Efficient Approximations for Learning Phylogenetic HMM Models from Data

V. Jojic*, N. Jojic*, C. Meek*, D. Geiger‡,
A. Siepel†, D. Haussler†, and D. Heckerman*
*Microsoft Research, ‡Technion Israel, †UC Santa Cruz
vjojic@psi.toronto.edu

September 2003, Updated January 2004

Technical Report
MSR-TR-2003-62

We consider models useful for learning an evolutionary or phylogenetic tree from data consisting of DNA sequences corresponding to the leaves of the tree. In particular, we consider a general probabilistic model described in [16] that we call the phylogenetic-HMM model which generalizes the classical probabilistic models of Neyman and Felsenstein. Unfortunately, computing the likelihood of phylogenetic-HMM models is intractable. We consider several approximations for computing the likelihood of such models including an approximation introduced in [16], loopy belief propagation, and several variational methods. We demonstrate that, unlike the other approximations, variational methods are accurate and are guaranteed to lower bound the likelihood. In addition, we find that the variational approximation that performs best is the one whose q distribution corresponds to the classic Neyman–Felsenstein model. The application of our best approximation to data from the Cystic Fibrosis Transmembrane Conductance Regulator (CFTR) gene region across nine eutherian mammals reveals a CpG effect.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction

We consider the problem of learning an evolutionary or phylogenetic tree from data consisting of DNA sequences corresponding to the leaves of the tree. We concentrate on a standard probabilistic model-selection approach wherein models are scored by some criterion (e.g., maximum likelihood, Bayesian Information Criterion) and some heuristic search method is used to find a tree or set of trees with high scores (e.g., [5, 6]). We further concentrate on methods for computing the likelihood of a given model.

The classic probabilistic model used in this approach is described by (e.g.) Neyman [13] and Felsenstein [6]. The model incorporates several strong assumptions including (1) evolution takes place independently at each base-pair site, (2) the base-pair substitution rate is uniform over sites, (3) there is no recombination, (4) there is no lateral gene transfer, and (5) the sequences are aligned. There have been numerous efforts to relax these assumptions (e.g., [7, 11, 16, 17, 20]). In this paper, we address the relaxation of the first assumption by considering the combined tree-HMM model described in [16] in which base-pair substitutions are dependent on neighboring bases. We call this hybrid model a *phylogenetic-HMM model*. We do not address the relaxation of the other assumptions so as to isolate the effects of the first assumption and to avoid substantial added complexity.

One important drawback of phylogenetic-HMM models is that evaluating the likelihood of such a model (and hence finding parameters that maximize this likelihood) is intractable. Recently, Siepel and Haussler [16] introduced an efficient approximation for computing the likelihood of phylogenetic-HMM models. Unfortunately, as we shall see, this approximation has no accuracy guarantees and, thus, may be inappropriate for use in model selection. In this paper, we describe phylogenetic-HMM models in terms of Bayesian networks, also known as directed acyclic graphical (DAG) models (e.g., [14]). We introduce several approximations developed for graphical models based on *variational techniques* (e.g., [10]) that efficiently yield a lower-bound on the likelihood of a phylogenetic-HMM model. In experiments on real data, we show that these lower bounds are tight. We also describe another approximation for computing likelihood in graphical models known as *loopy belief propagation* (e.g., [14]). This approximation has no accuracy guarantees and, as we show experimentally, yields poor likelihood estimates.

In Section 2, we describe phylogenetic-HMM models in terms of Bayesian networks or DAG models. In Section 3, we describe the approximation for evaluating the likelihood of phylogenetic-HMM models presented in [16]. In Section 4 through 5, we describe variational methods and methods based on loopy belief propagation for this task. In Section 6, we introduce structured variational techniques tailored to our model. Structured variational approximations go beyond the standard mean-field approximation, and our tailored approximations (to our knowledge) have not been described previously. In Section 7, we overview the learning algorithms; and in Section 8, we discuss experimental results on real data and find that, among the approximations, the one that performs best is the one that uses the structure of the classic classic Neyman–Felsenstein model. In Section 9, we apply this approximation to data from the Cystic Fibrosis Transmembrane Conductance Regulator (CFTR) gene region across nine eutherian mammals and, in doing so, identify a CpG effect (high mutation rate of CG to TG, due to methylation and spontaneous deaminiation).

2 The Model

The phylogenetic-HMM models that we consider are identical to those described in [16]. As we shall see, it will be convenient to describe these models as DAG models.

Given a domain of interest having a set of finite variables $\mathbf{s} = (s_1, \ldots, s_n)$ with a positive joint distribution $p(\mathbf{s})$, a *DAG model for* $\mathbf{s}$ is a pair $(\mathcal{G}, \mathcal{P})$ where $\mathcal{G}$ is a directed acyclic graph and $\mathcal{P}$ is a set of conditional probability distributions. Each node in $\mathcal{G}$ corresponds to a variable in $\mathbf{s}$. We use S_i to refer to both the variable and its corresponding node. Arcs in the graph correspond to probabilistic

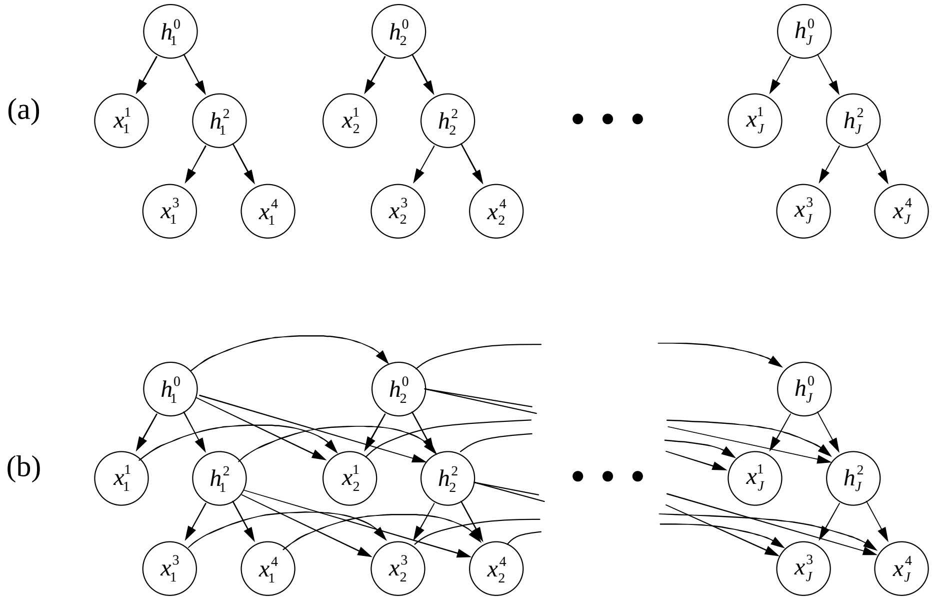


Figure 1: Probabilistic phylogenetic models expressed as DAG models. (a) The Neyman–Felsenstein tree model. (b) The dinucleotide phylogenetic-HMM model.

dependencies or, more precisely, the absence of arcs correspond to probabilistic independencies. These independencies are precisely those that allow us to write the joint distribution as follows:

$$p(\mathbf{s}) = \prod_{i=1}^{N} p(s_i|\mathbf{pa}(s_i)) \tag{1}$$

where $\mathbf{pa}(s_i)$ are the parents of s_i in the graph. The distributions $p(s_i|\mathbf{pa}(s_i))$ are called *local probability distributions.*

The DAG model structure corresponding to the tree model of (e.g.) [6] is shown in Figure 1a. The variable h_j^i corresponds to the unknown nucleotide in ancestor species i at nucleotide site j. The variable x_j^i corresponds to the observed nucleotide in existing species i at site j. The strong assumption that evolution takes place independently at each nucleotide side is reflected in the lack of arcs among the sites. The DAG model corresponding to a simple phylogenetic-HMM model is shown in Figure 1b. In this *dinucleotide model* [16] the identity of nucleotide at site j is dependent on the ancestor nucleotide at site j (as in the Neyman–Felsenstein model), as well as the corresponding child and parent nucleotides at site $j-1$.

