BU-1329-MB

Bayesian Restoration of a Hidden Markov Chain with Applications to DNA Sequencing

Gary A. Churchill Betty Lazareva

January 13, 1997

Abstract

Hidden Markov models (HMMs) are a class of stochastic models that have proven to be powerful tools for the analysis of molecular sequence data. A hidden Markov model can be viewed as a black box that generates sequences of observations. The unobservable internal state of the box is stochastic and is determined by a finite state Markov chain. The observable output is stochastic with distribution determined by the state of the hidden Markov chain. We present a Bayesian solution to the problem of restoring the sequence of states visited by the hidden Markov chain from a given sequence of observed outputs. Our approach is based on a Monte Carlo Markov chain algorithm that allows us to draw samples from the full posterior distribution of the hidden Markov chain paths. The problem of estimating the probability of individual paths and the associated Monte Carlo error of these estimates is addressed. The method is illustrated by considering a problem of DNA sequence multiple alignment. The special structure for the hidden Markov model used in the sequence alignment problem is considered in detail. In conclusion we discuss certain interesting aspects of biological sequence alignments that become accessable through the Bayesian approach to HMM restoration.

1 Introduction

1.1 Hidden Markov Models

Hidden Markov models (HMMs) are a class of stochastic models that have proven to be useful in a wide range of applications for modeling highly structured sequences of data. Applications of HMMs to the problem of machine speech recognition have been reviewed by Juang and Rabiner (1991). Models for ion channel kinetics have been developed by Fredkin and Rice (1992). This paper will focus on HMMs that have proven to be useful in molecular biology applications. We introduce a Bayesian approach to problem of restoring the hidden states.

A hidden Markov model can be viewed as a black box that generates sequences of observations. The unobservable internal state of the box is stochastic and is determined by a finite state Markov chain. The observable outputs of the black box are stochastic with distribution determined by the current state of the hidden Markov chain. Let $\{s_t,\ t = 0, 1, 2, \ldots\}$ be an unobserved Markov chain on the state space $\{1, 2, \ldots, L\}$ and let $\{y_t,\ t = 0, 1, 2, \ldots\}$ be an observed process that takes values in the set $\{1, 2, \ldots, K\}$.

The restriction to discrete observations is not essential but it is adequate for the applications considered here. The observed data will be either DNA or protein sequences. A DNA sequence can be represented as a string of characters on the alphabet $\{A, C, G, T\}$, $K = 4$. The individual letters represent the different *bases* in the linear DNA molecule. In our example, we extend this alphabet to include the letter N and thus $K = 5$. A protein sequence can be represented as a string over a $K = 20$ letter alphabet in which letters represent the different amino acid types.

In more detail, an HMM with L hidden states and K observable outputs is specified by three sets of distributions. First is the *initial distribution* of the hidden Markov chain

$$\Pr(s_0 = i),\ i \in \{1, \ldots, L\}. \tag{1}$$

Second is the *transition distribution* of the hidden Markov chain as represented by the $L \times L$ matrix $\Lambda = [\lambda_{ij}]$ with elements

$$\lambda_{ij} = \Pr(s_{t+1} = j \mid s_t = i),\ i \in \{1, \ldots, L\},\ j \in \{1, \ldots, L\}. \tag{2}$$

Third is the set of *output distributions* of the hidden states as represented by the $L \times K$ matrix $\Pi = [\pi_{ij}]$ with elements

$$\pi_{ij} = \Pr(y_t = j \mid s_t = i),\ i \in \{1, \ldots, L\},\ j \in \{1, \ldots, K\}. \tag{3}$$

Both matrices Λ and Π are stochastic, *i.e.*, they are formed by nonegative numbers and their row sums are equal to one. Thus the parameter $\theta \equiv (\Lambda, \Pi)$ takes values in a compact set Θ which is a direct product of L L-dimensional and L K-dimensional simplexes.

The number of hidden states and their connectivity, the set of nonzero λ_{ij}, together define the *architecture* of an HMM. The choice of an architecture is typically driven by an application for which the HMM is intended. It is convenient to consider a minor variation on the basic setup, as follows. Along with the states that produce outputs, we consider two additional states that do not produce any output. We call these *begin* (**B**) and *end* (**E**). The rest will be referred to as "main" states. Without loss of generality we assume that the initial distribution is concentrated in the state **B**. Thus $\Pr(s_0 = B) = 1$. The state transition matrix Λ, whose dimension becomes $(L+2) \times (L+2)$, is modified as follows

1. The state **B** is unattainable from any state including itself, $\lambda_{iB} = 0$, for all i.

2. State **E** is absorbing so that $\lambda_{EE} = 1$ and is recurrent so there is a stopping time $n^* = \min\{k : s_k = E, k \geq 0\}$ such that $\Pr(n^* \leq \infty) = 1$.

3. The direct transition from state **B** to state **E** is not allowed, $\lambda_{BE} = 0$.

Introduction of the absorbing state **E** allows us to deal with finite realizations of the HMM up to the stopping time n^*. We put $n = n^* - 1$ and use the following notation

for the sequence of hidden states and the corresponding sequence of outputs

$$\begin{aligned} \mathbf{s} &\equiv s_1\, s_2\, \cdots\, s_n, \\ \mathbf{y} &\equiv y_1\, y_2\, \cdots\, y_n. \end{aligned}$$

The states $s_0 = \mathbf{B}$ and $s_{n+1} = \mathbf{E}$ will be supressed in the notation, except where they are explicitly needed below.

