

# Beam Methods for the Profile Hidden Markov Model

Sam Blasiak <sup>\*</sup>      Huzefa Rangwala <sup>†</sup>      Kathryn Blackmond Laskey <sup>‡</sup>

## Abstract

The Profile Hidden Markov Model (PHMM) is commonly used to represent biological sequences. We present a method for transforming the Profile HMM into an equivalent standard HMM where each transition is associated with a single emission. Using this transformation, we develop a beam method, which includes a novel variational adaptation of the infinite-HMM beam sampling technique, to create a fast inference algorithm. We evaluate our algorithm on both synthetic data and protein sequence datasets, showing that our beam method can lead to considerable improvements in runtime while maintaining the model’s ability to concisely represent sequences.

**Keywords:** Hidden Markov Model, Beam Search, Profile, Protein Sequence Analysis, Non-parametric

## 1 Introduction

Profile Hidden Markov Models (PHMM)[4] have been useful in bioinformatics applications for analyzing amino acid, DNA, and RNA sequences. A PHMM is a left-to-right HMM that captures position-specific commonalities of amino acids within a group of related proteins. Profile HMMs are often constructed from multiple sequence alignments (MSAs) [3], which are arrangements of amino acid sequences used to identify regions of structural similarity or evolutionary relationships. Profile HMMs can be used in applications such as classifying sequences, querying sequence databases, and constructing multiple sequence alignments for additional sets of sequences.

A significant challenge associated with the PHMM is in developing efficient inference algorithms, especially when the number of hidden states is large. We accomplish this task by observing that the PHMM can be viewed as an equivalent standard HMM under an aggregation of non-emitting hidden states. This equiva-

lent standard HMM, which we call the Geometric Transition HMM, allows us to use beam methods [7, 11, 15], both traditional and novel, to run an approximate version of the forward-backward [12] algorithm. We show experimentally that these beam methods can lead to significant performance gains compared to the standard forward-backward algorithm in PHMMs.

The combined application of the GT-HMM and beam methods constitutes an advance in understanding Profile HMMs. This combination has the potential to increase the computational efficiency of bioinformatics software that currently relies heavily on running the forward-backward algorithm over Profile HMMs.

## 2 Background

The Hidden Markov Model (HMM) explains a sequence of observed symbols by postulating that a hidden state is responsible for generating each observation. In an HMM, the sequence of hidden states has the Markov property – that is, the value of each hidden state depends on past states only through the immediately preceding hidden state. A number of extensions have been proposed to the basic HMM. These include the Bayesian HMM [8], where prior probabilities are added to emission and transition probabilities, and non-parametric HMMs [1, 10], which make no *a priori* assumptions about the number of hidden states.

There are two key requirements for a non-parametric HMM suitable for modeling amino acid sequences: (i) transition probabilities from each hidden state should be concentrated on a small subset of the entire set of possible hidden states; and (ii) transitions out of all hidden states should share a common, countably-infinite set of possible destination states.

A common model for prior probabilities on an unbounded number of latent components is the Dirichlet Process (DP) [6] distribution. Condition (i) can be satisfied by using a DP prior for transition probabilities of an HMM. However, if separate, uncoupled DP distributions are used for transitions out of each hidden state, then with probability 1, the set of possible destination states for transitions out of any two hidden states will be disjoint [1], violating condition (ii).

To ensure that multiple observed symbols share

<sup>\*</sup>Department of Computer Science, George Mason University. Email: sblasiak@gmu.edu

<sup>†</sup>Department of Computer Science, George Mason University. Email: rangwala@cs.gmu.edu

<sup>‡</sup>Department of Systems Engineering and Operations Research, George Mason University. Email: klaskey@gmu.edu

hidden states, Beal et. al. [1] use the Hierarchical Dirichlet Process (HDP) [14]. The HDP models a prior over emission distributions for an unbounded number of hidden states as a set of Dirichlet Processes with a shared base measure given by a root Dirichlet Process. Using this construction allows HDP-HMM to satisfy condition (ii), thereby allowing a sparse set of hidden states to model the observed sequence.

Another infinite HMM is the Stick Breaking HMM (SB-HMM) [10]. The SB-HMM assigns a separate stick-breaking prior [13] to each transition probability. These stick-breaking priors are parametrized by an infinite length vector of concentration parameters, which are each given Gamma hyperpriors to make the number of prior parameters on the model finite. This careful combination of stick-breaking distributions allows each transition probability to be attached to a single set of emission parameters, also fulfilling condition (ii).

**2.1 Profile HMMs** A Profile HMM [4] is an HMM with specific restrictions on transitions and emissions. The model is similar to the Bakis model [12] used in speech recognition in that it is a left-to-right, non-ergodic HMM. Like other left-to-right HMMs, the PHMM’s utility lies in its ability to capture an archetypal sequence or sequence fragment through the emission distributions of a portion of the model’s hidden states. Profile HMMs include three types of hidden states: *Match* states, which describe the archetypal sequence, *Insert* states, which allow the model to account for symbols not included in the archetypal sequence, and *Delete* states, which do not emit a symbol and allow the model to skip a *Match* or *Insert* state.

A sequence of symbols is generated from a PHMM by traversing states in the finite automata shown in Figure 1. The model begins in a designated start state,  $z_0$ , then transitions to the first *Match*, *Insert*, or *Delete* state. If the transition moves to a *Match* or *Insert* state, then a symbol is emitted. Typically, emissions from *Insert* states are evenly distributed across the symbol alphabet, while *Match* state emissions are attuned to symbol frequencies associated with a particular position in the archetypal sequence. If the model transitions to a *Delete* state, no symbol is emitted. From the  $k^{th}$  *Match*, *Insert*, or *Delete* state, denoted respectively as  $(M, k)$ ,  $(I, k)$ , or  $(D, k)$ , the PHMM can move to either *Insert* state  $(I, k)$ , *Match* state  $(M, k + 1)$ , or *Delete* state  $(D, k + 1)$ . The standard PHMM can transition to a separate terminal state only from the  $K^{th}$  *Match*, *Insert*, or *Delete* state.

The joint probability of an observed sequence,  $x_{1:T}$ , and set of hidden states,  $z_{1:|z|}$  is given by

Symbol	Description
$\Sigma$	the set of observed symbols; $ \Sigma $ indicates the number of observed symbols.
$T$	the length of the observed sequence
$t$	indexes a position in a sequence
$K$	the number of base hidden states, where a “base hidden state” can be thought of as an index to the <i>Match</i> columns in a multiple sequence alignment
$k \in \{1 \dots K\}$	indexes a base hidden state in a Profile HMM
$(s, k)$	indexes the current base hidden state, $k$ , as well as the current choice of $s = \{Match, Insert, Delete\}$ , which are indicated in subscripts by the capital letters $\{M, I, D\}$
$x_t \in \{1 \dots  \Sigma \}$	an observed symbol in sequence $n$ at position $t$
$z_{1: z }$ or $\vec{z}$	A sequence of hidden states consisting of pairs $(s \in \{Match, Insert, Delete\}, k \in 1..K)$ . Each sequence of pairs forms a path through the DFA shown in Figure 1. Unlike the standard HMM, due to non-emitting <i>Delete</i> states, the number of hidden states used to generate an observed sequence in the PHMM could be larger than the length of the observed sequence, so we use $ z $ to indicate the length of the sequence of hidden states.
$z_0 = (Match, 0)$	The first hidden state is defined as an observed start state, allowing us to encode the distribution of start transitions inside the main transition matrix.
$n_{(s,k),s'}$	The number of transitions from hidden state $(s, k)$ to one of the following three states: $s' = M \implies (s', k') = (M, k + 1)$ ; $s' = I \implies (s', k') = (I, k)$ ; $s' = D \implies (s', k') = (D, k + 1)$ for a given sequence of hidden states $z_{1: z }$ . Note that the choice of $s'$ uniquely determines the value of $k$ for at the destination state of the transition.
$n_{(s,k),m}$	The number of emissions of symbol $m$ from hidden state $(s, k)$ for a given sequence of hidden states $z_{1: z }$ .
$A$	The transition probability matrix. $A_{(s,k),s'}$ indicates the probability of transitioning from state $(s, k)$ to one of the following three states: $s' = M \implies (s', k') = (M, k + 1)$ ; $s' = I \implies (s', k') = (I, k)$ ; $s' = D \implies (s', k') = (D, k + 1)$ . For instance, $A_{(M,1),D}$ represents the transition probability from hidden state $(M, 1)$ to hidden state $(D, 2)$ .
$B$	The emission probability matrix. $B_{(s,k),m}$ indicates the probability of emitting symbol $m$ from hidden state $(s, k)$ , where $s \in \{M, I\}$ can only be a <i>Match</i> or <i>Insert</i> state. For instance, $B_{(M,1),\text{a}}$ represents the probability of emitting an “a” from state $(M, 1)$ .
$\alpha_{s,s'}$	Dirichlet prior parameters on transition probabilities. $A_{(s,k),s'}$ shares the prior parameter $\alpha_{s,s'}$ for all $k$ .
$\beta_{s,m}$	Dirichlet prior parameters on emission probabilities. $B_{(s,k),m}$ shares the parameter $\beta_{s,m}$ for all $k$ .

