteins. Indeed the correct prediction of the protein topology, given the conservation of the barrel architecture may greatly help in threading procedures, especially when sequence homology is low. Furthermore reliable methods, endowed with a low rate of false positives, can also help in genome annotation on the basis of protein structure prediction [3,4]. The problem of predicting beta barrel membrane proteins has been recently addressed with machine learning approaches, and among them Hidden Markov Models (HMMs) have been shown to outperform previously existing methods [5]. HMMs were developed for alignments [6,7], pattern detection [8,9] and also for predictions, as in the case of the topology of all-alpha and

Table 1: Q_{ok} accuracy obtained with the four different decoding algorithms

Proteins	Viterbi	1-best	posterior	posterior-Viterbi
cross-validation				
1a0spTOT	-	-	-	OK
1bxwaTOT	-	OK	OK	OK
1e54	-	-	OK	OK
1ek9aTOT	-	-	OK	OK
1fcpaTOT	-	-	-	-
1fepTOT	-	-	-	OK
1i78a	-	-	OK	OK
1k24	-	-	-	OK
1kmoaTOT	-	-	OK	OK
1prn	-	-	-	-
1qd5a	-	-	OK	OK
1qj8a	-	-	OK	OK
2mpra	-	-	OK	OK
2omf	-	-	OK	OK
2por	-	-	-	-
$<Q_{ok}>$	0.0	0.07	0.60	0.80
blind-test				
1mm4	-	-	OK	-
1nqf	-	-	-	OK
1p4t	OK	OK	OK	OK
1uyn	-	-	-	OK
1t16	-	-	-	-
$<Q_{ok}>$	0.20	0.20	0.40	0.60

all-beta membrane proteins [10-17]. When HMMs are implemented for predicting a given feature, a *decoding* algorithm is needed. With decoding we refer to the assignment of a path through the HMM states (which is the best under *a suitable measure*) given an observed sequence *O*. In this way, we can also assign a label to each sequence element [18,19]. More generally, as stated in [20], the decoding is the prediction of the labelsof an unknownpath. The most famous decoding procedure is the Viterbi algorithm, which finds the most probable allowed path through the HMM model. Viterbi decoding is particularly effective when there is a *single best path* among others much less probable. When several paths have similar probabilities, the posterior decoding or the 1-best algorithms are more convenient [20]. The posterior decoding assigns the state path on the basis of the posterior probability, although the selected path might be not allowed.

In this paper we address the problem of preserving the automaton grammar and concomitantly exploiting the posterior probabilities, without the need of the post-processing algorithm [12,21]. Prompted by this, we design a new decoding algorithm, the posterior-Viterbidecoding (PV), which preserves the automaton grammars and at thesame time exploits the posterior probabilities. A related idea, that is specific for pairwise alignments was previously introduced to improve the sequence alignment accuracy [22]. We show that PV performs better than the other algorithms when we test it on the problem of predicting the topology of beta-barrel membrane proteins.

Results and Discussion

Testing the decoding algorithms on all-beta membrane proteins

In order to test our decoding algorithm on real biological data, we used a previously developed HMM, devised for the prediction of the topology of beta-barrel membrane proteins [12]. The hidden Markov model is a sequence-profile-based HMM and takes advantage of emitting vectors instead of symbols, as described in [12].

Since the previously designed and trained HMM [12] emits profile vectors, sequence profiles have been computed from the alignments as derived with PSI-BLAST [23] on the non-redundant database of protein sequences ftp://ftp.ncbi.nlm.nih.gov/blast/db/.

The results obtained using the four different decoding algorithms are shown in Table 1, where the performance is tested with a leave-one-out cross validation procedure for the first 15 proteins and as blind-test for the latter 5 (see Methods). It is evident that for the problem at hand the Viterbi decoding and the 1-best are unreliable, since only one of the proteins is correctly assigned. In this case the posterior decoding is more efficient and can correctly assign 60% and 40% of the proteins, in cross-validation and on the blind set, respectively. Here the posterior decoding is used without MaxSubSeq, introduced before to recast the grammar [12].

From Table 1 it evident that the new PV decoding is the best performing decoding achieving 80% and 60% accuracy in cross-validation and on the blind set, respectively. This is done ensuring that predictions are consistent with the designed automaton grammar.