Additional complex models are defined in [16], wherein two or more previous slices influence the identity of the base as a given position. To discuss all such models, we introduce the following notation. First, we often drop the explicit indication of whether a variable is observed or not, using s_j^i to refer to the variable for species i at site position j. Further, we use the regularity of the Bayes net to define the connectivity by two sets of parent indices for each variable:

- $\mathcal{T}_i$ is a set of species indices that are parents of the ith species. This parent information is the same for each site j and defines the phylogenetic tree. In Figure 1, for example $\mathcal{T}_3 = 2$

- $\mathcal{C}_j$ is a set of site indices that are parents of the jth site. This parent information is the same in each sequence i, and defines the Markov-chain model. In Figure 1b, for example $\mathcal{C}_2 = 1$. In fact, in this paper, it is always true that $\mathcal{C}_j = \{j-1\}$, but the derivations presented here can be carried out in an analogous way for more general situations, for example, when $\mathcal{C}_j = \{j-1, j-2, ..., j-k\}$.

The indices of all parents of a variable s^i_j can then be written as

$$\mathcal{P}(s^i_j) = \mathcal{T}_i \times \mathcal{C}_j \quad \cup \quad \{i\} \times \mathcal{C}_j \quad \cup \quad \mathcal{T}_i \times \{j\}, \tag{2}$$

and the parent variables are

$$\mathbf{pa}(s^i_j) = \{s^k_l\}_{(k,l)\in\mathcal{P}(s^i_j)}, \tag{3}$$

or, to use a different notation,

$$\mathbf{pa}(s^i_j) = s_{\mathcal{P}(s^i_j)}. \tag{4}$$

We will also use $\mathbf{s}^{\mathcal{T}_i}_j$ to denote the parents of s^i_j that share its site index j, and $\mathbf{s}^i_{\mathcal{C}_j}$ to denote the parents that share its species index i. The joint probability distribution is

$$p(\mathbf{s}) = \prod_{i,j} p(s^i_j | s_{\mathcal{P}(s^i_j)}). \tag{5}$$

For instance, if $\mathcal{C}$ and $\mathcal{T}$ define a chain each, i.e., $\mathcal{T}_i = \{i-1\}$, $\mathcal{C}_j = \{j-1\}$, then $\mathbf{pa}(s^i_j) = \{s^{i-1}_{j-1}, s^i_{j-1}, s^{i-1}_j\}$, and the resulting grid probability model is defined as $p(\mathbf{s}) = \prod_{i,j} p(s^i_j | s^{i-1}_{j-1}, s^i_{j-1}, s^{i-1}_j)$.

Our experiments are restricted to the dinucleotide model wherein $\mathcal{C}_j = \{j-1\}$ (although the tree is not reduced to a chain and $\mathcal{T}_i \neq \{i-1\}$). As in a regular phylogenetic tree, each node has a single parent species, and thus $\mathcal{T}_i$ has a single element, unless i is the root, in which case it is empty. Thus, $\mathbf{pa}(s^i_j)$ still has at most three variables as in the case of a grid model. However, we derive the methods in a general form so as to apply to other neighborhood relations, including the case when each site j in a sequence is connected to several previous sites, rather than just to site $j-1$, and the case of the horizontal gene transfer, where $\mathcal{T}_i$ could have more than one variable.

In the remainder of this section, we examine the local probability distributions $p(s|\mathbf{pa}(s))$ in our models. These distributions correspond to well-known models of DNA substitution.

First, consider the Neyman–Felsenstein model wherein evolution at each slice is independent. Here, we need $p(s^i_j|s^k_j)$, where species k is the parent of species i. Models of DNA substitution for this case are generally based on a continuous-time Markov model of base substitution, with the instantaneous rate of replacement of each base for each other defined by a rate matrix Q [18,22]. As a continuous-time Markov matrix, $Q = \{q_{i,j}\}$ $(1 \leq i, j \leq 4)$ is constrained to have each of its rows sum to zero. In an "unrestricted" model, Q has $4^2 - 4 - 1 = 11$ free parameters. To determine the local distribution, we assume that the Markov process defined by Q runs for a given (evolutionary) time t. Let $P(t)$ be the matrix of substitution probabilities—that is, the local distribution—for length t (note that $P(t)$ is a *discrete* Markov matrix, with rows summing to 1, while Q is a *continuous* Markov matrix, with rows summing to 0). $P(t)$ is thus given by the solution to the differential equation $\frac{d}{dt}P(t) = P(t)Q$ with initial conditions $P(0) = I$, which is $\boldsymbol{P(t) = e^{Qt}}$. Q is generally diagonalizable as $Q = S\Lambda S^{-1}$, allowing $P(t)$ to be computed as $\boldsymbol{P(t) = Se^{\Lambda t}S^{-1}}$, where $e^{\Lambda t}$ is the diagonal matrix obtained by exponentiating each element on the main diagonal of Λt.

Now, consider the local probability distributions for the dinucleotide model used in our experiments. The local distribution is built from a continuous-time Markov model for dinucleotide pairs associated with a 16×16 Q matrix. Given time t, we obtain the conditional distribution of a dinucleotide pair given its parent species—say $p(s^i_j, s^i_{j-1}|s^k_j, s^k_{j-1})$—in the same manner as described for the Neyman–Felsenstein model. This distribution then determines $p(s^i_j|s^i_{j-1}, s^k_j, s^k_{j-1})$. To reduce the number of possible free parameters in Q, we assume that substitutions are strand symmetric. For example, we assume that the substitution rate for CG to TG is equal to that for CG to CA. In addition, simultaneous substitutions involving more than one base are disallowed (despite biological evidence for such changes [2]). This model is called the U2S model in [16].

We note that this model for substitutions is inconsistent in spirit with our dinucleotide model. In particular, the dinucleotide model assumes that substitutions are dependent on the current and previous base-pair sites. If such dependence were allowed to propagate across the many generations implicit in a single edge of an evolutionary tree model, then the substitution at any given site would be a function of the base pairs at all other sites. The same inconsistency is present and noted in [16]. Methods for removing this inconsistency for the simple (two-sequence) case have been discussed [1, 9, 15], but these methods are difficult to extend to the general case.

3 A Simple Markov-Chain Approximation

As discussed in the introduction, a key task in learning evolutionary trees from data is the evaluation of a given model's score. Here, we shall restrict our attention to the likelihood or log-likelihood score:

$$\log p(\mathbf{x}|\theta) = \log \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}|\theta) \tag{6}$$

where $\mathbf{x}$ and $\mathbf{h}$ are the set of all x^i_j and h^i_j, respectively, and θ are the parameters that specify the local probability distributions. The computation of this likelihood is also important, because it corresponds to the E step of Expectation–Maximization (EM) and EM-like algorithms that are used to identify the maximum likelihood parameters for a model (see Section 7).

Equation 6 is an example of *probabilistic inference*—the computation of some marginal or conditional probability of interest from a joint distribution. Felsenstein [6] showed how to perform this inference exactly and efficiently for his model. Pearl [14] discovered the same algorithm—essentially a tree version of dynamic programming in which independence relations are used to rearrange sums of products as products of sums. As is done in the machine-learning community, we also shall use the phrase *belief propagation* to refer to probabilistic inference in a DAG model.

Unfortunately, no efficient inference method for the phylogenetic-HMM models are known. In fact, inference in a DAG model is NP-hard [3], and it is known that the presence of undirected cycles in DAG models increases the computational burden of inference. The phylogenetic-HMM models contain numerous undirected cycles, making it extremely unlikely that the inference Equation 6 can be performed efficiently.