Table 1: Parameter definitions for the Profile Hidden Markov Model

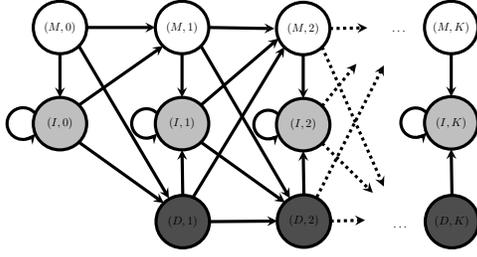


Figure 1: The PHMM’s underlying Deterministic Finite-state Automata (DFA) [12]: *Match* states are represented with a white background, *Insert* states by light gray, and *Delete* states by dark gray. A path through the DFA generates a sequence of observed symbols. In many PHMM constructions, transitions to the final state of the model (not pictured) can occur only from states  $\{(M, K), (I, K), (D, K)\}$ . In the model described in Section 3, any *Match* or *Insert* state can transition to the final state.

$$\begin{aligned}
 & p(x_{1:T}, z_{1:|z|} | A, B) \\
 &= \prod_{(s,k),s'} (A_{(s,k),s'})^{n_{(s,k),s'}} \prod_{(s \in \{M,I\}, k), m} (B_{(s,k),m})^{n_{(s,k),m}}
 \end{aligned} \tag{2.1}$$

where  $A_{(s,k),s'}$  indicates the probability of transitioning from state  $(s, k)$  to the next  $s' \in \{\textit{Match}, \textit{Insert}, \textit{Delete}\}$ , and  $B_{(s,k),m}$  indicates the probability of emitting symbol  $m$  from state  $(s, k)$ . Descriptions of parameters are given in Table 1. The joint probability is similar to the standard HMM, and inference uses the forward-backward algorithm [12]. The forward-backward algorithm computes the sum of all possible hidden state combinations using dynamic programming over a lattice<sup>1</sup> of all possible transitions in the model. Unlike the standard HMM, the PHMM includes non-emitting transitions, which forces us to alter the forward-backward algorithm slightly to use a three-dimensional lattice, shown in Figure 2, rather than the two-dimensional lattice of the standard HMM.

As with the Bayesian HMM, Dirichlet priors can be added to the transition and emission probabilities of the PHMM to produce a model with the following joint probability:

$$\begin{aligned}
 & p(x_{1:T}, z_{1:|z|}, A, B | \alpha, \beta) \\
 &= \prod_{(s,k),s'} (A_{(s,k),s'})^{n_{(s,k),s'}} \prod_{(s \in \{M,I\}, k), m} (B_{(s,k),m})^{n_{(s,k),m}} \\
 & \quad \prod_{(s,k)} \text{Dir}(A_{(s,k),\cdot}; \alpha_{s,s'}) \prod_{(s,k)} \text{Dir}(B_{(s,k),\cdot}; \beta_{s,m})
 \end{aligned} \tag{2.2}$$

<sup>1</sup>We follow [12] in using the term “lattice” to refer to the directed acyclic graph of transitions between hidden states associated with each observed symbol. This graph is also commonly referred to as a trellis in the literature [7].

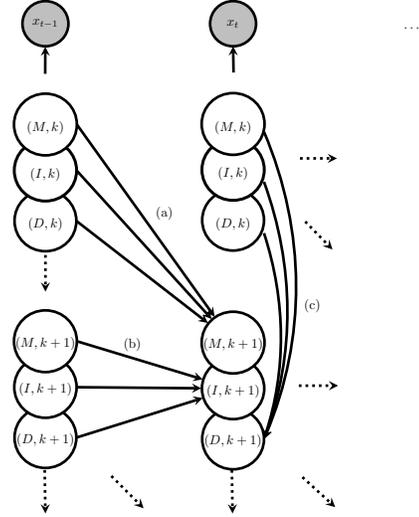


Figure 2: A portion of the three-dimensional lattice used in the PHMM forward algorithm. The horizontal dimension represents elements in the sequence, the vertical dimension values of  $k$ , and the depth dimension the choice of *Match*, *Insert*, or *Delete*. Observed symbols are emitted either by transitioning to a *Match* state (a) or an *Insert* state (b). Transitions to a *Delete* state (c) do not emit a symbol.

### 3 The Infinite Profile HMM

To construct an infinite PHMM, we consider the case where  $K$  is unbounded. In addition, we modify the PHMM so that the model can transition to the end state from any *Match* or *Insert* state (Figure 1). This modification allows the model to emit a sequence of observed symbols without needing to traverse infinitely many *Delete* states. We believe that we are the first to propose an infinite version of the PHMM.

Unlike the HDP-HMM and SB-HMM, we run into difficulties constructing a non-informative prior. An uninformative prior on PHMM transitions places a uniform distribution on all paths through the three-dimensional lattice. In the limit of infinite states, this leads to a Dirichlet prior that places all of its probability mass on *Delete* transitions<sup>2</sup>. Rather than using an uninformative prior in our model, we choose priors that make reasonable assumptions about how amino-acid sequences in a dataset should align for practical purposes. For example, it is unreasonable to expect that a model will use more than  $K = \sum_n |x_n|$  states. With these assumptions in mind, we note that the major difference between the finite and infinite models is that in the finite model we encode our assumptions about

<sup>2</sup>To see this, note that each path that generates an observed sequence passes through a fixed number of *Match* and *Insert* states but can use an arbitrarily large number of *Delete* states.

Symbol	Description
$z_t^{(GT)}$	The hidden state associated with the emitted symbol at position $t$ of the sequence, taking on values from the pair $(k \in 1..K, s \in \{\text{Match, Insert}\})$ , $z_t^{(GT)} \in (s, k)$ ( <i>Delete</i> states are not included in this sequence).
$A_{(s,k),(s',k')}^{(GT)}$	The probability of a sequence of PHMM transitions beginning at state $(s, k)$ ending at state $(s', k')$ , $s, s' \in \{M, I\}$ .
$n_{(s,k),(s',k')}^{(GT)}$	Given a set of hidden states, $z_{1:T}^{(GT)}$ , the total number of transitions in the dataset between hidden states $(s, k)$ and $(s', k')$ .

Table 2: Additional parameter definitions for the GT-HMM

the length of a sequence alignment in the value of  $K$ . In contrast, for the infinite model, our assumptions about the total length of the alignment are encoded in the Dirichlet priors on transition probabilities. This characteristic of the model can be seen visually in the upper right-hand heat map in Figure 7. In the heat map, areas of high intensity under uniform transition and emission probabilities at the start of inference do not reach the model’s truncation threshold,  $K$ . The model should therefore be more flexible in the sense that a small number of hidden states should produce emitted symbols and the number of these states should reflect characteristics of the dataset rather than be determined by a preset model parameter.

We can also show that, under certain conditions, aggregated transition probabilities in the Bayesian PHMM are generated from stick-breaking priors, but due to space considerations, we present this derivation as an addendum<sup>3</sup>. We believe that this is the first time an infinite PHMM has been proposed.