Comparison with other available HMMs

In Table 2 we show the results of the comparison between our HMM-decoding with those obtained from the available web servers, based on similar approaches [16,17,21]. The pred-tmbb server [16] allows the user to test three different algorithms (namely Viterbi, 1-best and posterior). Differently from us they find that their HMM does not show significant differences among the three decoding algorithms. This dissimilar behaviour may be due to several concurring facts: i) the different HMM models, ii) pred-tmbb runs on a single-sequence input, iii) pred-tmbb is trained using the Conditional Maximum Likelihood [24]. The second server PROFtmb [17] is based on a method that exploits multiple sequence information and posterior probabilities. Their decoding is related to the

Table 2: PV accuracy compared with other algorithms and HMM models

Method	Q_2	SOV	SOV(BetaTM)	SOV(Loop)	Q_{ok}
cross-validation					
Posterior-Viterbi[1]	0.82	0.87	0.92	0.81	0.80
Viterbi[1]	0.63	0.33	0.27	0.35	0.0
1-best[1]	0.65	0.41	0.37	0.41	0.07
HMMB2HTMR[2]	0.83	0.87	0.88	0.84	0.73
PROFTmb[3]	0.83	0.87	0.88	0.84	0.73
pred-tmbb[4] (Viterbi)	0.78	0.83	0.81	0.82	0.60
pred-tmbb[4] (1-best)	0.78	0.83	0.81	0.82	0.60
pred-tmbb[4] (posterior)	0.78	0.82	0.80	0.82	0.60
blind-test					
Posterior-Viterbi[1]	0.80	0.81	0.84	0.74	0.60
Viterbi[1]	0.62	0.38	0.35	0.40	0.20
1-best[1]	0.63	0.38	0.36	0.40	0.20
HMMB2HTMR[2]	0.80	0.81	0.84	0.74	0.60
PROFTmb[3]	0.72	0.65	0.72	0.58	0.40
pred-tmbb[4] (Viterbi)	0.71	0.73	0.79	0.71	0.20
pred-tmbb[4] (1-best)	0.71	0.73	0.79	0.71	0.20
pred-tmbb[4] (posterior)	0.72	0.75	0.81	0.71	0.20

1 Model taken from Martelli et al., 2002 [12]
2 Fariselli et al. 2003 [21]
3 Bigelow et al., 2004 [17]
4 Bagos et al., 2004 [16]

posterior-Viterbi; however, in their algorithm the authors first obtained the posterior sum contracted into two possible labeling (inner/outer loops and transmembrane as we did in [12]), then they made use of the explicit value of the HMM transition probabilities ($a_{i,j}$). In this way they count the transition probabilities twice (implicitly in the posterior-probability and directly into their algorithm) and the PROFtmb performance is not very different from ours.

Finally, the third server **HMMB2TMR** [21] achieves a performance quite similar to that obtained with PV decoding. To do that **HMMB2TMR** takes advantage of the MaxSubSeq algorithm on top of the posterior sum decoding. However, although MaxSubSeq is a very general two-class segment optimization algorithm, it is a post processing procedure that has to be applied after a HMM decoding. On the contrary, PV is a general decoding algorithm and it is more useful when the underlying predictor is a HMM, where more than two labels and different constraints can be introduced into the automaton grammars.

Conclusion

The new PV decoding algorithm is more convenient in that it overcomes the difficulties of introducing a problem-dependent optimization algorithm when the automaton grammar is to be re-cast. When one-state-path dominates we may expect that PV does not perform better than the other decoding algorithms, and in these cases the 1-best is preferred [20]. Nevertheless, we show that when several concurring paths are present, as in the case of our beta-barrel HMM, PV performs better than the others. Although PV takes more time than other algorithms (the posterior + the Viterbi time), the PV asymptotic computational time-complexity still remains $O(N^2 \cdot L)$ (where L and N are the protein length and the number of states, respectively) as for the other decodings. As far as the memory requirement is concerned, PV needs the same space-complexity of the Viterbi and posterior ($O(N \cdot L)$), while 1-*best* in the average case requires less memory, and can also be reduced [20]. When computational speed is an issue, Viterbi algorithm is the fastest and the time complexity order is $time(viterbi) \leq time(1\text{-}best) \leq time(PV)$. Finally, PV satisfies any HMM grammar structures, including automata containing silent states, and it is applicable to all the possible HMM models with an arbitrary number of labels and without having to work out a problem-dependent optimization algorithm.