In this section, we examine a simple *Markov-chain approximation* introduced in [16] for performing this inference. For simplicity, we describe this approximation for the dinucleotide model only. To understand this approximation, let $\mathbf{x}_j$ denote the set of observed nucleotides at site j. In the approximation, we model the observed data as a Markov chain:

$$p(\mathbf{x}|\theta) \cong p(\mathbf{x}_1|\theta) \prod_{j=2}^{J} p(\mathbf{x}_j|\mathbf{x}_{j-1}, \theta) \tag{7}$$

We further approximate each term by imposing additional (and mutually inconsistent) independence assumptions on the dinucleotide model in Figure 1b. The approximate computations of the first three terms in Equation 7 are illustrated in Figure 2. In (a), $p(\mathbf{x}_1|\theta)$ is computed efficiently by assuming

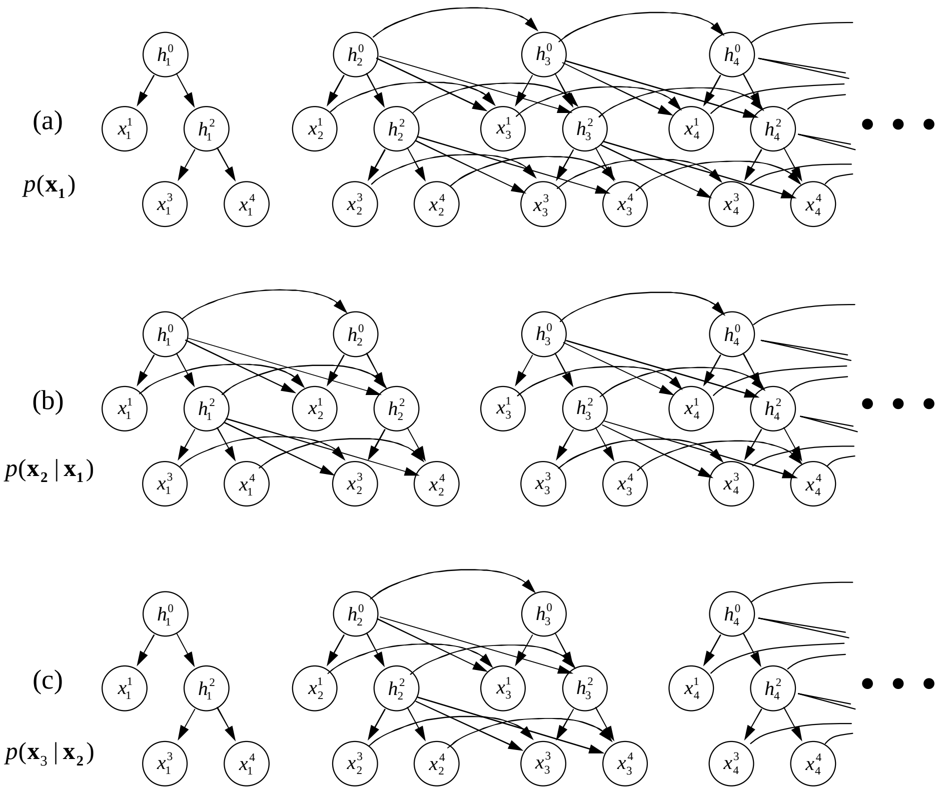


Figure 2: An illustration of the simple Markov-chain approximation [16] for $p(\mathbf{x}|\theta)$. (a) The model used to compute $p(\mathbf{x}_1|\theta)$. (b) The model used to compute $p(\mathbf{x}_2|\mathbf{x}_1, \theta)$. (c) The model used to compute $p(\mathbf{x}_3|\mathbf{x}_2, \theta)$.

that evolution at the first nucleotide site is independent of the nucleotides at the remaining sites. In (b), $p(\mathbf{x}_2|\mathbf{x}_1, \theta)$ is computed efficiently by assuming that evolution at the first two nucleotide sites is independent of the nucleotides at the remaining sites. In (c), $p(\mathbf{x}_3|\mathbf{x}_2, \theta)$ is computed efficiently by assuming that evolution at the second and third nucleotide sites is independent of the nucleotides at the remaining sites. The computations of the remaining terms are similar.

As may be expected, this approximation has no accuracy guarantees and consequently may be inappropriate as a criterion for model selection. To understand this observation, consider a tree structure that has a root node and two leaves. For simplicity, suppose that our alphabet has only two letters and the two sequences at the leaves are identical: 212. Consider two parameterized models for this structure having the following dinucleotide Q matrices (columns and rows are indexed by dinucleotides in lexicographic order):

$$\begin{pmatrix} -2 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -3 \end{pmatrix} \qquad \begin{pmatrix} -3 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 1 & 1 & -2 \end{pmatrix}$$

Then, for the first model, the true and approximate likelihood of the data are 0.5361 and 0.0175, respectively. The approximation underestimates the true likelihood. For the second model, the true and approximate likelihood of the data are 0.01609 and 0.4602, respectively. In contrast to the first situation, the approximate overestimates the true likelihood.

In what follows, we consider an approximation that comes with a guarantee.

4 Free energy and log likelihood

As noted in the introduction, a standard criterion to optimize in graphical models is the likelihood or the log likelihood of the observed data, obtained by summing or integrating over the hidden variables for a given set of parameters $\boldsymbol{\theta}$,

$$\log p(\mathbf{x}|\theta) = \log \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}|\theta). \tag{8}$$

However, for many models, including the model we study in this paper, finding the maximum-likelihood parameters and even just the computation of the log likelihood is intractable. Thus, approximate methods must be used. The machine-learning community has recently started to converge to a unified view of various approximations. This view is based on an alternative cost, named *free energy* for its similarity with the quantity used in statistical physics, bounds the negative log likelihood of the data, and is defined as

$$\begin{aligned} F &= \sum_{\mathbf{h}} q(\mathbf{h}) \log \frac{q(\mathbf{h})}{p(\mathbf{x}, \mathbf{h}|\theta)} = \\ &= \sum_{\mathbf{h}} q(\mathbf{h}) \log q(\mathbf{h}) - \sum_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{x}, \mathbf{h}|\theta) \end{aligned} \tag{9}$$

where $q(\mathbf{h})$ is an arbitrary distribution. Making the substitution $q(\mathbf{h}) = p(\mathbf{h}|\mathbf{x}, \theta))$ yields $F = -\log p(\mathbf{x}|\theta)$. In addition, using Jensen's inequality, it can be shown that

$$F \geq -\log p(\mathbf{x}|\theta) \tag{10}$$

for *any* probability distribution $q(\mathbf{h})$—that is for any function $q(\mathbf{h})$ such that $\sum_{\mathbf{h}} q(\mathbf{h}) = 1$. Thus, q is seen as an approximate posterior distribution, that can be used to compute a lower bound on the log likelihood of the data.

Estimating the posterior distribution can be achieved by minimizing the free energy. If q is unrestricted, then F is minimized at $q = p(\mathbf{h}|\mathbf{x})$. For example, if in our model, $\mathcal{C}_j$ are all empty, the model decomposes into a collection of independent graphs (except that they share parameters), and for each of them, the free energy has the form:

$$\begin{aligned} F_{tree} = & \sum_{\{h^i\}_{i=1}^I} q(\{h^i\}_{i=1}^I) \log q(\{h^i\}_{i=1}^I) - \\ & \sum_{\{h^i\}_{i=1}^I} q(\{h^i\}_{i=1}^I) \sum_i \log p(s^i|\mathbf{s}^{\mathcal{T}_i})), \end{aligned} \tag{11}$$

where s denotes both hidden ($\mathbf{h}$) and observed ($\mathbf{x}$) variables, as described earlier. Minimizing this energy with respect to $q(\{h^i\}_{i=1}^I)$ leads to $q(\{h^i\}_{i=1}^I) = p((\{h^i\}_{i=1}^I|\mathbf{x})$ and $F = -\log p(\mathbf{x})$. We will later use this observation in reverse—that is we iteratively will update various costs of this form, but instead of optimizing the cost using a numerical procedure, we will use belief propagation (described in Section 2), to compute optimal $p((\{h^i\}_{i=1}^I|\mathbf{x})$ and $\log p(\mathbf{x})$ efficiently.