**3.1 The Geometric Transition HMM** To construct a set of efficient inference methods, we present a transformation from the PHMM to an equivalent HMM, which we refer to as the Geometric Transition HMM (GT-HMM) due to the geometrically decreasing transition probabilities from each hidden state. This transformation merges sequences of *Delete* transitions with a single terminating *Match* or *Insert* transition so that every transition becomes associated with an emission. This transformation is similar to existing techniques in speech recognition applications for transforming an HMM with non-emitting states to one that emits after every transition [7]. To the best of our knowledge, we are the first to propose this transformation for PHMMs.

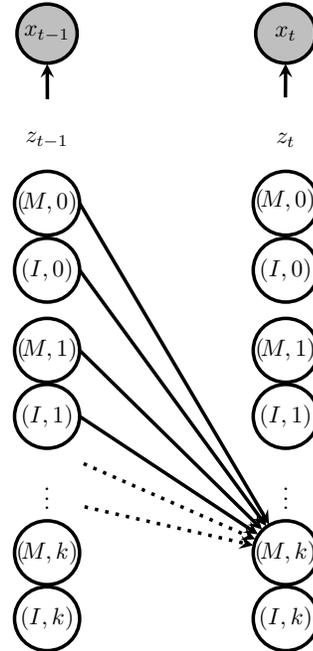


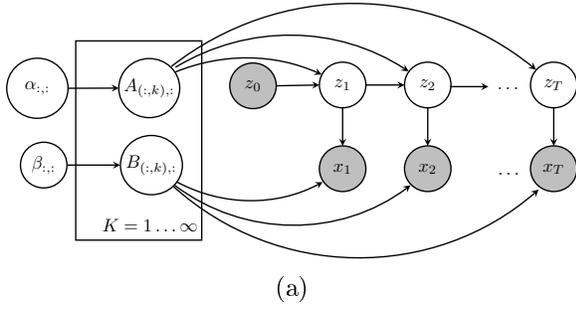
Figure 3: A section of the two-dimensional lattice used for GT-HMM inference showing transitions used to compute the forward recurrence for state  $(M, k)$  for position  $t$  in the sequence. Unlike the PHMM, the GT-HMM no longer includes *Delete* states. In addition, transitions have been added from all states in column  $t - 1$  to states in column  $t$  with larger values of  $k$ .

We illustrate the effect of the transformation from PHMM to GT-HMM in Figure 3, which shows all transitions to hidden state  $(M, k)$  emitting  $x_t$  from the set of previous hidden states that could emit  $x_{t-1}$  in a portion the lattice used for the forward-backward algorithm. The GT-HMM does not contain any non-emitting *Delete* states. Instead, it merges *Delete* states in the PHMM with emitting states, allowing transitions from a much larger number of states in column  $t - 1$ .

The infinite PHMM can be represented by a plate diagram using the GT-HMM as shown in Figure 4a. The generative procedure for the infinite PHMM is given in Figure 4b.

Equation 3.3 shows GT-HMM transition probabilities to *Match* states in terms of the original PHMM transitions, where (a) indicates a sequence of delete transitions followed by a *Match*, (b) indicates a single *Match* transition, and (c) disallows transition to a state with a value of  $k$  less than or equal to that of the current state. Similarly, Equation 3.4 shows GT-HMM transition probabilities to *Insert* states in terms of the original PHMM transitions, where (d) indicates a sequence of *Deletes* followed by an *Insert*, (e) indicates a single *Insert* transition, and (f) disallows transition to a state

<sup>3</sup>[http://cs.gmu.edu/~sblasiak/PHMM\\_stick\\_breaking.pdf](http://cs.gmu.edu/~sblasiak/PHMM_stick_breaking.pdf)



(a)

$$\begin{aligned}
 A_{(s,k),:} &\sim \text{Dir}(\alpha_{s,:}) \\
 B_{(s,k),:} &\sim \text{Dir}(\beta_{s,:}) \\
 z_t &\sim \text{Mult}\left(A_{z_{t-1},(:,,:)}^{(GT)}\right) \\
 x_t &\sim \text{Mult}(B_{z_t,:})
 \end{aligned}$$

(b)

Figure 4: (a) The plate diagram of the infinite PHMM, and (b) the generative process for the infinite PHMM. Note that emission and transition probabilities are from the GT-HMM. We use “:” symbols in subscripts as in Matlab notation, indicating a vector or matrix with all possible values of the replaced parameter.

with a smaller value of  $k$ .

$$A_{(s,k),(M,k')}^{(GT)} = \begin{cases} A_{(s,k),D} \left( \prod_{k''=k+1}^{k'-2} A_{(D,k''),D} \right) A_{(D,k'-1),M} & k < k' - 1 & (a) \\ A_{(s,k),M} & k = k' - 1 & (b) \\ 0 & k \geq k' & (c) \end{cases} \quad (3.3)$$

$$A_{(s,k),(I,k')}^{(GT)} = \begin{cases} A_{(s,k),D} \left( \prod_{k''=k+1}^{k'-1} A_{(D,k''),D} \right) A_{(D,k'),I} & k < k' & (d) \\ A_{(s,k),I} & k = k' & (e) \\ 0 & k > k' & (f) \end{cases} \quad (3.4)$$

The probability of a set of sequences for the GT-HMM is given by the same expression as the standard HMM, but with hidden states parameterized by  $(s, k)$  pairs with  $s \in \{M, I\}$  and  $k \in [1 \dots K]$ :

$$p(x_{1:T}, z_{1:|z|} | A, B) = \prod_{(s,k),(s',k')} \left( A_{(s,k),(s',k')}^{(GT)} \right)^{n_{(s,k),(s',k')}^{(GT)}} \prod_{(s,k),m} (B_{(s,k),m})^{n_{(s,k),m}} \quad (3.5)$$

Because each transition under the GT-HMM results in an emission, we can now consider all possible transitions between hidden states  $(s, k)$  and  $(s', k')$ , which allows forward and backward recursions to be run using the standard two-dimensional lattice (shown in Figure 3 for transitions to a single *Match* state) rather than the three-dimensional lattice commonly used in PHMM inference.

#### ALGORITHM 4.1. Variational Inference

Repeat until the variational bound converges:

1) Compute Expectations

$$E[n_{(s,k),s'}] \text{ from the forward-backward algorithm}$$

$$E[n_{(s,k),m}] \text{ from the forward-backward algorithm}$$

$$E[\log A_{(s,k),s'}] = \Psi(\tilde{\alpha}_{(s,k),s'}) - \Psi\left(\sum_{s''} \tilde{\alpha}_{(s,k),s''}\right)$$

$$E[\log B_{(s,k),m}] = \Psi(\tilde{\beta}_{(s,k),m}) - \Psi\left(\sum_{m'} \tilde{\beta}_{(s,k),m'}\right)$$

2) Maximize with respect to variational parameters

$$\tilde{\alpha}_{(s,k),s'} \leftarrow \alpha_{s,s'} + E[n_{(s,k),s'}]$$

$$\tilde{\beta}_{(s,k),m} \leftarrow \beta_{s,m} + E[n_{(s,k),m}]$$

$$\tilde{A}_{(s,k),s'} \leftarrow \exp(E[\log A_{(s,k),s'}])$$

$$\tilde{B}_{(s,k),m} \leftarrow \exp(E[\log B_{(s,k),m}])$$

Figure 5: Variational Inference algorithm for the infinite PHMM. Parameter definitions are given in Tables 1 and 2. The  $\Psi$  symbol indicates the digamma function:  $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}$ .

In addition, Equations 3.3 and 3.4 indicate that increasing the destination state index  $k'$  while holding the source index  $k$  fixed causes additional  $A_{(D,k),D}$  terms (*Delete-to-Delete* transitions) to be included in  $A_{(s,k),(s',k')}^{(GT)}$ , leading to both an exponential decrease in the GT-HMM transition probability and to a limit of 0 as  $k$  increases indefinitely.