Suppose that we observe N independent realizations of an HMM. We will denote the set of observed outputs by

$$\mathbf{Y} \equiv \left\{ \begin{array}{lcl} \mathbf{y}_1 & = & y_{1,1}\, y_{1,2}\, \cdots, y_{1,n_1} \\ \vdots & & \vdots \\ \mathbf{y}_N & = & y_{N,1}\, y_{N,2}\, \cdots, y_{N,n_N} \end{array} \right\}.$$

Table 1 shows an example of six DNA sequences $(\mathbf{y}_1, \ldots, \mathbf{y}_6)$ that are the data for our analysis in section 4. The sequences of paths through the hidden Markov chain that produced $\mathbf{Y}$ will be denoted by

$$\mathbf{S} \equiv \left\{ \begin{array}{l} \mathbf{s}_1 = s_{1,1}\, s_{1,2}\, \cdots, s_{1,n_1} \\ \vdots \\ \mathbf{s}_N = s_{N,1}\, s_{N,2}\, \cdots, s_{N,n_N} \end{array} \right\}.$$

Our goal in this work is to develop a method of restoring the sequences of the paths $\mathbf{S}$ given the observed outputs $\mathbf{Y}$.

Hidden Markov models can have large parameter spaces because there may be many possible state transitions and because each state can have its own unique output distribution. Depending on the application, it may be desirable to allow all non–zero parameter values to vary freely. At the other extreme, we may require that some subsets of parameters take identical values. Constraints of this type are referred to as "tied" parameterizations. A less extreme form of combining information can be achieved by imposing a hierarchical model on the parameters in which sets of parameter values are assumed to be drawn from a common distribution. In

TAGACAGGNGCCCCACTGGAGGAATGAGGTCACCAACCAACCTTCAAAAACTT
TAGACAGGGNCCCACTGGAGGAATGAGGTCACCAACCAACCTTCAAAAACTT
TAGANAGGGCCTCCACTGGGGAAATGAAGGTACCNACCAACCTTCAAAACTT
TAGACCAGGNGCTCCACTGGAGGAATGAGGTCACCAACCAACCTTCAAAAACTT
TAGACAGGGCCTCCACTGGAGATNTGAGGTCACCAACCAACCTTCAAAAACTT
TAGACAGGGGCTCCACTGGAGGAATGAGGTCACCAACCAACCTTCAAAAACTT

Table 1: An unaligned set of DNA sequences

our example, we use both tied parameter constraints and hierarchical modeling to reduce the dimensionality of the parameter space.

1.2 Examples of HMMs

We present some examples of HMMs that have proven to be useful in molecular biology applications. It is worthwhile to consider two classes of architectures. First is the *recurrent* architecture in which any main state may be reached from any other main state. Second is the *left-to-right* architecture, in which the main states do not recurr. Of course, arbitrarily complex HMMs can be constructed with both recurrent and non–recurrent components. See, for example, White *et al.* (1994).

1.2.1 Two–state recurrent architecture

Consider a hidden Markov chain with two main states denoted by 0 and 1 and binary outputs $\{0, 1\}$. This two–state recurrent architecture is illustrated in Figure 1. Its transition probability matrix, defined on the extended state space $\{B, 0, 1, E\}$, is

$$\Lambda = \begin{bmatrix} 0 & \lambda_{B0} & \lambda_{B1} & 0 \\ 0 & \lambda_{00} & \lambda_{01} & \lambda_{0E} \\ 0 & \lambda_{10} & \lambda_{11} & \lambda_{1E} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The output distribution is specified by

$$\Pi = \begin{bmatrix} \pi_{00} & \pi_{01} \\ \pi_{10} & \pi_{11} \end{bmatrix}.$$

This HMM generates nonhomogeneous binary sequences that consist of homogeneous regions of two types, with distinct frequencies of zeros and ones. This model and the more general L–state, K–output recurrent model were applied by Churchill (1989, 1992) to identify regions with distinct functions in DNA sequences based on differences in local base frequencies.

1.2.2 Left–to–right architectures

An example of a left–to–right architecture with a state space $\{I_1, I_2, M_1, \ldots, M_k\}$ is shown in Figure 2. This HMM is analogous to a model proposed by Lawrence *et al.* (1993) for the purpose of locating conserved *pattern* elements in a set of otherwise unrelated protein sequences. Notice that there is only one possible transition (that occurs with probability one) out of each of the states $M_1, \ldots, M_{k-1}$ and thus a typical HMM path will be of the form $(I_1)^i, M_1, \ldots, M_{k-1}, (I_2)^j$. This model will generate a block of k adjacent amino acids with a characteristic pattern as defined by the output distributions of the states M_i. The pattern is located in a random sequence background with amino acid frequencies determined by the output distributions of the states I_1 and I_2. The output parameters of the I–states are tied to produce identical background distributions before and after occurrence of the pattern. In Lawrence *et al.*, the prior distribution of the pattern location is explicit. In the HMM, the *a priori* lengths of the sequences before and after the pattern are geometric with parameters $\lambda_{I_1 M_1}$ and $\lambda_{I_2 I_2}$ respectively. Variations on this model can easily be developed to allow for multiple occurences (or absence) of the pattern in some of the sequences. A Gibbs sampling algorithm for the model of Lawrence *et al.* (1993) has been described by Liu *et al.* (1995b). A similar algorithm could be based on the methods described in this paper.

1.2.3 Mutation–Deletion–Insertion Models

A more elaborate example of a left-to-right HMM, the Mutation–Deletion–Insertion (MDI) architecture, is shown in Figure 3. This model has become a very popular tool for the problem of aligning multiple protein sequences (Krogh *et al.*, 1994; Baldi *et al.*, 1994; Eddy, 1996). The MDI hidden Markov chain has three types of main states. The backbone of the chain consists of *mutation* states $\{M_1, M_2, \ldots, M_L\}$. Each mutation state M_i has a corresponding *deletion* state D_i. Following the state **B** there is an *insertion* state I_0 and following each of the mutation states M_i there is an insertion state I_i. When the Markov chain visits any of the states M_i or I_i, it produces an output y according to $\Pr(y \mid M_i)$ or $\Pr(y \mid I_i)$. The states D_i are silent and do not produce any output.