Methods

The hidden Markov model definitions

For sake of clarity and compactness, in what follows we make use of explicit BEGIN (B) and END states and we do not treat the case of the silent (null) states. However, their inclusion in the algorithms is only a technical matter and can be done following the prescriptions indicated in [18,19].

An observed sequence of length L is indicated as O (= $O_1 ... O_L$) both for a single-symbol-sequence (as in the

standard HMMs) or for a vector-sequence as described before [12]. $\lambda(s)$ indicates the label associated to the state s, while Λ (= Λ_i,... Λ_L) is the list of the labels associated to each sequence position i obtained after the application of a decoding algorithm. Depending on the problem at hand, the labels may identify transmembrane regions, loops, secondary structures of proteins, coding/non coding regions, intergenic regions, etc. A HMM consisting of N states (indicated below with s and k) is therefore defined by three probability distributions:

Starting probabilities

$$a_{B,k} = P(k|B) \quad (1)$$

Transition probabilities

$$a_{k,s} = P(s|k) \quad (2)$$

Emission probabilities

$$e_k(O_i) = P(O_i|k) \quad (3)$$

The forward probability is

$$f_k(i) = P(O_1,O_2...O_i,\pi_i = k) \quad (4)$$

which is the probability of having emitted the first partial sequence up to position i ending at state k. The backward probability is:

$$b_k(i) = P(O_{i+1},... O_{L-1}, O_L|\pi_i = k) \quad (5)$$

which is the probability of having emitted the sequence starting from the last element back to the (i+1)th element, given that we end at position i in state k. The probability of emitting the whole sequence can be computed using either the forward or backward probabilities according to:

$$P(O|M) = f_{END}(L + 1) = b_B(0) \quad (6)$$

Forward and backward probabilities are also necessary for updating the HMM parameters, using the Baum-Welch algorithm [18,19]. Alternatively a gradient-based training algorithm can be applied [18,20].

Viterbi decoding

Viterbi decoding finds the path (π) through the model which has the maximal probability [18,19]. This means that we look for the path which is

$$\pi^v = argmax_{\{\pi\}} P(\pi|O, M) \quad (7)$$

where O(= O_1,... O_L) is the observed sequence of length L and M is the trained HMM model. Since the $P(O|M)$ is independent of a particular path π, Equation 7 is equivalent to

$$\pi^v = argmax_{\{\pi\}} P(\pi, O|M) \quad (8)$$

$P(\pi, O|M)$ can be easily computed as

$$P(\pi, O \mid M) = \prod_{i=1}^{L} a_{\pi(i-1),\pi(i)} e_{\pi(i)}(O_i) \cdot a_{\pi(L),END} \quad (9)$$

where by construction $\pi(0)$ is always the *BEGIN* state (B).

Defining $v_k(i)$ as the probability of the most likely path ending in state k at position i, and $p_i(k)$ as the trace-back pointer, π^v can be obtained running the following dynamic programming algorithm called Viterbi decoding:

• Initialization

$$v_B(0) = 1 \; v_k(0) = 0 \; for \; k \neq B$$

• Recursion

$$v_k(i) = [\max_{\{s\}}(v_s(i-1)a_{s,k})]e_k(O_i)$$

$$p_i(k) = argmax_{\{s\}} v_s(i-1)a_{s,k}$$

• Termination

$$P(O,\pi^v \mid M) = \max_{\{s\}}[v_s(L)a_{s,END}]$$

$$\pi_L^v = argmax_{\{s\}}[v_s(L)a_{s,END}]$$

• Traceback

$$\pi_{i-1}^v = p_i(\pi_i^v) \quad for \quad i = L \ldots 1$$

• Label assignment

$$\Lambda_i = \lambda(\pi_i^v) \quad for \quad i = 1 \ldots L$$

where $\lambda(s)$ is the label associated to the s state.

1-best decoding

The 1-best labeling algorithm described here is Krogh's previously described variant of the N-best decoding [20]. Since there is no exact algorithm for finding the most

probable labeling, 1-best is an approximate algorithm which usually achieves good results in solving this task [20]. Differently from Viterbi, the 1-best algorithm ends when the most probable labeling is computed, so that no trace-back is needed.