## 4 Inference

We use a structured variational algorithm for inference, factoring the variational distribution as follows:

$$q(\vec{z}, A, B) = q(\vec{z}) \prod_{(s,k)} q(A_{(s,k),:}) \prod_{(s,k)} q(B_{(s,k),:})$$

The update equations for maximizing the variational bound<sup>4</sup> on the marginal likelihood and minimizing the KL divergence between the true posterior and variational posterior are given in Algorithm 4.1 (Figure 5).

Similar structured variational methods for performing HMM inference are described in [8, 10, 2], so we omit the details of our derivation. Care must be taken,

<sup>4</sup>The full expression for the bound is provided in Appendix A.

however, to correctly convert counts of transitions (the sufficient statistic associated with transition probabilities) in the GT-HMM ( $E[n_{(s,k),(s',k')}^{(GT)}]$ ) back to PHMM counts ( $E[n_{(s,k),s'}]$ ), used to update the variational parameter,  $\hat{A}$ :

$$E[n_{(s,k),s'}] = \sum_{(s,k),s' \in (s,k),(s',k')} E[n_{(s,k),(s',k')}^{(GT)}] \quad (4.6)$$

If a PHMM transition is an element of the aggregate GT-HMM transition, then we add this expectation to the expectation under the PHMM.

These expected emission and transition counts, which are normally computed using the forward-backward algorithm, are computed for the GT-HMM using a beam search version of the forward backward algorithm described in the next section.

We use a truncated model for our variational approximation, setting the value of  $K$  to a large, but finite, value. Truncation, of course, transforms the infinite model into a finite model. To ensure that this approximation is accurate, it is critical to set the truncation levels large enough to capture the bulk of the posterior probability mass over the set of hidden states.

**4.1 Beam Methods** Beam methods [7, 11] increase the speed of computation in message passing algorithms by eliminating hidden variable assignments that do not contribute significantly to the expectation or marginal probability. Specifically, we would like to eliminate, without major degradation in accuracy, as many terms as possible from the sum over  $z_{t-1}$  in each forward recurrence,  $p(z_t, x_{1:t}) = \sum_{z_{t-1}} p(x_t|z_t)p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1})$ , where  $z_{t-1}$  indicates the column in the lattice associated with the  $t-1$ <sup>th</sup> observation. Figure 6 shows a beam containing the values of  $z_{t-1}$  that are retained in the sum.

Beam methods are not effective in the standard PHMM because the forward and backward recurrences in the three-dimensional lattice are computed over only three states, *Match*, *Insert*, and *Delete* (Figure 1). In contrast, the forward and backward recurrences in the GT-HMM involve sums over the  $O(K)$  moves ( $K$  is the truncation level of the PHMM), between adjacent columns in the two-dimensional lattice (Figure 3). This allows us to eliminate a larger number of terms from the sum in each recurrence.

Although the GT-HMM transformation has the potential to speed up inference, it may have the opposite effect in certain cases. These cases can be quantified by a straightforward analysis of the number of messages passed for each observed symbol. The forward-backward algorithm on the GT-HMM passes  $O(T \cdot K^2)$

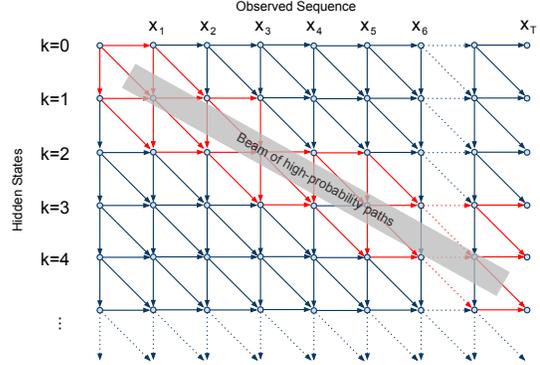


Figure 6: The lattice used for PHMM inference. Observed sequences are generated by paths of hidden states through the graph. Transitions marked in red indicate a potential beam of highly probable paths within the total set of possible paths. *Match*, *Insert*, and *Delete* states are merged for clarity.

messages per sequence, while on the PHMM it passes  $O(T \cdot K(3^2))$  messages. This means that for the beam method to run faster than the standard PHMM forward-backward algorithm, the size of the beam,  $K^{(beam)}$ , (if we assume a constant beam size) must satisfy  $K^{(beam)} < 3\sqrt{K}$ . Therefore, if the number of PHMM hidden states needed to accurately model the set of observed sequences is known to be small beforehand, the standard forward-backward algorithm may be more desirable than our beam methods. However, if the number of hidden states needed to accurately model a dataset is large, then beam methods will be useful in PHMM inference because the size of  $K^{(beam)}$  depends primarily on the prior parameters in the distributions over transition and emission probabilities,  $\alpha$  and  $\beta$ , rather than the truncation level,  $K$ .

In the GT-HMM, we apply the KL divergence beam method from [11]. However, the KL-divergence approach by itself is not enough. We are still required to compute  $O(K)$  forward messages for each lattice column  $t$  even though some these messages are very close to zero. These tiny values result from the combined effect of a small number of forward messages in the beam of column  $t-1$  and exponentially decreasing transition probabilities. For a faster beam method, we can specify, for each column of the lattice, a threshold on the hidden state index  $k$ , and stop computing forward messages when the threshold is exceeded.

We derive this thresholding criterion from an adaptation of the auxiliary variable beam method described by Van Gael et. al. [15]. In the auxiliary variable method, a uniformly distributed auxiliary variable,  $u_t$ , is added to the model. The distribution of  $u_t$  is conditioned on the values of hidden states  $t$  and  $t-1$ , yielding

the joint distribution:

$$p(u_{1:T}, z_{1:T}, x_{1:T} | A, B) = \prod_t p(u_t | z_t, z_{t-1}) p(z_t | z_{t-1}) p(x_t | z_t)$$

$p(u_t | z_t, z_{t-1})$  is chosen so that marginalizing with respect to  $u_{1:T}$  returns the original joint distribution:  $p(u_t | z_t, z_{t-1}) = \frac{\mathbb{I}(u_t < p(z_t | z_{t-1}))}{p(z_t | z_{t-1})}$ , where  $\mathbb{I}(\omega) = 1$  if  $\omega$  is true and 0 otherwise and  $z_t$  indicates the hidden state at position  $t$  in the sequence.

We use the auxiliary variables in conjunction with a variational argument to justify truncating the computation of forward probabilities using a fixed thresholding criterion. With the addition of auxiliary variables, forward messages in the GT-HMM can be expressed as follows:

$$p(z_t, x_{1:t}) = \int_{u_t} \sum_{z_{t-1}} p(x_t | z_t) p(u_t | z_t, z_{t-1}) p(z_t | z_{t-1}) p(z_{t-1}, x_{1:t-1}) \quad (4.7)$$

where  $x_{1:t}$  indicates the sub-sequence of observed symbols from positions 1 to  $t$ .

This formulation allows us to construct the following variational bound:

$$\begin{aligned} \log p(z_t, x_{1:t}) &\geq \int_{u_t} \sum_{z_{t-1}} q_{z_t}(z_{t-1}, u_t) \log \frac{p(x_{1:t}, z_t, z_{t-1}, u_t)}{q_{z_t}(z_{t-1}, u_t)} \\ &= \log p(x_t | z_t) + \\ &\int_{u_t} \sum_{z_{t-1}} q_{z_t}(z_{t-1}, u_t) \log \frac{p(u_t | z_t, z_{t-1}) p(z_t | z_{t-1}) p(z_{t-1}, x_{1:t-1})}{q_{z_t}(z_{t-1}, u_t)} \end{aligned} \quad (4.8)$$

We assume a factorization of  $q_{z_t}(z_{t-1}, u_t) = q_{z_t}(z_{t-1}) q_{z_t}(u_t)$ , which allows us to maximize with respect to the individual factored portions of the distribution. This maximization results in following expressions (derivation in Appendix B):

$$q_{z_t}(z_{t-1}) \propto \mathbb{I}(p(z_t | z_{t-1}) \geq \tilde{\theta}_t) p(z_{t-1}, x_{t-1}) \quad (4.9)$$

$$q_{z_t}(u_t) = \frac{\mathbb{I}(u_t < \tilde{\theta}_t)}{\tilde{\theta}_t} \quad (4.10)$$

where  $\tilde{\theta}_t$  is a variational parameter specifying a truncation threshold for each position in the observed sequence. A set of approximate posteriors,  $q_{z_t}(z_{t-1})$  and  $q_{z_t}(u_t)$ , are computed for the forward recurrence,  $p(z_t, x_{1:t})$ , for each value of  $z_t$ , hence the subscript on the variational distributions.