For the more complex case, when each variable is connected both vertically and horizontally, belief propagation cannot be used (except as an approximation), and (as we shall see) we are better off optimizing F by varying the function q with the aim of lowering it to be as close to $-\log p(\mathbf{x})$ as possible. Such approaches are known as *variational techniques* in the machine-learning community.

Because the inequality (10) is satisfied for all probability distributions q, it is possible to limit the search space to approximate forms of the function q that lead to more tractable optimization. As the bound is tighter the closer we get to the true posterior, one should naturally attempt to limit the severity of the approximations. Usual ways of approximating the posterior are to either choose a particular functional form (e.g., exponential, even if the true posterior does not follow this form), and/or to decouple hidden variables that are in the true posterior correlated. These approximations are typically less severe when the posterior has a small number of pronounced modes, although the optimization of the free energy could get stuck in a local minima.

In the next section, we review the standard approximation that assumes all hidden variables in the model are mutually independent. This approximation, usually referred to as the *mean-field approximation*, predates the variational techniques, as does another related technique, the iterative conditional modes (ICM). From the variational optimization point of view, both of these techniques use the same factorization of the posterior, with ICM further limiting the form of each distribution to the Dirac form.

In Section 5.2, we present a related technique that instead of optimizing for q directly, optimizes its marginals. This technique is equivalent to belief propagation on graphs, but is used as an approximation on graphs that have loops (such as ours), and when used that way, its popular name is *loopy belief propagation.* One of the problems with this technique is that it does not directly fall into the category of variational methods and it is not guaranteed to converge. However, when it converges, it converges to a local maximum of the Bethe free energy [23].

As the posterior distribution in these problems is relatively broad, these simple approximations tend to provide poor bounds on the log likelihood. In Sections 6.1 and 6.2 , we present new approximations that are less restrictive and that lead to considerably better bounds on the log likelihood.

5 Standard simple posterior approximations

5.1 Mean field and iterative conditional modes: full factorization of the posterior

One efficient form of q is

$$q = \prod_{i,j} q(h^i_j), \tag{12}$$

where for each variable h^i_j, the individual distribution is normalized—that is, $\sum_{h^i_j} q(h^i_j) = 1$—which guarantees the required normalization of the entire posterior $\sum_{\mathbf{h}} q = 1$. This property simplifies the free energy into the following form:

$$\begin{aligned} F &= \sum_{i,j} \sum_{h^i_j} q(h^i_j) \log q(h^i_j) - \qquad (13) \\ &-\sum_{i,j} \left[q(h^i_j) \prod_{(k,\ell)\in\mathcal{P}(h^i_j)} q(h^k_\ell) \right] \log p(h^i_j | h_{\mathcal{P}(h^i_j)}) \\ &-\sum_{i,j} \left[\prod_{(k,\ell)\in\mathcal{P}(x^i_j)} q(h^k_\ell) \right] \log p(x^i_j | x_{\mathcal{P}(x^i_j)}) \end{aligned}$$

This isolates the immediate effect of any $q(h^i_j)$ from far away variables and allows a sequence of local optimization of the free energy to be performed:

- Initialize all distributions $q(h^i_j)$ to be uniform.

- LOOP
  - Keeping all other $q(h^i_j)$ fixed, update $q(h^1_1)$ so as to minimize F.
  - Keeping all other $q(h^i_j)$ fixed, update $q(h^1_2)$ so as to minimize new F.

    ...
  - Keeping all other $q(h^i_j)$ fixed, update $q(h^I_J)$ so as to minimize new F.
- END

Each step above lowers the free energy, and since the free energy is bounded from below by the log likelihood of the data $\log p(\mathbf{x})$, the procedure must converge. That is, at some point, F will stop changing significantly and the loop with exit.

Each minimization is tractable, as the targeted distribution $q(h^i_j)$ only affects a small number of terms in the summation. For example, when $\mathcal{T}_i = \{i-1\}$, $\mathcal{C}_j = \{j-1\}$, and thus $\mathbf{pa}(h^i_j) = \{h^{i-1}_{j-1}, h^i_{j-1}, h^{i-1}_j\}$, then the terms in the free energy that involve $q(h^i_j)$ are just

$$\sum_{h^i_j} q(h^i_j) \log q(h^i_j) - \sum_{h^i_j} q(h^i_j) \Bigg[\quad E \log p(h^i_j|\mathbf{pa}(h^i_j)) + E \log p(h^{i+1}_j|\mathbf{pa}(h^{i+1}_j)) + \\ E \log p(h^i_{j+1}|\mathbf{pa}(h^i_{j+1})) + E \log p(h^{i+1}_{j+1}|\mathbf{pa}(h^{i+1}_{j+1}))] \Bigg]$$

where

$$\begin{aligned}
E \log p(h^i_j|\mathbf{pa}(h^i_j)) &= \sum_{h^{i-1}_{j-1}, h^{i-1}_j, h^i_{j-1}} q(h^{i-1}_{j-1}) q(h^i_{j-1}) q(h^{i-1}_j) \log p(h^i_j|\mathbf{pa}(h^i_j)), \\
E \log p(h^{i+1}_j|\mathbf{pa}(h^{i+1}_j)) &= \sum_{h^i_{j-1}, h^{i+1}_j, h^{i+1}_{j-1}} q(h^i_{j-1}) q(h^{i+1}_{j-1}) q(h^{i+1}_{j-1}) \log p(h^{i+1}_j|\mathbf{pa}(h^{i+1}_j)), \\
E \log p(h^i_{j+1}|\mathbf{pa}(h^i_{j+1})) &= \sum_{h^{i-1}_j, h^{i-1}_{j+1}, h^i_{j+1}} q(h^{i-1}_j) q(h^i_{j+1}) q(h^{i-1}_{j+1}) \log p(h^i_{j+1}|\mathbf{pa}(h^i_{j+1})), \\
E \log p(h^{i+1}_{j+1}|\mathbf{pa}(h^{i+1}_{j+1})) &= \sum_{h^{i+1}_{j+1}, h^i_{j+1}, h^{i+1}_j} q(h^{i+1}_{j+1}) q(h^{i+1}_j) q(h^i_{j+1}) \log p(h^{i+1}_j|\mathbf{pa}(h^{i+1}_j)).
\end{aligned}$$

Thus, minimizing the free energy under the constraint $\sum q(h^i_j) = 1$ results in the following step in the variational optimization:

$$q(h^i_j) \propto \exp \left[E \log p(h^i_j|\mathbf{pa}(h^i_j)) + E \log p(h^{i+1}_j|\mathbf{pa}(h^{i+1}_j)) + E \log p(h^i_{j+1}|\mathbf{pa}(h^i_{j+1})) + E \log p(h^{i+1}_{j+1}|\mathbf{pa}(h^{i+1}_{j+1})) \right],$$

where the actual q to be used in the next step, and eventually in the energy computation, is obtained by normalizing the above function.

The posterior can be further simplified if it is assumed to have a single peak and all zeros—that is, only the information about the most likely letter for h^i_j is preserved:

$$\hat{h}^i_j = \arg\max q(h^i_j). \tag{14}$$

It can be shown that this further reduces the algorithm to simply iteratively optimizing the *joint* probability distribution one hidden variable at a time. While the mean field method is somewhat prone to local maxima, this method, known as *iterative conditional modes* (ICM), has even worse accuracy, due to additional approximations. These approximations make the bound very loose, and further, the iterative algorithm is even more prone to local maxima.

5.2 Loopy belief propagation: Approximating marginals using belief propagation

A problem with a mean field technique is that it treats hidden variables as independent. Whereas the posterior is optimized to take into account the dependencies in the model, if there is a lot of uncertainty about the hidden variables in the posterior, the correlations among the possible configurations are not captured. Instead of optimizing only for single-variable marginals, it would be useful to also optimize for marginals defined on the subsets of the hidden variables. Obviously, the single-variable and multi-variable marginals would have to agree, and this could be done by introducing Lagrange multipliers.