For sake of clarity, here we present a redundant description, in which we define H_i as the set of all labeling hypotheses surviving as 1-best for each state s up to sequence position i. In the worst case the number of distinct labeling-hypotheses is equal to the number of states, h_i^s is the current partial labeling hypothesis associated to the s state from the beginning to the i-th sequence position. In general several states may share the same labeling hypothesis. Finally, we use $\oplus$ as the *string concatenation operator*, so that 'AAAA'⊕'B' = 'AAAAB' (the empty string is " and the empty set is ∅). Thus 1-best algorithm can be described as

• Initialization

$v_B(") = 1\ v_k(") = 0\ for\ k \neq B$

$v_k(\lambda(k)) = a_{B,k} \cdot e_k(O_1)$

$H_1 = \{\lambda(k) : a_{B,k} \neq 0\}\ H_i = \varnothing\ for\ i \neq 1$

• Recursion

$$\forall \quad h \in H_i$$

$$v_k(h \oplus \lambda(k)) = [\sum_s v_s(h) \cdot a_{s,k})]e_k(O_{i+1})$$

$$h_{i+1}^k = argmax_{h \in H_i}[\sum_s v_s(h) \cdot a_{s,k}] \oplus \lambda(k)$$

$$H_{i+1} \leftarrow H_{i+1} \cup \left\{h_{i+1}^k\right\}$$

• Termination

$$\Lambda = argmax_{h \in H_L} \sum_s v_s(h) \cdot a_{s,END}$$

With 1-best decoding, we do not need to keep a backtrace matrix since Λ is computed during the forward steps.

Posterior decoding

The posterior decoding finds the path which maximizes the product of the posterior probability of the states [18,19]. Using the usual notation for forward ($f_k(i)$) and backward ($b_k(i)$) we have

$$P(\pi_i = k|O,M) = f_k(i)b_k(i)/P(O|M) \quad (10)$$

The path π^p which maximizes the posterior probability is then computed as

$$\pi_i^p = argmax_{\{s\}} P(\pi_i = s \mid O,M) \quad (11)$$

for $i = 1...\ L$. The corresponding label assignment is

$$\Lambda_i = \lambda(\pi_i^p) \quad for \quad i = 1 \ldots L \quad (12)$$

If we have more than one state sharing the same label, labeling can be improved by summing over the states that share the same label *(posterior sum)*. In this way we can have a path through the model which maximizes the posterior probability of being in a state with *label* λ when emitting the observed sequence element, or more formally:

$$P(label(O_i) = \lambda \mid O,S) = \sum_{\lambda(s)=\lambda} P(\pi_i = s \mid O,M) \quad (13)$$

$$\Lambda_i = argmax_{\{\lambda\}} P(label(O_i) = \lambda \mid O,\ S) \quad (14)$$

where i ranges from 1 to L.

The posterior-decoding drawback is that the state path sequences π^p or Λ may be not feasible paths.

However, this decoding can perform better than Viterbi, when more than one highly probable path exists [18,19]. In this case a post-processing algorithm that recasts the original topological constraints is recommended [21].

In the sequel, if not differently indicated, with the term *posterior* we mean the posterior sum.

Posterior-Viterbi decoding

Posterior-Viterbi decoding is based on the combination of the Viterbi and posterior algorithms. After having computed the posterior probabilities we use a Viterbi algorithm to find the best allowed posterior path through the model. A related idea, specific for pairwise alignments was previously introduced to improve the sequence alignment accuracy [22].

In the PV algorithm, the basic idea is to compute the path π^{PV}

$$\pi^{PV} = argmax_{\{\pi \in A_p\}} \prod_{i=1}^{L} P(\pi_i \mid O,M) \quad (15)$$

where A_p is the set of the allowed paths through the model, and $P(\pi_i|O,M)$ is the *posterior* probability of the state assigned by the path π at position i (as computed in Eq. 10).