The distribution  $q_{z_t}(z_{t-1})$  is over  $z_{t-1}$  rather than  $z_t$ . Therefore, it does not make sense to use these quantities directly for computing forward recurrences. Instead, we pretend that we are using  $q_{z_t}(z_{t-1})$  to sample values of  $z_{t-1}$  in the backward direction given that we know the value of  $z_t$ . If  $q_{z_t}(z_{t-1})$  is empty, i.e., there are no transitions to  $z_t$  where  $p(z_t | z_{t-1}) \geq \tilde{\theta}_t$ , then

Name	Hidden States ( $K$ )	Emitted Symbols ( $M$ )	Description
PHMM <sub>M</sub>	20	2	90% probability of a match transition; emission of symbol zero from match states increases as the value of $k$ increases; uniform emissions from insert states
PHMM <sub>I</sub>	20	3	99.9% probability transitioning to an insert state and a 90% probability of remaining in an insert state at the 10th hidden state
PHMM <sub>D</sub>	40	3	47.5% probability of a delete transition and a 47.5% probability of a match transition from the central 20 hidden states; 90% probability of a match transition from all other hidden states

Table 3: Profile HMMs used to generate synthetic datasets.

we would have no way to sample a hidden state given that the value of the current hidden state is  $z_t$ . We therefore use this criterion for truncating computation of forward recurrence values: if transitions from all states in the beam for lattice-column  $t-1$  to the current state  $z_t = (s, k)$  fall below  $\tilde{\theta}_t$ , then we do not compute values of  $p(z_t = (s, k'), x_{1:t})$  for  $k' > k$ .

## 5 Experiments

We evaluated the effectiveness of the beam method using the GT-HMM model, computing both the speed of the inference and test-set perplexities. We compared our approach to the forward-backward algorithm in the standard PHMM. We conducted experiments on a synthetic dataset generated from three PHMMs designed to emphasize different possible latent configurations. The structures of these PHMMs are described in Table 3. We also ran experiments on an ASTRAL<sup>5</sup>-filtered subset of the Structural Classification of Proteins (SCOP) [9] database. The SCOP database categorizes proteins into a multilevel hierarchy that captures commonalities between protein structures at different levels of detail. The ASTRAL compendium provides versions of SCOP datasets filtered to remove significantly similar sequences, allowing for less biased modeling. Tests were run using 5-fold cross validation on a subset of the SCOP 1.75 dataset filtered at 95% identity. This dataset contained classes having between 80 and 200 sequences.

We evaluated the ability of our beam method to increase the speed of inference while maintaining an accuracy (in terms of perplexity) comparable to the standard (no beam) forward-backward algorithm (We refer to the standard PHMM as the “no beam” model in results Tables 4, 5, and 6). For all experiments, we conducted inference using the variational algorithm described in Figure 5. The algorithm was stopped when

<sup>5</sup><http://astral.berkeley.edu/>

either the criterion,  $1 - \frac{B_t}{B_{t-1}} < 10^{-6}$ , was reached, where  $B_t$  indicates the value of the variational bound at iteration  $t$ , or until 300 iterations were performed. We set the prior parameters,  $\alpha$  and  $\beta$ , uniformly to 0.5 and truncated the infinite model by setting  $K$  to be twice the maximum length sequence in the dataset. Learning transitions from *Insert* states tends to drive inference toward bad local minima with multiple chains of insertions. Hence, we fixed the distribution of *Insert* transitions to uniform.

To compute perplexity under the Bayesian model, we approximated  $A$  and  $B$  using the expectations from the variational distributions  $q(A; \tilde{\alpha})$  and  $q(B; \tilde{\beta})$ :  $perp = \exp\left(-\frac{\sum_n \log p(x_n | E_{q(A)}[A], E_{q(B)}[B])}{\sum_n |x_n|}\right)$ . For all experiments, perplexity was computed using the standard (no beam) forward-backward algorithm.

In all experiments, we tested three settings of beam parameters, “narrow,” “medium,” and “wide” with threshold settings of  $\epsilon = [10^{-2}, 10^{-3}, 10^{-4}]$  and  $\tilde{\theta} = [10^{-16}, 10^{-17}, 10^{-18}]$ . The parameter  $\epsilon$  indicates the threshold used for the KL divergence beam and  $\tilde{\theta}$  indicates the threshold used for the auxiliary variable beam. Smaller beam thresholds are associated with larger beams.

**5.1 Synthetic Data** To test the performance of our beam method, we generated 100 training and 25 test sequences from the PHMM models described in Table 3. We then compared PHMMs inferred on the training set using our beam methods against models inferred using the standard forward-backward algorithm by both computing perplexities on test sets and recording the time needed for inference. Table 4 shows the results of these experiments. The beam methods achieved a lower perplexity than the standard forward-backward algorithm, with an average decrease of 17% over all beam thresholds. The beam methods were able to converge to a better optimum by eliminating paths of hidden states that were not useful for inference. The results confirmed our expectations that beam methods improve runtime, with an average improvement of 84% over all of the thresholds tested (Table 4a). As the threshold values increased, more hidden states were included in the beam, and runtime increased by an average of 13% from the narrow to the medium beam and 9.5% from the medium to the wide beam (Table 4b).

To show that our beam methods capture similar characteristics of the data as standard forward-backward inference, we present a series of heat maps. Cells in each heat map (Figure 7) associate expectations of the number of paths through hidden states to

Synthetic Data: Test Set Perplexities				
Model/ SCOP Category	Beam			No Beam
	Narrow	Medium	Wide	
PHMM <sub>M</sub>	1.72	1.71	1.71	2.15
PHMM <sub>I</sub>	2.27	2.21	2.22	2.65
PHMM <sub>D</sub>	2.34	2.33	2.33	2.74

(a)

Synthetic Data: Inference Times (seconds)				
Model/ SCOP Category	Beam			No Beam
	Narrow	Medium	Wide	
PHMM <sub>M</sub>	11.36	22.98	35.31	101.11
PHMM <sub>I</sub>	3.95	38.91	47.86	280
PHMM <sub>D</sub>	9.2	36.09	54.96	245.57

(b)

Table 4: The charts above show a comparison of test-set perplexities (a) and run times (b) between inference using our beam methods and the standard (no beam) forward-backward method on sets of synthetic datasets. “Narrow,” “medium,” and “wide” indicate threshold settings of  $\epsilon = [10^{-2}, 10^{-3}, 10^{-4}]$  and  $\tilde{\theta} = [10^{-16}, 10^{-17}, 10^{-18}]$ , where  $\epsilon$  indicates the KL divergence beam threshold and  $\tilde{\theta}$  indicates the auxiliary variable threshold.

intensity in a two-dimensional grid - i.e., a cell  $(k, t)$  in the heat map is the sum expectations that a path of hidden states passing through either  $(M, k)$  or  $(I, k)$  generates symbol  $x_t$ . The heat maps depict expectations for the longest sequence in the training set generated by PHMM<sub>I</sub> collected at intervals of 20 steps during inference. We created heat maps for each beam threshold setting and for the no-beam experiment. For all levels of the beam threshold, the configuration of hidden state expectations nearly matches the configuration in the no-beam setting. The areas of red in the beam-inference heat maps indicate hidden states that were excluded from inference. Darker colors in the heat maps indicate larger values. Comparing areas of red in the graphs shows how lower beam thresholds consider larger numbers of hidden states.

