
Supporting online material

for:

Predicting transmembrane beta-barrels in proteomes

H Bigelow, D Petrey, J Liu, D Przybylski & Burkhard Rost

Model encoding

Initial model specification. The model is specified with four distinct types of information: labelled training sequences, architecture, label-to-architecture mapping, and architecture-to-emission mapping. In particular, the model consists of the 91 states connected as shown (Fig. 1). States are labelled to indicate both the assignment of tied parameters and the labelling for which the state is valid during training. For example, state Ua has its own parameters, and only has one valid label. In contrast, any of the 9 states ‘I’ (inner loop) or ‘O’ (outer loop) have the same set of tied emission parameters, and they are all valid for the same sequence label. The set of connections to and from each end of the transmembrane strand region allow modelling the variable-length strand overhangs explicitly. Also, the periplasmic loop region is modelled such that if there is a 4- or 5- residue beta-turn, it is explicitly modelled. We made this choice after observing the abundance of these two types of turns in the periplasmic regions of TMBs.

Labelled training sequences. Each sequence is labelled according to a two-state labelling, which is the desired format for actual prediction, and a multi-state labelling, which is necessary to intelligently determine a subset of valid paths to be considered during model training. The labelling reflects 3-D structure knowledge encoded into a set of discrete labels. The two-state labelling in this case is ‘beta/non-beta’. Here we specially define ‘beta’ to be beta-strand residues which are contiguous with a transmembrane beta-strand. In addition, short beta-bulges or non-beta (as classified by DSSP) within a transmembrane beta-strand are considered ‘beta’. The multi-state labelling is done manually by visual inspection of each protein 3-D structure. Both the number and particular choice of labels is

critical for the performance of the model. The labels correspond to subjectively chosen structural states within the 3-D structure, as determined either by the surrounding residues or supposed lipid molecules. To assign a given set of residues the *same* labelling means we expect that state to have a certain preference for amino acid composition. Assignment to different states reflects an expectation that the residue composition of those states will be significantly different, as a result of the energetic constraints of the protein under natural selection. Obviously this procedure is time consuming and subtle. And, it is not a purely structural consideration, but is dictated partly by the considerations of the behaviour of HMMs.

Architecture and validity criterion. The architecture is the set of states and their directed connections. The label-to-architecture mapping defines a validity criterion: a given path through the architecture is valid only if all of its states are valid for the input label sequence (the multi-state labelling discussed above). Consider the following: each label must have at least one designated ‘valid’ architectural state. Otherwise, a sequence containing such a label would have no valid paths, and no prediction could be made for it. On the other hand, each architectural state should be valid for at least one label, otherwise that state would never be used during the training. The mapping provides a way to express some degree of uncertainty in the process of creating a structure-based labelling.

Tying. The fourth piece of information is the grouping of architecture states sharing the same set of emission parameters. It is a many-to-one mapping known as ‘tying’, described in the excellent tutorial by Rabiner (1).

Mathematical derivation of profile-based HMM

Martelli et. al. derive a special case of the probability density function (pdf) for a continuous observation density HMM, as presented in the classic tutorial on HMMs by Rabiner (1), for specific application to amino acid sequence profiles. The Baum-Welch parameter estimation algorithm is presented in its general case in the Rabiner tutorial, and the special case is derived in(2). In his work and the present work, only paths consistent

with the structure-based labelling of the training sequences are considered during the re-estimation procedure, a technique first proposed by Krogh (3). Otherwise, it is identical with the traditional use of the Baum-Welch training procedure. In this work we use the same pdf and training algorithm. We re-present the derivation here in the notation of Rabiner for clarity and ease of reproducibility, including where applicable, the corresponding equation numbers.