The presence of silent states in the MDI introduces a minor complication into our description of these HMMs. It was implicit in our earlier definition of an HMM that there is a one–to–one correspondence between outputs and hidden states. However in the MDI model, as it is typically implemented, there may be hidden states (D–states) that are visited but have no corresponding output. We note that the output of an MDI model can viewed as the output of a standard HMM consisting of only M–states and I–states. This MI chain is embedded within the MDI chain and can be constructed by simply removing the D–states. The architecture of the MI chain includes additional transitions to replace the removed D–states. Unfortunately the additional transition parameters must be constrained in a rather complicated fashion to recover exactly the original MDI model. The output distributions of the MI model are identical to those of the MDI model. It follows that results derived for standard HMMs apply equally to MDI models.

1.3 Overview

The paper deals with the problem of restoring the hidden state sequences $\mathbf{S}$ for given data $\mathbf{Y}$ from Bayesian prospective. We consider a Gibbs sampler that samples from

the joint *a posteriori* distribution of $\mathbf{S}$ and θ. The non-trivial part of it, the conditional sampling of $\mathbf{S}$ given the parameter and data, was suggested in Churchill (1995). The data augmentation step (Tanner and Wong, 1987), *i.e.*, sampling $\mathbf{S}$ immediately from its conditional distribution, distinguishes our algorithm from that suggested by Robert *et al.* (1993) in which $\mathbf{S}$ is sampled componentwise. Related sampling algorithms are described in Eddy (1995) and in Liu *et al.* (1995a). Another approach (not HMM based) to studying the posterior distribution on multiple alignments is given by Allison *et al.* (1994).

The remainder of this paper is organized as follows. We first consider the problems of parameter estimation and state restoration for general HMMs. In section 2 we briefly review the maximum likelihood approach and present a Bayesian approach to these problems. A Monte Carlo Markov chain algorithm for restoring hidden state sequences is described in section 2.2.2. In section 3, we consider the special structure of the MDI model and use this to derive a more effecient sampling procedure. In section 4 we consider an example using DNA sequence data. We close with a brief discussion of the practicality of the Bayesian restoration method.

2 HMM restoration

2.1 Maximum Likelihood Approach

In the maximum likelihood approach to HMM restoration, no prior information on the parameter θ is assumed and the inference problems of parameter estimation and state restoration are addressed by first finding an MLE for θ and then restoring $\mathbf{S}$ conditionally given the estimated value.

The likelihood for θ takes the form

$$\begin{aligned} \Pr(\mathbf{Y} \mid \theta) &= \prod_{i=1}^{N} \Pr(\mathbf{y}_i \mid \theta) \\ &= \prod_{i=1}^{N} \sum_{\mathbf{s}_i} \Pr(\mathbf{y}_i \mid \mathbf{s}_i, \Pi) \Pr(\mathbf{s}_i \mid \Lambda) \end{aligned} \tag{4}$$

where

$$\Pr(\mathbf{y}_i \mid \mathbf{s}_i, \Pi) = \pi_{s_1, y_{i,1}} \cdot \pi_{s_2, y_{i,2}} \cdots \pi_{s_{n_i}, y_{i,n_i}} \tag{5}$$

and

$$\Pr(\mathbf{s}_i \mid \Lambda) = \lambda_{B s_{i,1}} \cdot \lambda_{s_{i,1} s_{i,2}} \cdots \lambda_{s_{i,n_i} E}. \tag{6}$$

In general the likelihood is intractable for direct maximization and the problem of maximum likelihood estimation is solved by the *Baum-Welch* algorithm (Baum and Petrie, 1966; Rabiner, 1989) which is an EM algorithm (Dempster et al., 1977) for HMMs. This algorithm is known to converge to a local maximum of the likelihood function (Baum *et al.*, 1970; Leroux, 1992). Many applications (*e.g.*, Krogh *et al.*, 1994) use a segmental k-means algorithm (Juang *et al.*, 1990) also known as "Viterbi training" in which $\Pr(\mathbf{Y}, \mathbf{S} \mid \theta)$ is maximized with respect to $\mathbf{S}$ and θ simultaneously. The two estimators of θ are generally rather close (Merkav *et.al*, 1991) however the segmental k-means algorithm is less computationally demanding.

Having obtained some parameter estimate $\widehat{\theta}$, we can restore $\mathbf{S}$ by independently restoring each $\mathbf{s}_i$. A global restoration finds a most probable path under $\Pr\left(\cdot \mid \mathbf{y}_i, \widehat{\theta}\right)$ using the Viterbi algorithm (Viterbi, 1967). Local restoration methods find the most probable state at each moment t. Both approaches to the path restoration problem have a certain weakness: the final solution is based on the point estimator of θ and fails to take into account other "reasonable" values of θ. Furthermore, it may be of interest to find not only an *optimal* multiple path but also to have access to reasonable alternative restorations. These concerns motivate our choice of the Bayesian paradigm for multiple path restoration.

2.2 Bayesian approach

We assume a prior distribution $\Pr(\theta)$ for the parameter $\theta \equiv (\Lambda, \Pi)$ so that the posterior distribution of the pair $(\mathbf{S}, \theta)$ is

$$\Pr(\mathbf{S}, \theta \mid \mathbf{Y}) \propto \Pr(\theta) \prod_{i=1}^{N} \Pr(\mathbf{y}_i \mid \mathbf{s}_i, \Pi) \Pr(\mathbf{s}_i \mid \Lambda) \tag{7}$$

where the last two terms are defined in (5) and (6), respectively. Integrating out the parameter θ in (7) we obtain the marginal posterior $\Pr(\mathbf{S} \mid \mathbf{Y})$ that will be our primary interest. Similarly, summing over all multiple paths, we obtain the marginal posterior of $\Pr(\theta \mid \mathbf{Y})$. These marginal posterior distributions are not practically computable, in part because of unassessable normalizing constants.