Defining a function $\delta^*(s, t)$ equal to 1 if $s \to t$ is an allowed transition of the model M, 0 otherwise; $v_k(i)$ as the probability of the most probable *allowed-posterior* path ending at state k having observed the partial $O_1,... O_i$ and p_i as the trace-back pointer, we can compute the best path π^{PV} using the Viterbi algorithm:

• Initialization

$$v_B(0) = 1 \; v_k(0) = 0 \; for \; k \neq B$$

• Recursion

$$v_k(i) = \max_{\{s\}}[v_s(i-1)\delta^*(s,k)]P(\pi_i = k \mid O,M)]$$

$$p_i(k) = argmax_{\{s\}}[v_s(i-1)\delta^*(s,k)]$$

• Termination

$$P(\pi^{PV} \mid M,O) = max_s[v_s(L)\delta^*(s,END)]$$

$$\pi_L^{PV} = argmax_{\{s\}}[v_s(L)\delta^*(s,END)]$$

• Traceback

$$\pi_{i-1}^{PV} = p_i(\pi_i^{PV}) \quad for \quad i = L \ldots 1$$

• Label assignment

$$\Lambda_i = \lambda(\pi_i^{PV}) \quad for \quad i = 1 \ldots L$$

An alternative approach, that directly maximizes the most probable labelling, is to substitute the posterior probability of a given state $P(\pi_i = k|O, M)$, with the posterior sum $P(label(O_i) = \lambda|O, M)$ (equation 14). In this case all the states that share the same label have the same probability for each sequence position. However, since the performances of this second version are slightly worse we do not show them.

Datasets

The problem of the prediction of the all-beta transmembrane regions is used to test the algorithm on a real data application. In this case we use a set that includes 20 constitutive beta-barrel membrane proteins whose sequences are less than 25% homologous and whose 3D structure have been resolved. The number of beta-strands forming the transmembrane barrel ranges from 2 to 22. Among the 20 proteins, 15 were used to train a circular HMM (described in [12]), and here are tested in cross-validation (1a0sP, 1bxwA, 1e54, 1ek9A, 1fcpA, 1fep, 1i78A, 1k24, 1kmoA, 1prn, 1qd5A, 1qj8A, 2mprA, 2omf, 2por). Since there is no detectable sequence identity among the selected 15 proteins, we adopted a leave-one-out approach for training the HMM and testing it. All the reported results are obtained during the testing phase, and the complete set of results is available at http://www.biocomp.unibo.it/piero/posvit. The other 5 new proteins (1mm4, 1nqf, 1p4t, 1uyn, 1t16) are used as a blind new test. Since our goal is to predict the beta-strands that span the membrane we score the methods using the annotations derived from the PDB files. An alternative approach not addressed here, is to predict the portion of the transmembrane beta-strands in contact with the lipid bilayer. This prediction is however out of the scope of our approach, since in real porins the localization of the beta-strands in contact with the membrane, has been so far estimated by means of different computational methods and assumptions [25].

Measures of accuracy

We used three indices to score the accuracy of the algorithms. The first one is Q_2 which computes the number of correctly assigned labels divided by the total number of observed symbols. Then we use the SOV index [26] to evaluate the segment overlaps. Finally, we also adopt a very stringent measure called Q_{ok}: a prediction is considered correct only if the number of transmembrane segments coincides with the observed one and the corresponding segments have a minimal overlap of m residues [21]. The value m is segment-dependent and for each segment pairs, is computed as

$$m = min\{|seg_{pr}|/2, |seg_{ob}|/2\} \quad (16)$$

where $|seg_{pr}|$ and $|seg_{ob}|$ are the predicted and observed segment lengths, respectively.

List of abbreviations

• HMM: hidden Markov model.

• PV: Posterior-Viterbi.

• *B:* Begin state.

Authors' contributions

PF developed the Posterior-Viterbi algorithm. PLM designed and trained the Hidden Markov Models. RC con-

tributed to the problem. PF, PLM and RC authored the manuscript.

Acknowledgements

We thank Anders Krogh for the help with the 1-best algorithm. This work was partially supported by the BioSapiens Network of Excellence, two grants of the Ministero della Istruzione dellUniversitá e della Ricerca (MIUR) 'Hydrolases from Thermophiles: Structure, Function and Homologous and Heterologous Expression' delivered to R.C. and 'Large scale modelling of proteases' delivered to P.F., a PNR 2001–2003 (FIRB art.8) and a PNR 2003–2007 (FIRB art.8).