The belief-propagation techniques can be derived in this manner. We use two type of marginals of the q distributions:

- Single-variable marginals $\phi(h_j^i) = \sum_{\mathbf{h}\setminus h_j^i} q(\mathbf{h})$ analogous to $q(h_j^i)$ in the
- Clique marginals $\phi(C_k) = \sum_{\mathbf{h}\setminus C_k} q(\mathbf{h})$, where C_k are all the variables in the k-th clique in the graph, e.g., $C_k = \{h_{j-1}^{i-1}, h_j^{i-1}, ..., h_{j+1}^{i+1}\}$.

Then, the entropy term $\sum_{\mathbf{h}} q(\mathbf{h}) \log q(\mathbf{h})$ in the free energy expression (9) is approximated by assuming:

$$\log q(\mathbf{h}) \approx \log \left(\prod_{i,j} \phi(h_j^i) \prod_k \frac{\phi(C_k)}{\prod_{(i,j)\in C_k} \phi(h_j^i)} \right). \tag{15}$$

When this expression is plugged back into (9), we obtain a different free energy used in statistical physics—the Bethe free energy–which is an approximation of the free energy, but comes without the convergence and bounding properties of the variational methods unless the graph is a tree. The Lagrange multipliers for this energy (using the constraint that the marginals agree) lead to standard belief propagation rules developed for trees both in the machine-learning and computational biology communities. Note that our model reduces to tree or chain structures when either $\mathcal{T}_i$ and $\mathcal{C}_j$ are empty, or equivalently, when the conditionals do not depend on one of those sets. In such a case, belief propagation is exact and equivalent to Felsenstein's algorithm.

When these rules are iteratively applied on the graphs with loops ($\mathcal{T}_i$ or $\mathcal{C}_j$ are not empty), the procedure in general may not even converge (although in our case it does). When it does converge, it converges to a local minimum of the Bethe free energy, which is often close to a local minimum of our target cost, the free energy. The result of the loopy belief propagation can be used, however, to estimate the free energy by substituting $q(h_j^i) = \phi(h_j^i)$ into (14).

Loopy belief propagation has scored significant success in the error-correcting-code community and it is believed to be extremely effective on loopy graphs with a large number of variables but low local connectivity, such as our model. It usually easily beats mean field methods, and the jury is out on how it compares with more sophisticated variational techniques in general. In the next section, we present new variational techniques for our model, that capture more correlations among variables than either loopy belief propagation or the mean field technique.

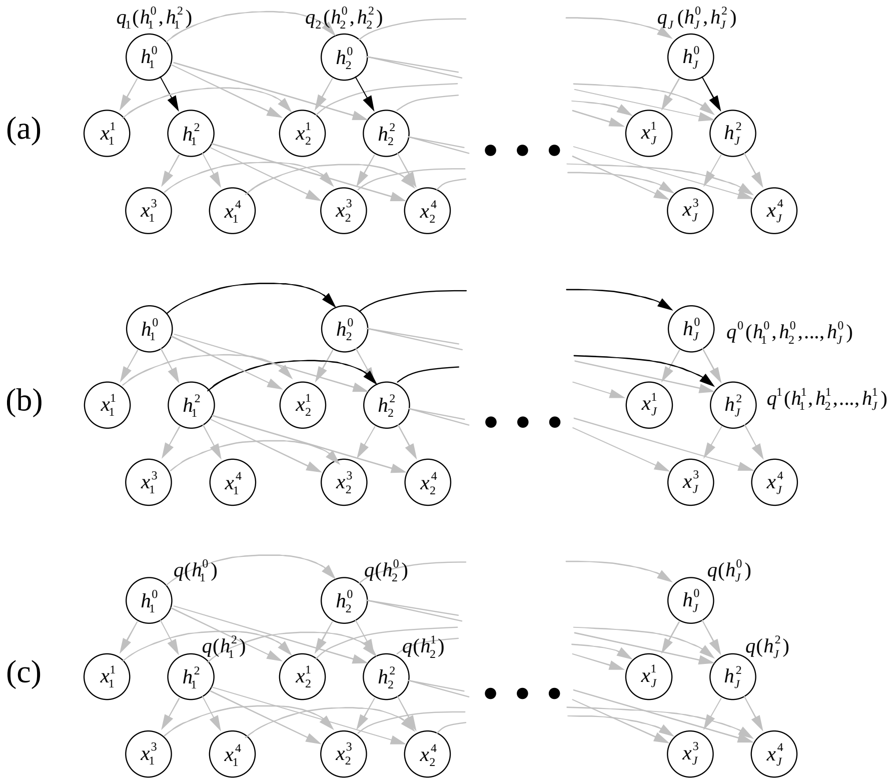


Figure 3: Variational approximations assume that conditioned on the data, the hidden variables form a simpler graph. For example, the product-of-trees posterior is shown in (a), the product of chains in (b), and the standard mean field technique in (c).

6 Structured variational approximations

In order to capture more correlations among the variables in the posterior, and at the same time avoid worrying about whether the marginal distributions agree, we can model the distribution q as a product of distributions defined on disjunct subsets of the model's hidden variables. In this section, we develop two such approximations. In the first approximation, the variables are grouped according to the index j—that is, each factor in q is a distribution over all variables in a single tree at site j. In the second approximation, we group variables according to the sequence index i. Figure 3 illustrates graphs for various q approximations. Variational inference can be seen as optimizing the agreement between the model and the simplified graph for particular data.

6.1 Product of trees

Under the assumption that the posterior can be factored into J different individual probability distributions, each defined over nucleotides in different sequences but at the same site j,

$$q = \prod_j q_j(\{h_j^i\}_{i=1}^I), \tag{16}$$

the free energy assumes a more tractable form

$$F = \sum_j \sum_{\mathbf{h}_j} q_j \log q_j - \sum_{i,j} \sum_{\mathbf{h}_j, \mathbf{h}_{k\in\mathcal{C}_j}} q_j \, [\prod_{k\in\mathcal{C}_j} q_k] \log p(s_j^i|\mathbf{pa}(s_j^i)), \tag{17}$$

where we use bold notation $\mathbf{h}_j = \{h_j^i\}_{i=1}^I$ to denote the set of all variables in the jth tree. This expression is easily arrived at using the fact that $\sum_{\mathbf{h}_\ell} q_\ell = 1$. Consequently, thus in $\sum_{\mathbf{h}} q \log p(s_j^i|\mathbf{pa}(s_j^i))$, the distributions that do not use s_j^i, or $\mathbf{pa}(s_j^i)$ drop out. Note again that for simplicity in notation s denotes both hidden and observed variables, but the posteriors q are defined only over hidden variables.

Each individual distribution q_j, defined on variables $\mathbf{h}_j = \{h_j^i\}_{i=1}^I$, is used in multiple terms in the above summation, as variables from $\mathbf{h}_j$ are sometimes used as parents and sometimes as children in the conditionals $\log p(s_j^i|\mathbf{pa}(s_j^i))$. As before, we can find the part of the free energy that depends on q_j:

$$\begin{aligned} F_{q_j} = & \sum_{\mathbf{h}_j} q_j \log q_j - \\ & \sum_{\mathbf{h}_j} q_j \sum_{\mathbf{h}_{k\in\mathcal{C}_j}} \left[\prod_{k\in\mathcal{C}_j} q_k\right] \sum_i \log p(s_j^i|\mathbf{pa}(s_j^i)) - \\ & \sum_{\mathbf{h}_j} q_j \sum_{k|j\in\mathcal{C}_k} \sum_{\mathbf{h}_{\mathcal{C}_k\setminus j}, \mathbf{h}_k} q_k \left[\prod_{\ell\in\mathcal{C}_k\setminus j} q_\ell\right] \sum_i \log p(s_k^i|\mathbf{pa}(s_k^i)). \end{aligned}$$

This can be rewritten in the form

$$F_{q_j} = \sum_{\mathbf{h}_j} q_j \log q_j - \sum_{\mathbf{h}_j} q_j \sum_i \log \phi(s_j^i, \mathbf{s}_j^{\mathcal{T}_i}), \tag{18}$$

where we define

$$\begin{aligned} \log \phi(s_j^i, \mathbf{s}_j^{\mathcal{T}_i}) = & \sum_{\mathbf{h}_{k\in\mathcal{C}_j}} \left[\prod_{k\in\mathcal{C}_j} q_k\right] \log p(s_j^i|\mathbf{pa}(s_j^i)) + \\ & \sum_{k|j\in\mathcal{C}_k} \sum_{\mathbf{h}_{\mathcal{C}_k\setminus j}, \mathbf{h}_k} q_k \left[\prod_{\ell\in\mathcal{C}_k\setminus j} q_\ell\right] \log p(s_k^i|\mathbf{pa}(s_k^i)). \end{aligned} \tag{19}$$

Note that if we assume that all relevant (neighboring) distributions but q_j are fixed, then (18) has exactly the form of the standard tree model (11). Thus, given the log potentials $\log \phi(s_j^i, \mathbf{s}_j^{\mathcal{T}_i})$, we can use the forward-backward algorithm (belief propagation on trees) to find q_j that optimizes F_{q_j} *exactly*, as discussed in Section 4. On the other hand, having computed the optimal q_j, it can be used to perform the necessary expectations in computation of the potentials (19) for the neighboring trees.