2.2.1 An MCMC algorithm

The following lemma, which gives a way to sample from the joint distribution $\Pr(\mathbf{S}, \theta \mid \mathbf{Y})$, is a trivial consequence of a Gibbs algorithm (Geman and Geman, 1984; Gelfand and Smith, 1990).

Lemma *The following iterative procedure generates a Markov chain*

$$\{(\mathbf{S}^m, \theta^m); m = 1, 2, \ldots\}$$

with stationary distribution $\Pr(\mathbf{S}, \theta \mid \mathbf{Y})$ *as* $m \to \infty$. *Starting from an initial value* θ^0, *iterate the two steps*

1. *For each* $i = 1, \ldots, N$ *independently sample* $\mathbf{s}_i^{m+1} \sim \Pr(\cdot \mid \mathbf{y}_i, \theta^m)$ *and*
2. *Sample* $\theta^{m+1} \sim \Pr(\cdot \mid \mathbf{Y}, \mathbf{S}^{m+1})$.

Corollary *One can estimate the posterior expectation of any function* $f(\mathbf{S}, \theta)$ *by taking the sample mean* $\widehat{f}_M = \frac{1}{M} \sum_{i=1}^{M} f(\mathbf{S}^i, \theta^i)$. *In particular, choosing* $f(\mathbf{S}, \theta)$ *to be an indicator function of the multiple path* $\mathbf{R}$ *we can estimate the posterior probability* $\Pr(\mathbf{R} \mid \mathbf{Y})$.

Notation. In the next two subsections we will describe algorithms that accomplish steps 1 and 2. First we introduce some notation. Let $\mathbf{x} = (x_1, ..., x_v)^T$ and $\mathbf{z} = (z_1, ..., z_v)^T$ be any two vectors then $\mathbf{x} \star \mathbf{z} \equiv (x_1 \cdot z_1,\ x_2 \cdot z_2, ..., x_v \cdot z_v)^T$ and $\mathbf{x}^{\mathbf{z}} \equiv x_1^{z_1} x_2^{z_2} ... x_v^{z_v}$. Let $|\mathbf{x}| \equiv |x_1| + |x_2| + ... + |x_v|$ denote l_1-norm of vector $\mathbf{x}$ and $A_{\cdot w} \equiv (a_{1w}, a_{2w}, ..., a_{uw})^T$ and $A_{w\cdot} \equiv (a_{w1}, a_{w2}, ..., a_{wv})$ denote, respectively, the column

and the row of a $u \times v$ matrix A corresponding to index w. We write $\mathbf{z}_{[s,t]} \equiv z_s z_{s+1} \ldots z_{t-1} z_t$, for any $1 \leq s \leq t$ to denote a subsequence of a sequence $\mathbf{z}$. Finally, the notation $\mathrm{MN}_1(\mathbf{p})$ is used to denote the multinomial distribution with parameters $\mathbf{p} = (p_1, p_2, ..., p_v)$.

2.2.2 Path sampler

Within this subsection the parameter $\theta = (\Pi, \Lambda)$ is fixed. We consider only a single sequence of observations $\mathbf{y} = y_1, y_2, \ldots, y_n$ generated by a path $\mathbf{s} = s_1, s_2, \ldots, s_n$ because multiple paths can be sampled independently.

The optimal nonlinear filter $\mathbf{f}(t) = (f_B(t), f_1(t), f_2(t), \ldots . f_L(t), f_E(t)))$, where $f_i(t) = \Pr\left(s_t = i \,|\, \mathbf{y}_{[1,t]}\right)$, $t = 0, 1, ..., n$ is given by the recursion (Stratanovich, 1960; Churchill 1989)

$$\mathbf{f}(t+1) = \frac{\Pi_{\cdot y_{t+1}} \star [\Lambda^T \mathbf{f}(t)]}{|\Pi_{\cdot y_{t+1}} \star [\Lambda^T \mathbf{f}(t)]|}, \tag{8}$$

with initial condition $\mathbf{f}(0)$= (1,0,0,....0). A non–normalized linear filtration is given by (Elliot *et al.*, 1994)

$$\mathbf{f}^*(t+1) = \Pi_{\cdot y_{t+1}} \star [\Lambda^T \mathbf{f}^*(t)] \tag{9}$$

with initial condition $\mathbf{f}^*(0) = (1, 0, 0, ...0)$. Comparison of (8) and (9) shows that $\mathbf{f}^*(t) = c(t)\mathbf{f}(t)$ for

$$c(t) = |\Pi_{\cdot y_1} \star [\Lambda^T \mathbf{f}(0)]| \times |\Pi_{\cdot y_2} \star [\Lambda^T \mathbf{f}(1)]| \times \cdots \times |\Pi_{\cdot y_t} \star [\Lambda^T \mathbf{f}(t-1)]| \tag{10}$$

In the following theorem, the filtration in (8) or (9) is used to obtain samples from the distribution $\Pr(\mathbf{S} \,|\, \mathbf{Y}, \theta)$. A proof is provided in Appendix A.

Theorem *Let* $\mathbf{s}^* = (s_1^*, s_2^*, ..., s_n^*)$ *be defined by the following recursion. Set* $s_{n+1}^* = E$, *then for* $t = n, n-1, ..., 1$

$$s_{t-1}^* \sim \mathrm{MN}_1\left(\frac{\mathbf{F}(t-1) \star \Lambda_{\cdot s_t^*}}{|\mathbf{F}(t-1) \star \Lambda_{\cdot s_t^*}|}\right) \tag{11}$$

where $F(t)$ *is either of* $f(t)$ *or* $f^*(t)$. *Then* $\mathbf{s}^* \sim \Pr(\cdot \,|\, \mathbf{Y}, \theta)$.