References

1. Schulz G: Beta-barrel membrane proteins. *Curr Opin Struct Biol* 2000, 10:443-447.
2. Casadio R, Fariselli P, PL M: **In silico prediction of the structure of membrane proteins: Is it feasible.** *Brief Bioinf* 2003, **4:**341-348.
3. Casadio R, Jacoboni I, Messina A, V DP: **A 3D model of the voltage-dependent anion channel (VDAC).** *FEBS Lett* 2003, **520:**1-7.
4. Casadio R, Fariselli P, Finocchiaro G, Martelli P: **Fishing new proteins in the twilight zone of genomes: The test case of outer membrane proteins in Escherichia coli K12, Escherichia coli O157:H7, and other Gram-negative bacteria.** *Protein Sci* 2003, **11:**1158-1168.
5. Bagos P, Liakopoulos T, SJ H: **Evaluation of methods for predicting the topology of -barrel outer membrane proteins and a consensus prediction method.** *BMC Bioinformatics* 2005, **1:**1-7.
6. Krogh A, Brown M, Mian I, Sjolander K, Haussler D: **Hidden Markov models in computational biology: Applications to protein modeling.** *Journal of Molecular Biology* 1994, **235:**1501-1531.
7. Baldi P, Chauvin Y, Hunkapiller T, McClure M: **Hidden Markov Models of Biological Primary Sequence Information.** *PNAS USA* 1994, **91:**1059-1063.
8. Mamitsuka H: **Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models.** *Proteins* 1998, **33:**460-474.
9. Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy S, Griffiths-Jones S, Howe K, Marshall M, Sonnhammer E: **The Pfam Protein Families Database.** *Nucleic Acids Research* 2002, **30:**276-280.
10. Tusnady G, Simon I: **Principles governing amino acid composition of integral membrane proteins: application to topology prediction.** *J Mol Biol* 1998, **283:**489-506.
11. Krogh A, Larsson B, von Heijne G, Sonnhammer E: **Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes.** *J Mol Biol* 2001, **305:**567-580.
12. Martelli P, Fariselli P, Krogh A, Casadio R: **A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins.** *Bioinformaticsi* 2002, **18:**S46-S53.
13. Martelli P, Fariselli P, Casadio R: **An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins.** *Bioinformatics* 2003, **19:**i205-i211.
14. Liu Q, Zhu Y, Wang B, Li Y: **A HMM-based method to predict the transmembrane regions of beta-barrel membrane proteins.** *Comput Biol Chem* 2003, **27:**69-76.
15. Viklund H, Elofsson A: **Best alpha-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information.** *Protein Sci* 2004, **13:**1908-1917.
16. Bagos P, Liakopoulos T, Spyropoulos I, SJ H: **PRED-TMBB: a web server for predicting the topology of beta-barrel outer membrane proteins.** *Nucleic Acids Res* 2004, **32:**W400-W404.
17. Bigelow H, Petrey D, Liu J, Przybylski D, B R: **Predicting transmembrane beta-barrels in proteomes.** *Nucleic Acids Res* 2004, **32:**2566-2577.
18. Baldi P, Brunak S: *Bioinformatics: the Machine Learning Approach Cambridge: MIT Press*; 2001.
19. Durbin R, Eddy S, Krogh A, Mitchinson G: *Biological sequence analysis: probabilistic models of proteins and nucleic acids Cambridge: Cambridge Univ Press*; 1998.
20. Krogh A: **Two methods for improving performance of a HMM and their application for gene finding.** In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology ISCB, AAAI Press*; 1997:179-186.
21. Fariselli P, Finelli M, Marchignoli D, Martelli P, Rossi I, R C: **MaxSubSeq: an algorithm for segment-length optimization. The case study of the transmembrane spanning segments.** *Bioinformatics* 2003, **19:**500-505.
22. Holmes I, Durbin R: **Dynamic programming alignment accuracy.** *J Comput Biol* 1998, **5:**493-504.
23. Altschul S, Madden T, Schaffer A, Zhang J, Zhang Z, Miller W, DJ L: **Gapped BLAST and PSI-BLAST: A new generation of protein database search programs.** *Nucleic Acid Res* 1997, **25:**3389-3402.
24. Krogh A: **Hidden Markov models for labeled sequences.** In *Proceedings 12th International Conference on Pattern Recognition. Singapore IEEE Comp Soc Press*; 1994:140-144.
25. Tusnady G, Dosztanyi Z, Simon I: **Transmembrane proteins in the Protein Data Bank: identification and classification.** *Bioinformatics* 2004, **20:**2964-2972.
26. Zemla A, Venclovas C, Fidelis K, B R: **A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment.** *Proteins* 1999, **34:**220-223.