Thus the optimization of the free energy can be performed by the following algorithm:

- Initialize all distributions q_j to be uniform.
- LOOP
  - Keeping all other q_j fixed, update q_1 so as to minimize F.

  - Keeping all other q_j fixed, update q_2 so as to minimize new F.

    ...

  - Keeping all other q_j fixed, update q_J so as to minimize new F.

- END

Each step lowers the free energy and the loop is escaped once the change in the free energy becomes negligible.

Like the mean field method, and unlike the loopy belief propagation or the simple Markov-chain approximation [16], this algorithm comes with the guarantee that it will converge to the value of the free energy that bounds the negative log likelihood of the data. Unlike both the mean field and the loopy belief propagation technique, this algorithm captures in the posterior the entire marginal distribution on all tree variables for each site j, and thus, as we show later, provides a significantly better bound than other techniques.

6.2 Product of chains

Another similar way to approximate the posterior is to factor it into I different individual probability distributions, each defined over all nucleotides in one sequence,

$$q = \prod_i q^i(\{h_j^i\}_{j=1}^J). \tag{20}$$

The variational optimization technique that uses this approximation is derived in the same way, essentially just by switching i and j and $\mathcal{T}_i$ and $\mathcal{C}_j$ in the equations of the previous section.

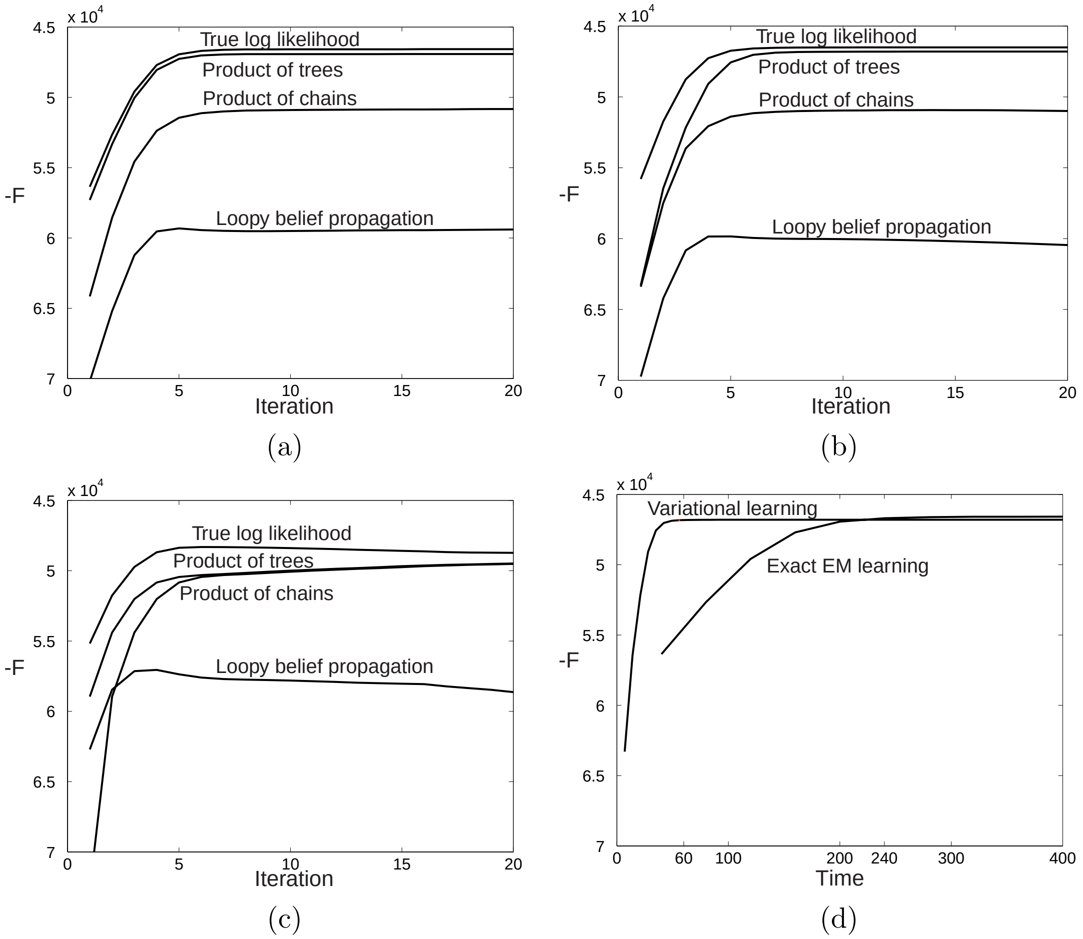
The advantage of this technique is that it groups many more variables into each distribution, as the length of the sequences can be of the order of hundred of thousands, while the number of different species is much smaller. However, the product-of-trees approximation focuses on the combinations of variables that are much more correlated, and (as we shall see in our experiments) is more accurate.

7 Learning

So far, we have discussed various ways of approximating the likelihood of the data. Each of these methods can be used when the task is to learn the optimal model parameters, for which the total log-likelihood of the data is maximized. The idea is again to minimize the free energy, as this minimization leads to the same result as log likelihood optimization, at least when the exact posterior is used. The generalized EM algorithm [12] used for learning consists of the following steps:

- Initialize all distributions q_j to be uniform, and choose random parameters $\boldsymbol{\theta}$.
- LOOP
  - E: Do several steps of variational optimization to update the posterior q so as to decrease $F(q, \boldsymbol{\theta})$ for the current guess on $\boldsymbol{\theta}$.
  - M: Find new parameters $\boldsymbol{\theta}$ so that $F(q, \boldsymbol{\theta})$ is decreased for the current guess on q.
- END (when F stops decreasing significantly)

In our experiments, the M step is performed by BFGS quasi-Newton optimization.



(a)

(b)

(c)

(d)

Figure 4: Quality of the bounds (a)-(c); and the computational cost comparison (d).

8 Experiments and discussion

In this section, we compare the performance of four different posterior approximations on the task of computing the log likelihood of the data as a model score: (1) variational inference using the product of trees approximation, (2) variational inference using the product of chains approximation, (3) loopy belief propagation, and (4) the simple Markov-chain approximation. The mean field and ICM approximations under perform the other techniques, and are omitted from this discussion.

As our main goal is to evaluate how close different approximations are to the *exact* inference, we also used an (expensive) technique to compute the exact log likelihood and posterior $p(\mathbf{h}|\mathbf{x})$. In this exact approach, we clique all observed nodes $\mathbf{x}_j$ and unobserved nodes $\mathbf{h}_j$, and consider them as new variables with a much larger configuration space. For example, when there are 5 hidden nodes in slice j in the model, the total number of possible configurations for $\mathbf{h}_j$ is 4^5. The modified model assumes a form of a single HMM; and we can estimate the true posterior in the form $\prod_j p(\mathbf{h}_j|\mathbf{h}_{j-1}, \mathbf{x})$ using the forward-backward algorithm, thus avoiding the brute force search over all 4^{I*J} configurations. Even when done in this manner, exact inference is still extremely slow, and that limited us to models with only three species at the leaves, hence two hidden sequences in the inner nodes of tree.