Thus to sample a path, we first solve the forward equations (8) or (9) and then sample backwards. This algorithm is analogous to the Viterbi algorithm in that it samples a single path on the backward pass. However the path is stochastic and thus in repeated iteration will explore more of the space of possible restorations than the determinstic Viterbi algorithm. It is interesting to compare the two algorithms. The forward equation for Viterbi algorithm takes the form

$$\tilde{f}_k(t+1) = \pi_{ky_{t+1}} \max_i [\lambda_{ik} \tilde{f}_i(t)] \tag{12}$$

with initial condition $\tilde{\mathbf{f}}(0) = (1, 0, 0, ...0)$. The backward Viterbi procedure is defined by the following recursion. Set $\tilde{s}_{n+1} = E$. Then for $t = n-1, \ldots, 1$

$$\tilde{s}_t = \text{argmax}_i \{\lambda_{i\tilde{s}_{t+1}} \tilde{f}_i(t)\}. \tag{13}$$

In our algorithm, the forward pass operation sums over all possible paths, whereas the Viterbi algorithm seeks an optimal path. On the reverse pass, our algorithm samples the next state whereas the Viterbi algorithm chooses the path that generated the optima on its forward pass. Thus the sampling algorithm retains the computational efficiency of the Viterbi algorithm but it explores a wider range of paths.

2.2.3 Parameter sampler

The u-dimensional Dirichlet distribution $D(\mathbf{a})$ with parameter $\mathbf{a} = (a_1, a_2, ..., a_u)$, $a_i \geq 0$ is defined on the u-dimensional simplex $\{\mathbf{x} = (x_1, x_2, ..., x_u) : |\mathbf{x}| = 1, x_i \geq 0\}$ and has density $d(\mathbf{x}; \mathbf{a}) = A(\mathbf{a})^{-1} \mathbf{x}^{\mathbf{a}-\mathbf{1}}$, where $A(\cdot)$ is the normalizing constant. If every row of matrices Λ and Π is distributed *a priori* according to Dirichlet distribution with certain parameters, the posterior distribution of the rows will also be Dirichlet but with shifted parameters. This follows from the conjugacy of Dirichlet and multinomial distributions (Robert, 1994, p. 103). A similar conjugacy property holds when the prior distribution is a Dirichlet mixture.

The following lemma describes a method for sampling from the conditional posterior distribution $\Pr(\theta \mid \mathbf{Y}, \mathbf{S})$. It involves augmented data sufficient statistics for θ, namely, matrices $C^{\Lambda} \equiv [c_{ij}^{\Lambda}]$ and $C^{\Pi} \equiv [c_{ij}^{\Pi}]$, where c_{ij}^{Λ} is the number of transitions to j state from i state and c_{ij}^{Π} is the number of outputs j from state i. When some parameter values are tied, the dimensions of the sufficient statistics can be reduced.

Lemma *Let the rows of matrices* Π *and* Λ *be a priori independently distributed according to Dirichlet distribution* $\Lambda_{i\cdot} \sim D(\mathbf{a}_i^{\Lambda})$, $i = 0, 1, 2, .., L$ *and* $\Pi_{i\cdot} \sim D(\mathbf{a}_i^{\Pi})$, $i = 1, 2, ..., L$. *Then the posterior distribution* $\Pr(\theta \mid \mathbf{Y}, \mathbf{S})$ *is a product of independent Dirichlet distributions over the rows of the matrices* Λ *and* Π, *where i-th row is distributed according to* $\Lambda_{i\cdot} \sim D(\mathbf{a}_i^{\Lambda} + C_{i\cdot}^{\Lambda})$, *or* $\Pi_{i\cdot} \sim D(\mathbf{a}_i^{\Pi} + C_{i\cdot}^{\Pi})$.

3 The MDI Model

In this section, we develop a detailed specification of an MDI model. This model is applied to study the posterior distribution of a DNA sequence alignment in section 4. We begin with a brief description of the DNA sequencing problem. More detailed descriptions can be found in Hunkapillar *et al.* (1991) and Churchill (1995).

3.1 DNA sequence alignment

We have a collection of DNA sequences that are independently copied from a common *prototype* sequence, $\mathbf{r} = r_1, \ldots, r_L; r_i \in \{A, C, G, T\}$, by a process that introduces errors in the form of *substitutions*, *deletions* and *insertions*. Each realization, $i = 1, \ldots, N$, of the MDI chain will generate a sequence $\mathbf{y}_i$ with elements $y_{ij} \in \{A, C, G, T, N\}$. The output character N is sometimes generated by DNA sequencing devices to represent ambiguous determination of a base. Each M–state in the MDI chain is associated with an element of the prototype sequence, *i.e.*, M_i is associated with r_i. This association will affect the output distribution of the M–state. For example, if the state M_i is associated with $r_i = A$, the most likely

output of state M_i is the letter A. A substitution error occurs when the output is a letter other than A. A deletion error occurs when the state D_i is visited, thus bypassing M_i, and no letter is generated as output. An insertion error occurs when the state I_i is visited thus generating extraneous letters in the output sequence. To summarize, a visit of D_i state results in a deletion of r_i in the copying process; k successive visits of I_i state result in an insertion of k letters after i-th position in the prototype; a visit of M_i state results in copying r_i with possible substitution error.

Restoration of $\mathbf{s}_i$ establishes a correspondence between the elements of $\mathbf{y}_i$ and the states of the MDI model. Furthermore, the multiple path restoration of $\mathbf{S}$ establishes a correspondence among all elements of all the DNA sequences via their correspondence with the M–states. This correspondence is a *multiple sequence alignment* (Waterman, 1995) and our goal here is to study its probability distribution.