## 5.2 SCOP Datasets: Uniform Initialization

To compare the infinite PHMM’s performance on the SCOP dataset, we ran experiments using 5-fold cross validation on each SCOP category. Models were initialized using uniform expectations of transition and emission probabilities, i.e. we initialize variational parameters  $\hat{A}_{(s,k),s'} = \frac{1}{3}$  (standard PHMM) and  $\hat{B}_{(s,k),m} = \frac{1}{|\Sigma|}$ . Results on the SCOP datasets (Table 5) show a similar pattern as in the synthetic experiments: in comparison to the no-beam method, the beam method improved both perplexity (3.51% for the narrow beam, 4.56% for the medium beam, and 4.75% for wide beam averaged

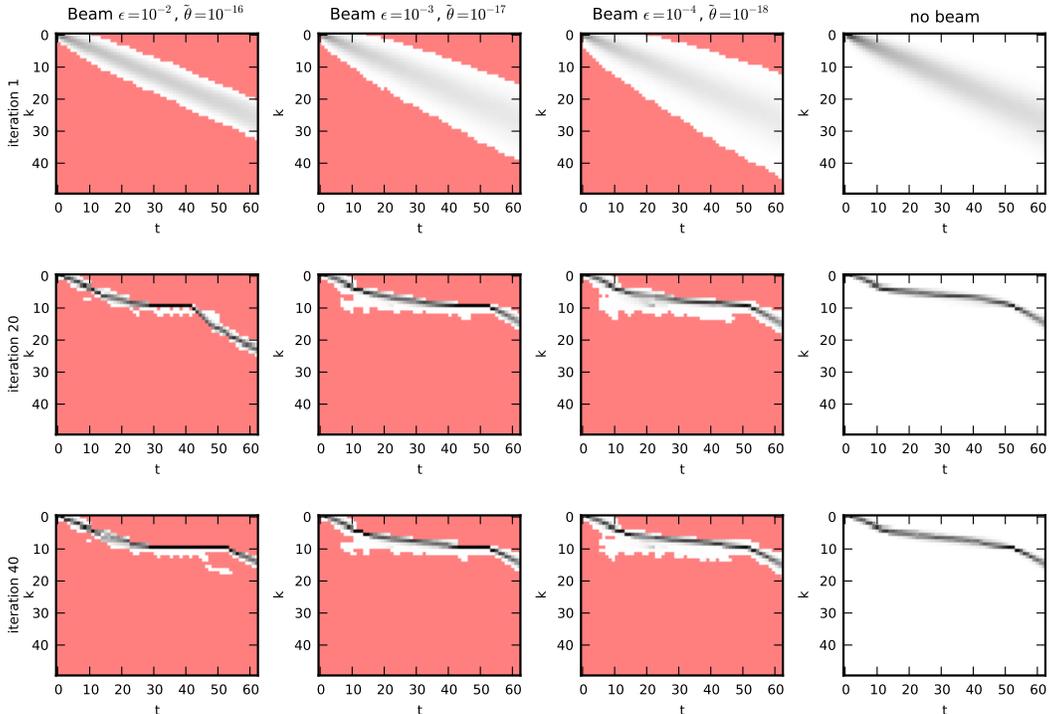


Figure 7: A set of heat maps generated at 20-step intervals during inference on the PHMM<sub>I</sub> dataset for a variety of beam settings. Each cell in the heat map indicates the expectation that either a *Match* or *Insert* state with parameter  $k$  (row index) generated the observed symbol at position  $t$  (column value) in the heat map. Darker colors indicate higher values, and red indicates that a hidden state was excluded from the beam.

over all categories - see Table 5a) and substantially decreased inference runtime (98.60% for the narrow beam, 96.90% for the medium beam, and 94.46% for the wide beam, averaged over all categories - see Table 5b).

To better visualize the effect of the beam on inference, we plotted the variational bound on the marginal likelihood against runtime for the first fold of the 5-fold training partition (Figure 8a). As expected, narrower beam thresholds converge faster, and the beam methods converge faster than the no-beam forward-backward algorithm for all thresholds tested. Figure 8b shows a zoomed portion of the graph in Figure 8a toward the end of convergence of the beam methods. This portion shows that inference using the beam method, unlike the standard forward-backward algorithm, does not necessarily increase the variational bound at every step. If the optimum that the algorithm approaches is well-behaved, then it is possible for the beam method to perform better than the non-beam method by ignoring irrelevant states. However, the beam could also ignore useful states, causing inference to progress to a non-optimal point in the model’s parameter space.

**5.3 SCOP Datasets: MSA initialization** We ran additional inference experiments where we attempted to improve perplexities from PHMMs constructed using multiple sequence alignments (MSAs) [3]. The MSAs were created on each training partition of our SCOP 1.75 dataset using the MUSCLE program [5] with default settings. We constructed PHMMs from each MSA using an 80% *Match*-state cutoff value. The cutoff values were used to determine the number of non-blanks in an alignment column that would cause us to associate that column with a *Match* state in the derived PHMM. The PHMM size  $K$  was set to the number of *Match* columns, and emission probabilities for each *Match* state were estimated using the frequencies of amino acids in each *Match* column of the alignment. We computed initial perplexities of the PHMM-MSAs using the marginal likelihood,  $p(x_n|A, B)$ , computed directly from the (no-beam) forward-backward algorithm.

Once these initial PHMMs were constructed, we used expected counts as a starting point for variational inference using both the standard forward-backward algorithm as well as our beam method with the same set of beam thresholds as in the previous set of SCOP 1.75 experiments. Unlike the previous experiments, we did not fix insert transition probabilities.

Narrower beams (large thresholds) produced higher

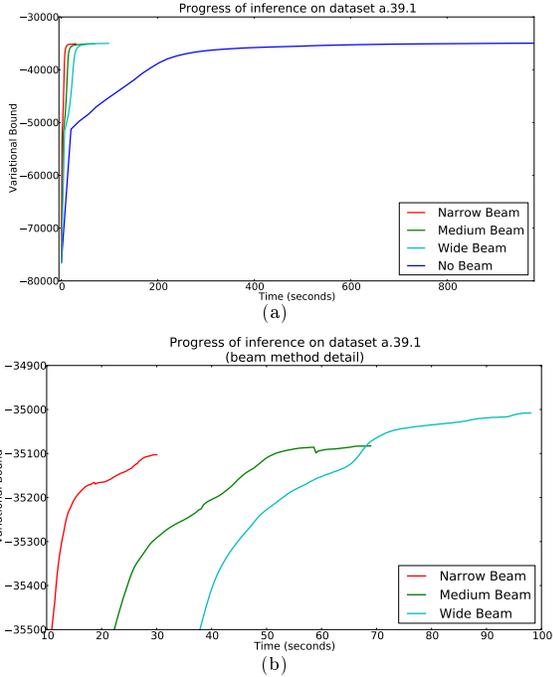


Figure 8: (a) Shows a comparison of rates of improvement of the variational bound between the beam algorithms using the three separate thresholds and the standard (no-beam) forward-backward algorithm. As the beam threshold decreases, inference speed increases. All beam settings converge faster than the no-beam method. (b) Shows detail for the beam methods. Unlike the non-beam method, each iteration of beam inference is not guaranteed to increase the variational bound. Both graphs show inference on superfamily a.39.1 from the SCOP 1.75, 95% dataset. “Narrow,” “medium,” and “wide” indicate threshold settings of  $\epsilon = [10^{-2}, 10^{-3}, 10^{-4}]$  and  $\bar{\theta} = [10^{-16}, 10^{-17}, 10^{-18}]$ , where  $\epsilon$  indicates the KL divergence beam threshold and  $\bar{\theta}$  indicates the auxiliary variable threshold.

perplexities than wider beams. In some cases, beams that were too narrow led to inference steps that decreased the variational bound. The beam methods improved perplexity from the MSA-induced PHMM for 5, 7, and 10 out of 26 categories for the narrow, medium, and wide beams respectively (see Table 6a for a full list of perplexities). For the categories where the beam method produced an improvement in perplexity, we saw average improvements in runtime of 92.68%, 95.88%, and 97.79% for the narrow, medium, and wide beams over no-beam inference, respectively (see Table 6b for a full list of runtimes).