We ran tests on two subsets of data used in [16]. Both data sets correspond to trees with three species at the leaves. Set A consisted of sequences from cow, mouse and human; and set B had sequences from cow, pig, and dog. Sequences were of length 133Kb and 99k, respectively, and were taken from the region of the Cystic Fibrosis Transmembrane Conductance Regulator (CFTR) gene. The alignment used in [16] was used here as well.

Figure 4 illustrates the tightness (or looseness) of the various bounds. In Figure 4(a), we show exact log likelihood as well as the free energy of the variational and loopy-belief-propagation approximations during exact EM learning—that is, during parameter learning that uses an exact-inference E step. These curves illustrate how tight the various approximations are for a range of model parameters. In Figure 4(b), we show a similar graph, but for parameters obtained during generalized EM learning based on the product of trees approximation. In Figure 4(c), the plots are for the parameters obtained during generalized EM learning based on the product of chains approximation. In order to lower the total amount of computation, these experiments were performed on shorter 20Kb sequences from dataset A. Note that in each of these graphs, only one curve has to be monotonic by definition—the one that matches the technique used in the generalized EM learning. For the other curves, the only guarantee is that the curve corresponding to exact log likelihood must be above all others.

When tracking behavior of various bounds during exact EM iterations, we see that, for the product of trees approximation, the bound is much tighter than for the other two approximations. The difference between the bound and the true likelihood is due to the dependencies that are absent from the approximation but are present in the true posterior. The product of chains approximation, which is not capturing the correlations arising from evolution, is performing worse. Finally, the loopy belief propagation technique that captures only short range dependencies performs worst of all.

Also interesting to note, the approximation errors accumulate during EM learning, resulting in a much worse final result in the product of chains approximation than one would expect looking at Figure 4(a). This illustrates the importance of using an approximation that bounds the log likelihood as tightly as possible.

Figure 4(d) illustrates the computational gain that we obtain by using the product of trees approximation on this relatively small task. In this graph, we show the log likelihood estimate as the function of time during EM and variational EM learning. The computational gains are even more dramatic for a larger number of longer sequences. The complexity of the variational E-step is linear in the number of hidden variables, which in turn is linear in number of data points; the complexity of the exact EM is exponential in the number of nodes in the tree and linear in length of the aligned sequences.

Finally we compared the simple Markov-chain approximation to the other bounds available on the full sets A and B. Using U2S parameters that are at a local maximum for the simple Markov-chain approximation, we computed the following values:

	Likelihood	
Method	set A	set B
Exact EM	-296413	-217843
Variational	-299756	-219427
Siepel-Haussler	-300139	-220408

We see that the simple Markov-chain approximation method of likelihood computation underestimates the likelihood of data, compared to the product-of-trees variational bound.

	Equilibrium frequencies															
	0.09	0.05	0.07	0.08	0.07	0.05	0.006	0.07	0.06	0.04	0.05	0.05	0.07	0.06	0.07	0.1
	Rate matrix															
AA	-2.57	0.28	0.88	0.24	0.31	0.00	0.00	0.00	0.65	0.00	0.00	0.00	0.21	0.00	0.00	0.00
AC	0.35	-4.01	0.33	1.80	0.00	0.34	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.29	0.00	0.00
AG	1.48	0.27	-3.34	0.33	0.00	0.00	0.31	0.00	0.00	0.00	0.73	0.00	0.00	0.00	0.21	0.00
AT	0.24	1.10	0.33	-3.35	0.00	0.00	0.00	0.33	0.00	0.00	0.00	1.10	0.00	0.00	0.00	0.24
CA	0.34	0.00	0.00	0.00	-4.03	0.36	1.12	0.27	0.23	0.00	0.00	0.00	1.70	0.00	0.00	0.00
CC	0.00	0.35	0.00	0.00	0.23	-4.39	0.28	1.60	0.00	0.27	0.00	0.00	0.00	1.65	0.00	0.00
CG	0.00	0.00	1.04	0.00	**15.62**	0.83	-34.99	1.04	0.00	0.00	0.83	0.00	0.00	0.00	**15.62**	0.00
CT	0.00	0.00	0.00	0.33	0.21	0.73	0.31	-3.34	0.00	0.00	0.00	0.27	0.00	0.00	0.00	1.48
GA	1.43	0.00	0.00	0.00	0.31	0.00	0.00	0.00	-3.36	0.23	0.87	0.22	0.30	0.00	0.00	0.00
GC	0.00	1.51	0.00	0.00	0.00	0.35	0.00	0.00	0.35	-4.43	0.35	1.51	0.00	0.35	0.00	0.00
GG	0.00	0.00	1.60	0.00	0.00	0.00	0.28	0.00	1.65	0.27	-4.39	0.35	0.00	0.00	0.23	0.00
GT	0.00	0.00	0.00	1.80	0.00	0.00	0.00	0.33	0.29	0.90	0.34	-4.01	0.00	0.00	0.00	0.35
TA	0.33	0.00	0.00	0.00	1.09	0.00	0.00	0.00	0.33	0.00	0.00	0.00	-3.50	0.33	1.09	0.33
TC	0.00	0.22	0.00	0.00	0.00	0.87	0.00	0.00	0.00	0.23	0.00	0.00	0.30	-3.36	0.31	1.43
TG	0.00	0.00	0.27	0.00	0.00	0.00	1.12	0.00	0.00	0.00	0.36	0.00	1.70	0.23	-4.03	0.34
TT	0.00	0.00	0.00	0.24	0.00	0.00	0.00	0.88	0.00	0.00	0.00	0.28	0.21	0.65	0.31	-2.57

Figure 5: Equilibrium distribution of nucleotides and rate matrix estimated using product of trees method on sequence from region of the Cystic Fibrosis Transmembrane Conductance Regulator (CFTR) gene in human genome and homologous sequences from eight eutherian mammals. The matrix is scaled so that at the equilibrium the expected number of substitutions is one on a branch of length one. The rates corresponding to CpG effect are shown in bold.

9 Application

We applied our best approximation, the product of trees approximation, to the Cystic Fibrosis Transmembrane Conductance Regulator (CFTR) gene in human genome and homologous sequences from eight eutherian mammals: chimp, baboon, cow, pig, cat, dog, mouse, and rat. These sequences have been selected from non-coding regions. The length of an aligned sequence for each of the species was 162743 nucleotides. We ran our approximation with a near uniform Q matrix and obtained rate matrix estimates presented in Figure 8. We found that the estimated rate matrix $\mathbf{Q}$ had mutation rates from CG to TG (and from CG to CA, due to the strand symmetry of the U2S model)—a "CpG effect." Our observation is consistent with that of [16] who examined shorter 20kb subsequences from the same region. Note that, with a single-site evolutionary model as opposed to a dinucleotide model, a CpG effect cannot be captured by the rate matrix.

The branch lengths for the unrooted phylogenetic tree that we learned are shown in Figure 9 along with branch lengths for UNR-mononucleotide [19] and HKY-mononucleotide [8] models. The latter two models were learned using PAML [21]. Note that the branch lengths for the dinucleotide model are roughly twelve percent shorter than those for the mononucleotide models.