3.2 Parameter Constraints and Prior Distributions

The dimensionality of the parameter space for an unconstrained MDI model can be very high even for models of modest size. We apply two different techniques for handling the high dimensionality of the parameter space. The output distributions $\tilde{\Pi}$ will be handled using a hierarchical model and the state transition parameters $\tilde{\Lambda}$ will be tied. The output parameters of M–states in our models are drawn from a common Dirichlet mixture distribution and the output parameters of the I–states are drawn from a common Dirichlet distribution. The transition parameters are tied in such a way that the probability of a deletion is constant and the probability of an insertion is also constant across the entire hidden Markov chain. These constraints appear to be reasonable as a first approximation for the DNA sequencing problem. In general the form of constraints on the model parameters should be carefully considered in the context of the application. Any number of variations on the parameter constraints and prior distributions are possible. We have chosen this particular combination to illustrate the method. We note that Dirichlet mixture

distributions have proven to be effective in protein sequence applications (Sjölander *et al.*, 1996). The choice of a prior distribution and its influence on the alignment are discussed in our example.

The output probabilities that correspond to state M_i form the i-th row of matrix $\tilde{\Pi}$. We assume that the prototype sequence $\mathbf{r} = r_1, r_2, \ldots, r_L$ (see Section 1.2.3) is *i.i.d.* with known letter frequencies $\alpha_s, s \in \{A, C, G, T\}$ and that the conditional prior distribution of $\tilde{\Pi}_{i\cdot} \equiv (\pi_{iA}, \pi_{iC}, \pi_{iG}, \pi_{iT}, \pi_{iN})$, given r_i, is Dirichlet with parameter $\mathbf{a}_{r_i} \equiv (a_{r_iA}, a_{r_iC}, a_{r_iG}, a_{r_iT}, a_{r_iN})$. However the prototype symbol r_i is unknown and the (unconditional) prior distribution of $\tilde{\Pi}_{i\cdot}$ is a mixture of four distributions

$$\Pi_{i\cdot} \sim \alpha_A D(\mathbf{a}_A) + \alpha_C D(\mathbf{a}_C) + \alpha_G D(\mathbf{a}_G) + \alpha_T D(\mathbf{a}_T),\ i = 1, \ldots, L. \qquad (14)$$

The output probabilities that correspond to states I_i are assumed to be identical for every $i = 0, 1, ..., L$ and form the 0-th row of matrix $\tilde{\Pi}$. Their prior distribution is Dirichlet with parameter $\mathbf{a}_I = (a_{IA}, a_{IC}, a_{IG}, a_{IT}, a_{IN})$.

The state transition probabilities are the same from all M states, as well as all I and D states, and are summarized in the following stochastic matrix

$$\tilde{\Lambda} = \begin{Bmatrix} \lambda_{MM} & \lambda_{MD} & \lambda_{MI} \\ \lambda_{DM} & \lambda_{DD} & \lambda_{DI} \\ \lambda_{IM} & \lambda_{ID} & \lambda_{II} \end{Bmatrix}.$$

The transition probabilities have a Dirichlet prior

$$(\lambda_{sM}, \lambda_{sD}, \lambda_{sI}) \sim D(\mathbf{b_s} = (b_{sM}, b_{sD}, b_{sI})\ ), s \in M, D, I. \qquad (15)$$

Observe that the most informative component in the above parametrization is the unknown prototype $\mathbf{r}$. As will be shown later it is convenient to include it in the set of parameters and to consider $\theta \equiv (\mathbf{r}, \tilde{\Pi}, \tilde{\Lambda})$.

3.3 Path sampler in the MDI model

The special structure of the MDI model allows for a computationally efficient variation of the filtration and resampling algorithms. In this section, we consider a single

observation $\mathbf{y}$ and will supress the double subscript.

First, we note that there is a one-to-one correspondence between the paths of the HMM that could have generated an observation $\mathbf{y} = y_1, y_2, ..., y_n$ and the paths from $(0,0)$ to (L,n) on the directed graph showed in Figure 4. Indeed, let $(B, s_1, s_2, ..., s_q, E)$ be any such path. Notice that the total number of M and I states in this path equals n, while the total number of M and D states equals L. We define a sequence of binary vectors $\mathbf{e} = (e_1, e_2, ..., e_q)$, $e_i \in \{\downarrow \equiv (1,0), \searrow \equiv (1,1), \rightarrow \equiv (0,1)\}$, such that, for $i = 1, ..., q$,

$$e_i = \begin{cases} \searrow & \text{if } s_i \text{ is an } M\text{-state,} \\ \downarrow & \text{if } s_i \text{ is a } D\text{-state} \\ \rightarrow & \text{if } s_i \text{ is an } I\text{-state.} \end{cases}$$

This sequence of binary vectors naturally defines a path on the graph, where the k-th vertex is given by $\sum_{i=0}^{k} e_i$. It is clear that this correspondence is one-to-one and that the graph path terminates in (L,n), *i.e.*, $\sum_{i=0}^{q} e_i = (L,n)$. Thus the problem of sampling $\mathbf{s}$ can be substituted by the problem of sampling $\mathbf{e}$.

The *path sampler* can be formulated in terms of $[\mathbf{p}_{ij}] \equiv [(p_{ij}^M, p_{ij}^D, p_{ij}^I)], i = 0, 1, ..n;\ j = 0, 1, ..., L$, where $p_{i,j}^s, s \in \{M, D, I\}$ is the probability that the chain visits a total of i M-states plus D-states with the last visited state being s and generates output $y_1, ..., y_j$. It is easy to verify that the matrix $[\mathbf{p}_{ij}]$ can be obtained by the following recursion for $i > 0$ and $j > 0$

$$\begin{aligned} p_{i,j}^M &= \pi_{iy_j} \mathbf{p}_{i-1,j-1} \tilde{\Lambda}_{\cdot M}, \\ p_{i,j}^D &= \mathbf{p}_{i-1,j} \tilde{\Lambda}_{\cdot D}, \\ p_{i,j}^I &= \pi_{0y_j} \mathbf{p}_{i,j-1} \tilde{\Lambda}_{\cdot I}, \end{aligned}$$

with boundary conditions

$$\mathbf{p}_{0,0} = (1,0,0); \mathbf{p}_{i,0} = (0, (\lambda_{DD})^i, 0); \mathbf{p}_{0,j} = (0, 0, (\lambda_{II})^j \prod_{k=1}^{j} \pi_{0y_j}).$$

The following lemma is the analogue of the theorem in 2.2.2 and can be found in Churchill (1995).