Using approximate expectations from the beam during inference does not guarantee an inference algorithm that increases the variational bound at each step (see Figure 8). To mitigate this concern, the beam can be increased to allow inference to perform more similarly to the no-beam method, as indicated by the trend

Uniform Initialization: Test Set Perplexities				
Model	Beam			No Beam
	Narrow	Medium	Wide	
Average	16.68	16.50	16.48	17.31
StDev	1.81	1.80	1.83	1.94

(a)

Uniform Initialization: Inference Times (seconds)				
Model	Beam			No Beam
	Narrow	Medium	Wide	
Average	85	205	373	9138
StDev	54	151	260	7959

(b)

Table 5: The charts above show comparisons of average test-set perplexities (a) and convergence times (b) between the beam method and the standard (no beam) forward-backward inference averaged over all folds and categories of our SCOP 1.75 dataset. Inference was initialized with uniform expected transition and emission distributions. “Narrow,” “medium,” and “wide” indicate threshold settings of  $\epsilon = [10^{-2}, 10^{-3}, 10^{-4}]$  and  $\bar{\theta} = [10^{-16}, 10^{-17}, 10^{-18}]$ , where  $\epsilon$  indicates the KL divergence beam threshold and  $\bar{\theta}$  indicates the auxiliary variable threshold.

of decreasing average perplexities in Table 6a as the beam thresholds decrease. However, increasing the size of the beam reduces the speed of the algorithm.

## 6 Conclusion

We present a beam search inference algorithm for Profile HMMs. Our method is based on a new approach for aggregating transitions in the PHMM to form an equivalent standard HMM. We call this standard HMM the Geometric Transition HMM. A sub-portion of our beam method, the variational auxiliary variable criterion, is a thresholding scheme derived from the beam sampling method in [15] and constitutes a novel way of adapting auxiliary-variable sampling methods to the variational realm. Separately, we show how the GT-HMM can be used to understand the PHMM as an unbounded-size model that uses a parsimonious number of hidden states to represent finite-length sequences.

In experiments on both synthetically generated datasets and a subset of the SCOP 1.75 dataset, we show that our beam method significantly increases inference speed over standard PHMM inference methods and also maintains the model’s ability to represent amino acid sequences. The ways of understanding PHMMs that we presented in this paper have allowed us to construct more efficient inference methods, which have the potential to increase the speed of commonly-used bioinformatics applications.

MSA Initialization: Test Set Perplexities					
Model/ SCOP Category	Beam			No Beam	MSA PHMM
	Narrow	Medium	Wide		
a.1.1	13.89	<b>13.34</b>	<b>12.85</b>	<b>12.65</b>	13.67
a.39.1	14.75	13.31	13.44	<b>11.18</b>	11.66
a.4.1	17.63	17.13	16.71	<b>14.56</b>	15.33
b.1.18	17.72	17.33	<b>16.99</b>	<b>16.70</b>	17.10
b.1.2	18.19	18.19	18.08	<b>16.41</b>	16.94
b.121.4	15.56	14.07	14.46	<b>12.87</b>	13.80
b.29.1	<b>16.17</b>	<b>16.47</b>	<b>16.11</b>	<b>15.46</b>	16.59
b.36.1	13.65	13.15	13.25	<b>9.56</b>	9.85
b.40.4	<b>17.57</b>	<b>17.57</b>	<b>17.57</b>	<b>17.69</b>	18.05
b.47.1	<b>9.98</b>	<b>9.85</b>	<b>9.75</b>	<b>9.86</b>	10.22
b.55.1	17.78	17.60	17.85	<b>16.36</b>	16.85
b.6.1	17.81	17.51	16.85	<b>14.83</b>	15.31
c.3.1	17.36	16.60	<b>16.43</b>	<b>15.89</b>	16.43
c.47.1	17.99	17.99	17.99	<b>15.45</b>	15.98
c.55.1	17.10	17.36	17.13	<b>16.65</b>	16.88
c.66.1	18.09	18.08	18.09	<b>16.49</b>	17.29
c.67.1	17.87	17.85	17.86	<b>15.02</b>	15.51
c.69.1	17.91	17.70	17.68	<b>16.39</b>	17.04
c.94.1	<b>13.47</b>	<b>13.01</b>	<b>12.97</b>	<b>13.61</b>	15.38
d.108.1	18.29	18.26	18.28	<b>17.03</b>	17.28
d.144.1	18.34	17.41	17.20	<b>10.62</b>	11.17
d.15.1	16.69	16.42	16.69	<b>14.05</b>	14.46
d.3.1	<b>15.61</b>	<b>15.42</b>	<b>15.62</b>	<b>14.13</b>	15.96
d.58.7	11.89	11.51	<b>11.50</b>	<b>11.21</b>	11.51
g.37.1	12.53	<b>11.65</b>	<b>11.67</b>	<b>11.28</b>	11.70
g.39.1	14.16	14.38	14.36	<b>12.88</b>	13.38
Average	16.08	15.74	15.67	<b>14.18</b>	14.60

(a)

MSA Initialization: Inference Times (seconds)				
Model/ SCOP Category	Beam			No Beam
	Narrow	Medium	Wide	
a.1.1	29	47	106	839
a.39.1	24	37	67	760
a.4.1	4	10	23	401
b.1.18	5	11	72	475
b.1.2	8	23	71	867
b.121.4	30	83	116	2023
b.29.1	28	122	311	2867
b.36.1	5	15	33	499
b.40.4	4	15	39	809
b.47.1	32	42	70	1365
b.55.1	15	22	37	582
b.6.1	33	31	69	709
c.3.1	18	60	165	2860
c.47.1	21	38	80	1545
c.55.1	11	22	86	993
c.66.1	21	62	171	3398
c.67.1	23	102	108	5133
c.69.1	34	151	323	5732
c.94.1	50	77	198	2497
d.108.1	6	19	32	767
d.144.1	8	17	28	2454
d.15.1	14	19	16	239
d.3.1	35	69	103	1346
d.58.7	17	18	43	366
g.37.1	2	3	11	60
g.39.1	16	15	26	168
Average	19	43	92	1529

(b)

Table 6: The charts above show comparisons of average test-set perplexities (a) and convergence times in seconds (b) between our beam methods, the standard (no beam) forward-backward, and the MSA-derived PHMM ((a) only) for 26 superfamilies of the SCOP 1.75 dataset. Bolded numbers in (a) indicate experiments where a test perplexity from the inferred PHMM was lower than that of the MSA-derived PHMM on the same category. Inference was initialized using the MSA-induced PHMM. “Narrow,” “medium,” and “wide” indicate threshold settings of  $\epsilon = [10^{-2}, 10^{-3}, 10^{-4}]$  and  $\tilde{\theta} = [10^{-16}, 10^{-17}, 10^{-18}]$ , where  $\epsilon$  indicates the KL divergence beam threshold and  $\tilde{\theta}$  indicates the auxiliary variable threshold.

## References

- [1] M. J Beal, Z. Ghahramani, and C. E Rasmussen. The infinite hidden markov model. *Advances in Neural Information Processing Systems*, 1:577–584, 2002.
- [2] M. J Beal and M. MA. Variational algorithms for approximate bayesian inference. *Unpublished doctoral dissertation, University College London*, 2003.
- [3] S. R Eddy. Multiple alignment using hidden markov models. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, volume 3, pages 114–120, 1995.
- [4] S. R Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755, 1998.
- [5] R. C Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792, 2004.
- [6] T. S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, 1(2):209–230, 1973.
- [7] F. Jelinek. *Statistical methods for speech recognition*. the MIT Press, 1997.
- [8] D. J.C MacKay. *Ensemble learning for hidden Markov models*. Technical report, Cavendish Laboratory, University of Cambridge, 1997.
- [9] A. G Murzin, S. E Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.
- [10] J. Paisley and L. Carin. Hidden markov models with stick-breaking priors. *Signal Processing, IEEE Transactions on*, 57(10):3905–3917, 2009.
- [11] C. Pal, C. Sutton, and A. McCallum. Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V, 2006.
- [12] L. Rabiner and B. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16,

1986.

- [13] J. Sethuraman. A constructive definition of dirichlet priors. Technical report, FLORIDA STATE UNIV TALLAHASSEE DEPT OF STATISTICS, 1991.
- [14] Y. W Teh, M. I Jordan, M. J Beal, and D. M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- [15] J. Van Gael, Y. Saatici, Y. W Teh, and Z. Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095, 2008.