An HMM consists of N hidden states $\mathbf{S} \equiv \{S_1, S_2, \dots, S_N\}$ connected by an $N \times N$ matrix of transition probability distributions $\mathbf{A} = \{a_{ij}\}$ connecting state S_i to state S_j , with initial state distribution $\vec{\pi} \equiv \{\pi_i\}$. Here we use as input to the model a sequence profile $\mathbf{O} \equiv \bar{O}_1 \bar{O}_2 \bar{O}_3 \dots \bar{O}_T$ where $\bar{O}_t \equiv \{O_t(A), O_t(C), \dots, O_t(Y)\}$ are the counts of each amino acid at position t , with the restriction that $\sum_k O_t(k) = P$, $1 \leq t \leq T$, with profile size P , alphabet size $A (=20 \text{ amino acids})$ and length of profile T . (time, in Rabiner's notation). The general form of the continuous observation density pdf for state S_j and observation vector \vec{O} given by Rabiner is:

$$b_j(\vec{O}) = \sum_k c_{jk} \Pi[\vec{O}, \mu_{jk}, \mathbf{U}_{jk}], 1 \leq j \leq N \text{ (cf. 49)}$$

where $\Pi[\vec{O}, \mu_{jk}, \mathbf{U}_{jk}]$ is a probability density, and the c_{jk} 's obey the stochastic constraint:

$$\sum_{k=1}^A c_{jk} = 1, 1 \leq j \leq N \text{ (cf. 50a)}$$

$$c_{jk} \geq 0, 1 \leq j \leq N, 1 \leq k \leq A \text{ (cf. 50b)}$$

These constraints ensure that $b_j(\vec{O})$ is also stochastic, given as $\int_{\vec{O}} b_j(\vec{O}) d\vec{O} = 1$ $1 \leq j \leq N$

(cf. 51). Martelli defines the special case of the continuous density pdf as

$$b_j(\vec{O}) \equiv \sum_k c_{jk} \frac{O(k)}{Z}, \text{ in which the normalization factor } Z \text{ is defined as } Z \equiv \int_{\vec{O}} \sum_k c_{jk} O(k).$$

(Martelli 3). They prove in the appendix that $Z = \frac{P^A}{A!} \sum_k c_{jk}$. (Martelli 15). Therefore, Z is

independent of state j , since $\sum_k^A c_{jk} = 1$ for all j , and independent of sequence position t since we've constrained all profile columns to sum to P (as in Martelli).

We now derive the Baum-Welch re-estimation equations with the continuous profile emission function $b_j(\vec{O}) \equiv \sum_k^A c_{jk} \frac{o(k)}{Z}$ in place of the single symbol emission function $b_i(k)$, in parallel with Rabiner's presentation, (equations 18-27, 36-43c) with the aim of making it straightforward for the reader to reproduce the procedure. In the following, λ indicates the complete set of model parameters, $\mathbf{A} \cup \mathbf{B}$. $Q \equiv \{q_t\}$ stands for 'an arbitrary path through the model' and q_t as 'the state of the model at time t '. Q^* indicates a valid path through the model (consistent with the structure-based sequence labelling).

The forward variable is defined as: $\alpha_t(i) \equiv P(\vec{O}_1 \vec{O}_2 \cdots \vec{O}_t, q_t = S_i | \lambda)$ (cf. 18) and derived inductively by:

initialization:
$$\alpha_1(i) = \pi_i b_i(\vec{O}_1), 1 \leq i \leq N \text{ (cf. 19)}$$

induction:
$$\alpha_{t+1}(j) = \sum_{i=1}^N \alpha_t(i) a_{ij} b_j(\vec{O}_{t+1}), 1 \leq t \leq T-1 \text{ (cf. 20)}$$

termination:
$$P(\mathbf{O} | \lambda) = \sum_{i=1}^N \alpha_T(i) \text{ (cf. 21)}$$