Lemma 4.1. *Let* $\mathbf{e}^* = (e_q^*, e_{q-1}^*, ..., e_1^*)$, $e_m^* \in \{\searrow, \downarrow, \rightarrow\}$. *be defined by the following recursion. Initialize* $(i,j)^0 = (L,n)$ *and* $m = 1$. *Then iterate the steps*

1. $e_m^* \sim \mathrm{MN}_1\left(\frac{\mathbf{P}(i,j)^m}{|\mathbf{P}(i,j)^m|}\right)$, *where the components of* $\mathbf{p}$ *correspond to* $\{\searrow, \downarrow, \rightarrow\}$,
2. $(i,j)^m = (i,j)^{m-1} - e_m^*$

for $m = 1, 2, \ldots$ *until* $(i,j)^m = (0,0)$. *Then* $\mathbf{e}^* \sim \Pr(\cdot \mid \mathbf{y}, \theta)$.

3.4 Parameter sampler

Given $\mathbf{Y}$ and $\mathbf{S}$, sufficient statistics for $\theta = (\mathbf{r}, \tilde{\Pi}, \tilde{\Lambda})$ form matrices $\tilde{C}^{\Lambda} \equiv [c_{st}], s \in \{M, D, I\}, t \in \{M, D, I\}$ and $\tilde{C}^{\Pi} \equiv [c_{iy}]$, $i = 0, 1, ..., L$, $y \in \{A, C, G, T, N\}$, where c_{st}, is the total number of transitions from state s to state t; $c_{0,y}$ is the total number of outputs of letter y from all I-states; c_{iy} is the total number of outputs of letter y from M_i-state, $i = 1, 2, ..., L$. Thus

$$\Pr(\theta \mid \mathbf{Y}, \mathbf{S}) \propto \prod_{s,t \in \{M,D,I\}} (\lambda_{st})^{b_{st}+c_{st}} \prod_{y \in \{A,C,G,T,N\}} (\pi_{0y})^{a_{0y}+c_{0y}} \prod_{i=1}^{L} \left(\alpha_{\mathbf{r}(i)} \prod_{y \in \{A,C,G,T,N\}} (\pi_{iy})^{a_{\mathbf{r}(i)y}+c_{iy}} \right). \tag{16}$$

We obtain the analogue of the lemma in 2.2.3.

Lemma *Let the parameter* $\theta \equiv (\mathbf{r}, \tilde{\Lambda}, \tilde{\Pi})$ *be distributed in accordance to (14) and (15), where* $\vec{\alpha} \equiv \{\alpha_A, \alpha_C, \alpha_G, \alpha_T\}$ *is known. Then the following two-stage sampling will generate samples from the Dirichlet mixture distribution* $\Pr(\theta \mid \mathbf{Y}, \mathbf{S})$

1. *Sample the prototype sequence* $\mathbf{r} = r_1, r_2, \ldots, r_L$ *independently according to*

$$r_i \sim \mathrm{MN}_1\left(\frac{\vec{\alpha} \star \mathbf{A}_i}{|\vec{\alpha} \star \mathbf{A}_i|}\right), \tag{17}$$

where $\mathbf{A}_i \equiv (A(\mathbf{a}_A + C_{i\cdot}), A(\mathbf{a}_C + C_{i\cdot}), A(\mathbf{a}_G + C_{i\cdot}), A(\mathbf{a}_T + C_{i\cdot}))$ *and* $A(\cdot)$ *is the Dirichlet normalizing constant.*

2. Then sample

$$\begin{aligned}
\tilde{\Lambda}_{s\cdot} &\sim D(\mathbf{b}_{s\cdot} + \tilde{C}^{\Lambda}_{s\cdot}),\ s \in \{M, D, I\} \\
\tilde{\Pi}_{0\cdot} &\sim D(\mathbf{a}_{r_0} + \tilde{C}^{\Pi}_{0\cdot}), \\
\tilde{\Pi}_{i\cdot} &\sim D(\mathbf{a}_{r_i} + \tilde{C}^{\Pi}_{i\cdot}),\ i = 1, \ldots, L,
\end{aligned}$$

where $r_0 = I$ handles insertion states.

3.5 Multimodality

The Gibbs sampler guarantees convergence to the target distribution. However, in practice the time to convergence may be unreasonably long. This can occur, for example, when the Gibbs sampler is stuck in one of several modes of the target distribution. The problem of monitoring convergence to a multimodal target distribution is addressed in the paper of Gelman and Rubin (1992). They give a profound discussion of the problem and suggest a general method to monitor convergence. However, the problem that arises in HMM restoration has two features that preclude direct application of this approach. First, the distribution is continuous in the parameter θ, and is discrete in the missing data component $\mathbf{S}$. Second, the posterior distribution, for MDI models in particular, can have a tremendous number of modes. Furthermore, it appears that once the Gibbs sampler finds a mode, it is often impossible in a practical sense for it leave. The source of the multimodality for MDI models is easily understood. For every prototype sequence $\mathbf{r}$ that differs form the "true" prototype sequence by a small number of insertions and/or deletions, there exist alignments that fit the data reasonably well. Once the sampler finds such an alignment, it will remain in a region of the alignment space corresponding to prototype sequences that differ from $\mathbf{r}$ only by substitutions. The total probability mass concentrated in this region is the probability that the true prototype sequence is in this set and may be rather small. In our experience with DNA data, we have found that only one or at most a few modes have any significant mass.

We are interested in identifying these massive modes and the corresponding set of prototype sequences. An ideal practical solution for the DNA problem (as detailed in the following paragraphs) would be to identify all of the massive modes, estimate their relative probabilities and find distributions of prototypes within those modes. Of course one cannot guarantee that all massive modes have been identified and it will be prudent to make many runs of the Gibbs sampler using different starting points.