10 Conclusions and Future Work

We have introduced novel variational approximations tailored to learning phylogenies where nearby sites are not assumed to evolve independently. Using real data, we have found that these approximations outperform the mean-field approximation, an approximation based on loopy belief propagation, and the simple Markov-chain approximation. Furthermore, we have seen that the structured variational approximation that performs best is the one whose q distribution has the independencies of the classic Neyman–Felsenstein model. In addition, we have applied our approximation to data from the Cystic

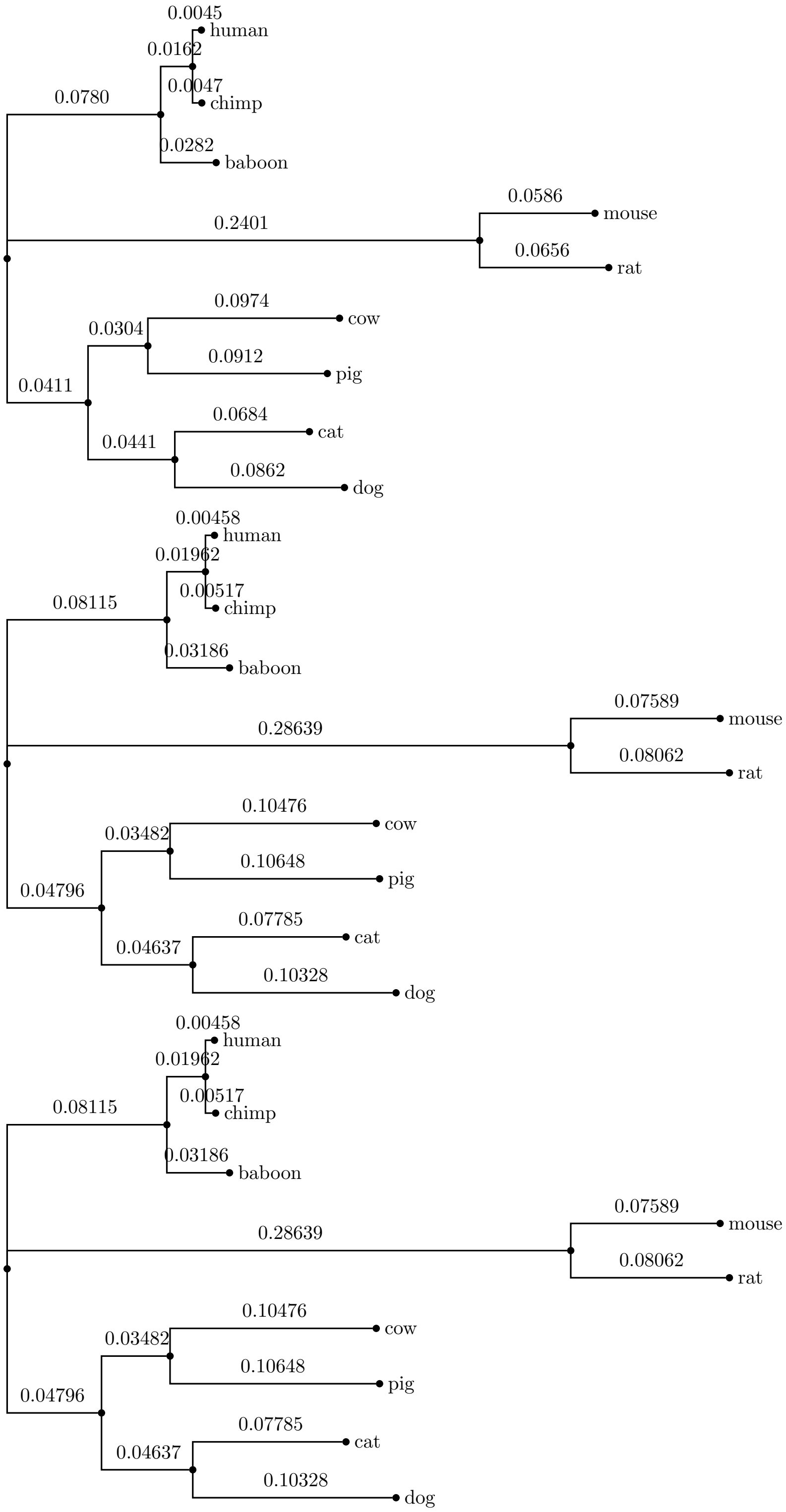


Figure 6: Tree branch lengths estimated using different models of evolution (top to bottom): U2S-dinucletide, UNR-mononucleotide, and HKY-mononucleotide. The product of trees variational approximation was used for estimation in the case of the U2S-dinucleotide model.

Fibrosis Transmembrane Conductance Regulator (CFTR) gene region across nine eutherian mammals to find a CpG effect.

The approximations that we have introduced in this paper can be applied to a variety of evolutionary models—for example, k-nucleotide models. For trinucleotide models ($k = 3$), we would expect to see higher mutation rates for silent mutations. For $k > 3$, we may find interesting long-range dependencies. Our approximations also can be applied to evolutionary models that include 'gap' or 'indel' letters in the alphabet. This extension combined with k-nucleotide models ($k \leq 6$) can model the evolution of micro satellite regions [4]. Such micro satellites evolve by addition or removal of tandem repeat units.

Finally, unlike the simple Markov-chain approximation, our approximations can be used to reconstruct ancestral sequences. Furthermore, it is possible that loopy belief propagation may outperform the variational approximations on this task. Further investigation is needed.

References

[1] P. F. Arndt, C. B. Burge, and T. Hwa. DNA sequence evolution with neighbor-dependent mutation. In G. Myers, S. Hannenhalli, S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the Sixth Annual International Conference on Computational Biology*, pages 32–38, New York, 2002. ACM.

[2] M. Averof, A. Rokas, K. H. Wolfe, and P. M. Sharp. Evidence for a high frequency of simultaneous double-nucleotide substitutions. *Science*, 287:1283–1286, 2000.

[3] G. Cooper. Computational complexity of probabilistic inference using Bayesian belief networks (Research note). *Artificial Intelligence*, 42:393–405, 1990.

[4] D. Dieringer and C. Schlötterer. Two distinct modes of microsatellite mutation processes: Evidence from the complete genomic sequences of nine species. *Genome Res.*, 13:2242–2251, 2003.

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

[6] J. Felsenstein. Evolutionary trees from DNA sequences. *J. Mol. Evol.*, 17:368–376, 1981.

[7] J. Felsenstein and G. A. Churchill. A hidden Markov model approach to variation among sites in rate of evolution. *Mol. Biol. Evol.*, 13:93–104, 1996.

[8] M. Hasegawa, H. Kishino, and T. Yano. Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, 22:160–174, 1985.

[9] J. L. Jensen and A.-M. K. Pedersen. Probabilistic models of DNA sequence evolution with context dependent rates of substitution. *Adv. Appl. Prob*, 32:499–517, 2000.

[10] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models.* MIT Press, 1999.

[11] L. Nakhleh, J. Sun, T. Warnow, C. Linder, B. Moret, and A. Tholse. Towards the development of computational tools for evaluating phylogenetic network reconstruction methods. *Proceedings of the Eighth Pacific Symposium on Biocomputing*, 8:315–326, 2003.

[12] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Kluwer, 1998.

[13] J. Neyman. Molecular studies of evolution: A source of novel statistical problems. In S.S. Gupta and J. Yackel, editors, *Statistical Decision Theory and Related Topics*, pages 1–27. Academic Press, New York, 1971.

[14] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[15] A.-M. K. Pedersen and J. L. Jensen. A dependent rates model and MCMC based methodology for the maximum likelihood analysis of sequences with overlapping reading frames. *Mol. Biol. Evol.*, 18:763–776, 2001.

[16] A. Siepel and D. Haussler. Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Molecular Biology and Evolution*, 10:1093, 2003.

[17] K. Strimmer and V. Moulton. Likelihood analysis of phylogenetic networks using directed graphical models. *Molecular Biology and Evoluation*, 17:875–881, 2000.

[18] S. Whelan, P. Liò, and N. Goldman. Molecular phylogenetics: State-of-the-art methods for looking into the past. *Trends Genet.*, 17:262–272, 2001.

[19] Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J. Mol. Evol.*, 39:306–314, 1994.

[20] Z. Yang. A space-time process model for the evolution of DNA sequences. *Genetics*, 139:993–1005, 1995.

[21] Z. Yang. PAML: A program package for phylogenetic analysis by maximum likelihood. *CABIOS*, 13:555–556, 1997.

[22] Z. Yang, N. Goldman, and A. Friday. Comparison of models for nucleotide substution used in maximum likelihood phylogenetic estimation. *Mol. Biol. Evol.*, 11:316–224, 1994.

[23] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.