Similarly the backward variable is defined as

$$\beta_t(i) \equiv P(\vec{O}_{t+1} \vec{O}_{t+2} \cdots \vec{O}_T, q_t = S_i | \lambda) \text{ (cf. 23)}$$

and derived as:

initialization:
$$\beta_T(i) = 1, 1 \leq i \leq N \text{ (cf. 24)}$$

induction:
$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\vec{O}_{t+1}) \beta_{t+1}(j), t = T-1, T-2, \dots, 1 \text{ (cf. 25)}$$

Next, expected values of emissions and transitions are defined, with the help of some intermediate definitions. First,

$$\gamma_t(j,k,\mathbf{O}) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk} \frac{o^{(k)}}{Z}}{\sum_{k=1}^A c_{jk} \frac{o^{(k)}}{Z}} \right]$$

is the “probability of being in state S_j at time t with the k th mixture component accounting for \vec{O}_t ” (see text p. 267) so that $\sum_{t=1}^T \gamma_t(j,k,\mathbf{O}) =$ expected # of symbols k emitted from state

$$S_j \text{ in sample } \mathbf{O}. \text{ Also, } \xi_t(i,j,\mathbf{O}) \equiv P(q_t = S_i, q_{t+1} = S_j | \mathbf{O}, \lambda) \equiv \frac{P(q_t = S_i, q_{t+1} = S_j, \mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda)} \text{ (cf.}$$

36) derived as:

$$\xi_t(i,j,\mathbf{O}) = \frac{\alpha_t(i)a_{ij}b_j(\vec{O}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(\vec{O}_{t+1})\beta_{t+1}(j)}. \text{ (cf. 37)}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j,\mathbf{O}) = \text{expected \# of transitions from state } S_i \text{ to state } S_j \text{ in sample } \mathbf{O}. \text{ (cf. 39b).}$$

Here we have explicitly shown the dependence of the variables γ and ξ on the training sample \mathbf{O} , in preparation for the use of multiple training samples. Let $\mathbf{D} \equiv \{\mathbf{O}^1, \mathbf{O}^2, \dots, \mathbf{O}^D\}$

be the set of independent training samples. We define $P(\mathbf{O}^d) \equiv \frac{1}{|\mathbf{D}|}$. Then, by Bayes' Theorem, we have that the total expected value of transitions from state S_i to state S_j given

all sequences $\mathbf{O}^d \in \mathbf{D}$ is $\sum_{\mathbf{O}^d \in \mathbf{D}} \sum_{t=1}^{T^d-1} \xi_t(i,j,\mathbf{O}^d)P(\mathbf{O}^d)$ and the total expected value of emissions

of symbol k from state S_j given all sequences $\mathbf{O}^d \in \mathbf{D}$ is

$$\sum_{\mathbf{O}^d \in \mathbf{D}} \sum_{t=1}^{T^d-1} \xi_t(i,j,\mathbf{O}^d)P(\mathbf{O}^d).$$

Using these expansions in the re-estimation equations yields:

$$\bar{a}_{ij} = \frac{\sum_{\mathbf{O}^d \in \mathbf{D}} \sum_{t=1}^{T^d-1} \xi_t(i,j,\mathbf{O}^d)P(\mathbf{O}^d)}{\sum_{i=1}^N \sum_{\mathbf{O}^d \in \mathbf{D}} \sum_{t=1}^{T^d-1} \xi_t(i,j,\mathbf{O}^d)P(\mathbf{O}^d)} \text{ and } \bar{c}_{jk} = \frac{\sum_{\mathbf{O}^d \in \mathbf{D}} \sum_{t=1}^{T^d-1} \gamma_t(j,k,\mathbf{O}^d)P(\mathbf{O}^d)}{\sum_k \sum_{\mathbf{O}^d \in \mathbf{D}} \sum_{t=1}^{T^d-1} \gamma_t(j,k,\mathbf{O}^d)P(\mathbf{O}^d)} \text{ (cf. 109)}$$

This second equation is the continuous observation density, multiple sample re-estimation equation version of Rabiner's equation 110. Finally, the parameters of the model $\{a_{ij}\}$ and $\{c_{jk}\}$ are set to the re-estimated maximum likelihood values $\{\bar{a}_{ij}\}$ and $\{\bar{c}_{jk}\}$. This completes one iteration of the Baum-Welch training procedure and the derivation.