## A Variational Bound

We bound the marginal likelihood of the infinite PHMM using the following variational distribution:

$$\begin{aligned} \log p(x_{1:T}|\alpha, \beta) &= \log \int_A \int_B \sum_{\mathcal{Z}} p(A|\alpha)p(B|\beta)p(x_{1:T}, \mathcal{Z}|A, B) \\ &\geq \int_A \int_B \sum_{\mathcal{Z}} q(\mathcal{Z}, A, B) \log \frac{p(A|\alpha)p(B|\beta)p(x_{1:T}, \mathcal{Z}|A, B)}{q(\mathcal{Z}, A, B)} \\ &\equiv \int_A \int_B \sum_{\mathcal{Z}} q(\mathcal{Z}) \left( \prod_{(s,k)} q(A_{(s,k)}) \right) \left( \prod_{(s,k)} q(B_{(s,k)}) \right) \\ &\quad \log \frac{p(A|\alpha)p(B|\beta)p(x_{1:T}, \mathcal{Z}|A, B)}{q(\mathcal{Z}) \left( \prod_{(s,k)} q(A_{(s,k)}) \right) \left( \prod_{(s,k)} q(B_{(s,k)}) \right)} \end{aligned}$$

After maximizing with respect to  $q(A)$  and  $q(B)$ ,  $\left( \prod_{(s,k)} q(A_{(s,k)}) \right) \left( \prod_{(s,k)} q(B_{(s,k)}) \right) = p(A|\alpha)p(B|\beta)p(x_{1:T}, \mathcal{Z}|A, B)$ . We then compute the variational bound for the truncated model as follows:

$$\begin{aligned} &\log p(x_{1:T}|\alpha, \beta) \\ &\geq \sum_{\mathcal{Z}} q(\mathcal{Z}) \log q(\mathcal{Z}) \\ &= \sum_{(s,k), s'} E_{q(\mathcal{Z})} [n_{(s,k), s'}] E_{q(A_{(s,k), :})} [\log A_{(s,k), s'}] \\ &\quad + \sum_{(s,k), m} E_{q(\mathcal{Z})} [n_{(s,k), m}] E_{q(B_{(s,k), :})} [\log B_{(s,k), m}] \end{aligned}$$

## B Auxiliary Variable Beam Method

**B.1 Maximum with respect to  $q(u_t)$**  The variational free energy can be written as

$$\begin{aligned} \mathcal{F}(q(u_t)) &= E_{q(u_t)} \left[ \log \frac{\exp E_{q(z_{t-1})} [\log p(u_t|z_t, z_{t-1})p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1})]}{q(u_t)} \right] \\ &\quad + H(q(z_{t-1})) \end{aligned} \tag{B.1}$$

By Gibb's inequality, the maximum with respect to  $q(u_t)$  is therefore

$$\propto \exp \left( E_{q(z_{t-1})} [\log p(u_t|z_t, z_{t-1})p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1})] \right) \tag{B.2}$$

with expectations computed as follows:

$$\begin{aligned} &E [\log p(u_t|z_t, z_{t-1})] \\ &= \sum_{z_{t-1}} q(z_{t-1}) \log p(u_t|z_t, z_{t-1}) \\ &= \sum_{z_{t-1}} q(z_{t-1}) \log \mathbb{I}(u_t < p(z_t|z_{t-1})) \\ &\quad - \sum_{z_{t-1}} q(z_{t-1}) \log p(z_t|z_{t-1}) \\ &= \begin{cases} -\infty & \exists z_{t-1} : (u_t \geq p(z_t|z_{t-1})) \wedge (q(z_{t-1}) > 0) \\ -\sum_{z_{t-1}} \mathbb{I}(q(z_{t-1}) > 0) q(z_{t-1}) \log p(z_t|z_{t-1}) & \text{o.w.} \end{cases} \end{aligned} \tag{B.3}$$

$$E [\log p(z_t|z_{t-1})] = \sum_{z_{t-1}: q(z_{t-1}) > 0} q(z_{t-1}) \log p(z_t|z_{t-1}) \tag{B.4}$$

The reduced expression for  $q(u_t)$  becomes

$$q(u_t) \propto \begin{cases} 0 & \exists z_{t-1} : (u_t \geq p(z_t|z_{t-1})) \wedge (q(z_{t-1}) > 0) \\ E_{q(z_{t-1})} [\log p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1})] & \text{o.w.} \end{cases} \tag{B.5}$$

This final expression gives us the form of  $q(u_t)$ , but the threshold above which  $q(u_t) = 0$  depends on the values where  $q(z_{t-1}) = 0$ . We can therefore choose an initial  $q(u_t)$  either by truncating  $q(z_{t-1})$ , or by providing  $q(u_t)$  with an explicit threshold. We choose to use the latter scheme:  $q(u_t) = \frac{\mathbb{I}(u_t < \tilde{\theta}_t)}{\tilde{\theta}_t}$  with an initial variational threshold parameter  $\tilde{\theta}_t$ .

**B.2 Maximum with respect to  $q(z_{t-1})$**  The variational free energy can be written as

$$\begin{aligned} &\mathcal{F}(q(z_{t-1})) \\ &= E_{q(z_{t-1})} \left[ \log \frac{\exp (E_{q(u_t)} [\log p(u_t|z_t, z_{t-1})]) p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1})}{q(z_{t-1})} \right] \\ &\quad + H(q(u_t)) \end{aligned} \tag{B.6}$$

By Gibb's inequality, the maximum with respect to  $q(z_{t-1})$  is

$$\propto \exp (E [\log p(u_t|z_t, z_{t-1})]) p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1}) \tag{B.7}$$

$E [\log p(u_t|z_t, z_{t-1})]$  is computed as follows:

$$\begin{aligned} &E [\log p(u_t|z_t, z_{t-1})] \\ &= \int_0^1 q(u_t) \log p(u_t|z_t, z_{t-1}) du_t \\ &= \int_0^{\tilde{\theta}_t} q(u_t) \log \mathbb{I}(u_t < p(z_t|z_{t-1})) du_t - \log p(z_t|z_{t-1}) \\ &= \begin{cases} -\infty & \tilde{\theta}_t \geq p(z_t|z_{t-1}) \\ -\log p(z_t|z_{t-1}) & \text{o.w.} \end{cases} \end{aligned} \tag{B.8}$$

with

$$\begin{aligned} &E_{q(u_t)} [\log \mathbb{I}(u_t < p(z_t|z_{t-1}))] \\ &= \int_0^{\tilde{\theta}_t} q(u_t) \log \mathbb{I}(u_t < p(z_t|z_{t-1})) du_t \\ &= \begin{cases} \int_0^{\tilde{\theta}_t} q(u_t) \log (1) du_t & \tilde{\theta}_t < p(z_t|z_{t-1}) \\ \int_0^{p(z_t|z_{t-1})} q(u_t) \log (1) du_t + \int_{p(z_t|z_{t-1})}^{\tilde{\theta}_t} q(u_t) \log (0) du_t & \tilde{\theta}_t \geq p(z_t|z_{t-1}) \end{cases} \\ &= \begin{cases} 0 & \tilde{\theta}_t < p(z_t|z_{t-1}) \\ -\infty & \tilde{\theta}_t \geq p(z_t|z_{t-1}) \end{cases} \end{aligned} \tag{B.9}$$

The expression for  $q(z_{t-1})$  therefore becomes

$$q(z_{t-1}) \propto \begin{cases} 0 & \tilde{\theta}_t \geq p(z_t|z_{t-1}) \\ p(z_{t-1}, x_{1:t-1}) & \text{o.w.} \end{cases} \tag{B.10}$$

The form of  $q(z_{t-1})$  shows us that a specific value of  $\tilde{\theta}_t$  will act as a cutoff, forcing values of  $q(z_{t-1})$  with associated  $p(z_t|z_{t-1})$  to zero. This setting of  $q(z_{t-1})$ , in turn, adjusts  $q(u_t)$  so that  $\tilde{\theta}_t$  moves to the smallest value of  $p(z_t|z_{t-1})$  greater than the initial  $\tilde{\theta}_t$ . After this second maximization step, no further changes occur in either variational distribution.