In the actual implementation of these equations, scaled versions of the α and β variables are used, because in general the actual α and β variables are underflow numbers. Fortunately, the expectations γ and ξ are ratios of probabilities involving the α 's and β 's, so they can be equivalently calculated using the scaled variables, since the scaling coefficients cancel out in the re-estimation equations. The scaling procedure is presented in detail elsewhere (4).

In this, as in Martelli's work, we consider only valid paths when computing the α and β variables during training. Valid paths are those consistent with the structure-based labelling of the training profiles. By Baum's theorem, we iterate the re-estimation procedure until the joint probability $\prod_{\mathbf{O}^d \in \mathbf{D}} P(\mathbf{O}^d, \bar{Q}^* | \lambda)$ increases by less than a certain cutoff. In the actual implementation, $\sum_{\mathbf{O}^d \in \mathbf{D}} \log(P(\mathbf{O}^d, \bar{Q}^* | \lambda))$ is computed to avoid underflow errors.

Decoding

Decoding is the procedure in which an unlabelled sequence profile is assigned a predicted label by the trained HMM. We use a combination of a local and global technique which seems to be the same as that used in Martelli. The individual probabilities for each state S_i and position t are known as:

$$\gamma_t(i) \equiv P(q_t = S_i | \mathbf{O}, \lambda) = \sum_k^A \gamma_t(i, k).$$

Since we are interested ultimately in a two-state (transmembrane strand, non-transmembrane strand) labelling, we further calculate

$$P(q_t = \beta\text{-strand} \mid \mathbf{O}, \lambda) = \sum_{S_i \in \beta\text{-strand}} P(q_t = S_i \mid \mathbf{O}, \lambda), 1 \leq t \leq T \text{ and}$$

$$P(q_t \neq \beta\text{-strand} \mid \mathbf{O}, \lambda) = 1 - P(q_t = \beta\text{-strand} \mid \mathbf{O}, \lambda), 1 \leq t \leq T$$

We define a ‘metric’ function to be used in the Viterbi algorithm:

$$g_i(\mathbf{O}, t) \equiv \begin{cases} P(q_t = \beta\text{-strand} \mid \mathbf{O}, \lambda) \forall i \text{ s.t. } S_i = \beta\text{-strand} \\ P(q_t \neq \beta\text{-strand} \mid \mathbf{O}, \lambda) \forall i \text{ s.t. } S_i \neq \beta\text{-strand} \end{cases}$$

Now we have a semi-global ‘metric’ that can be used in the Viterbi algorithm to find an optimal (in some sense) path consistent with the architecture of the model.

Initialization: $\delta_1(i) = \pi_i g_i(\mathbf{O}, 1) \quad 1 \leq i \leq N$ (cf. 32a)

Recursion: $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] g_i(\mathbf{O}, t), \quad 2 \leq t \leq T, 1 \leq j \leq N$ (cf. 33a)

Termination: $p^* = \max_{1 \leq i \leq N} [\delta_T(i)]$ (cf. 34a)

p^* is the path which is the final two-state per-residue prediction.

For whole-protein discrimination we use the log-odds, or ‘bits’ score $\log \left(\frac{P(\mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \text{null})} \right)$

where the null model is a linear Markov chain with emission parameters set equal to the background distribution of the set being scored.

References for 'Supporting online material'

1. Rabiner, L.R. (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc IEEE, 77, 257-286.
2. 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, S46-53.
3. Krogh, A. (1994) Hidden Markov models for labeled sequences. Proceedings of the 12th IAPR International Conference on Pattern Recognition, 140-144.
4. Baldi, P. and Brunak, S. (1998) Bioinformatics: The Machine Learning Approach. The MIT Press.