First, note that the marginal posterior distribution on multiple alignments has support on a finite set. Furthermore the Gibbs sampler splits this set into disjoint subsets corresponding to modes of the distribution. It is helpful that the marginal distribution of alignments can be found explicitly up to a constant. Indeed, if the initial distribution is Dirichlet we can sum over all $\mathbf{r}$ and integrate out $(\tilde{\Lambda}, \tilde{\Pi})$ in (16) to obtain

$$\Pr(\mathbf{S} \mid \mathbf{Y}) \propto \prod_{s \in \{M,D,I\}} A(\mathbf{b}_s + \tilde{C}^{\Lambda}_{s\cdot}) A(\mathbf{a}_I + \tilde{C}^{\Pi}_{0\cdot}) \prod_{i=1}^{L} \sum_{r \in \{A,C,G,T\}} \alpha_r \frac{A(\mathbf{a}_r + \tilde{C}^{\Pi}_{i\cdot})}{A(\mathbf{a}_r)}, \quad (18)$$

where $A(\cdot)$ is a normalizing constant of the Dirichlet distribution. When the Gibbs sampler is stuck in a subset of alignments, the probability of this subset can be determined up to a constant by summing (18) over all alignments in the set. In this way, the relative mass of different modes can be determined. This approach can be also used to discriminate between two models with different number of main states. Indeed, in this case the posterior distribution is defined on two disjoint spaces but one can still use (18) to evaluate $\Pr(\mathbf{S} \mid \mathbf{Y})$ and then to compare modes across different models.

The Bayesian restoration procedure is computationally intensive. The primary computational burden being the storage of many realizations of the multiple alignments. In practice one is often interested only in the, say 100, most probable alignments. By using formula (18) one can identify and store the best alignments and their *relative* probabilities. The efficiency of this approach is discussed further

in the Example. However, when the total probability of a mode is of interest, the storage problem cannot be avoided.

When several distinct prototypes are sampled, it will be desirable to evaluate their probabilities. Rao-Blackwellized estimates are known to have smaller asymptotic variance than estimates obtained directly from relative frequencies (Casella and Robert, 1996a). For any prototype, one can obtain an estimated probability as

$$\widehat{\Pr(\mathbf{r})} = 1/N \sum_{i=1}^{N} \Pr\left(\mathbf{r} \mid \mathbf{S^i}\right) = 1/N \sum_{i=1}^{N} \frac{\Pr\left(\mathbf{r}, \mathbf{S^i}\right)}{\Pr\left(\mathbf{S^i}\right)}, \tag{19}$$

where the numerator and denominator in the fraction can be evaluated up to the same constant via (18).

4 Example

A collection of DNA sequences described by Seto *et al.* (1993) was assembled using the program CAP (Huang, 1992). A small segment of this assembly was chosen to illustrate the Bayesian restoration method. Table 1 shows six DNA sequences $(\mathbf{y}_1, \ldots, \mathbf{y}_6)$ that form the raw data for our analysis.

The posterior distribution on alignments proved to be particularly sensitive to the prior distribution on the output parameters of the M–states. This happens because the total number of outputs from each M–state is small (at most six) and because the alignments are sensitive to substitution rates. The overall rate of substitution was chosen to be 0.008 based on other data (Lazareva *et al.*, 1997) and the *weight* of the prior distribution was taken to be about six. Thus for a state associated with prototype letter A we set $\mathbf{a}_A = (6, .012, .012, .012, .012)$ and similarly for C, G and T states. The prior distribution on the letters of the prototype was taken to be uniform, $\alpha_i = 1/4$. The prior distribution for the output of an I-state was uniform $\mathbf{a}_I = (1, 1, 1, 1, 1)$. Finally, because the posterior was less sensititive to the prior distribution on the (tied) state transition parameters, a uniform prior was chosen.

We note here that under the proposed model, certain classes of alignments have exactly the same probabilities. In particular, the placement of insertions within a run of identical bases is arbitrary. To minimize storage, we save only one representative of each insertion equivalence class. The number of members of each class is recorded as the *multiplicity* in table 2. Hereafter, the term alignment refers to an equivalence class of alignments.

The first task in the analysis was to determine the number of M–states needed in the model. The CAP alignment suggested a prototype with 54 states. However when compared with a model based on prototypes of length 53, the most massive mode of the 54 state model appears to be 10^7 times less likely. Figure 5 shows the accumulation (over Monte Carlo iterations) of probability mass for the two largest modes in each of the 53 and 54 state models. The remainder of our analysis assumes a model with 53 M–states and is focused on the single dominant mode. Within this mode, a Monte Carlo run of 5000 steps suggested that the alignment shown in Figure 6a would make a good regeneration point (see Appendix B). In a subsequent run of 100,000 steps, 30790 distinct multiple alignments were explored resulting in 352 tours. Only two prototypes were sampled with substantial frequency. They are distinguished from one another by having either C or G in the 10-th position. We will refer to these as the C–prototype and the G–prototype, respectively. The Rao-Blackwellized estimate (19) of the C–prototype probability is 0.699. For comparison, the relative frequency estimate is 0.713 with an estimated standard error of 0.012. The confidence interval was calculated using regeneration as described in Appendix B. We conclude that the most probable prototype sequence has C in the 10-th position. We note that the C–prototype agrees with the sequence (positions 10125–10174) reported by Seto *et al.* (1993).

Posterior probabilites for the top 100 multiple alignments within the mode are summarized in Figure 7a. The alignments are ordered with respect to their relative probabilities obtained from (18). The figure shows these probabilities (scaled by an

appropriate constant) and their Monte Carlo estimates. The most frequent variants of the multiple alignment are summarized in Figure 6b and table 2 identifies which variants correspond to the top 100 alignments. After 10,000 steps the Gibbs sampler had identified all of the top 100 alignments. Thus the inference about the shape of posterior distribution of alignments does not require much time. On the other hand the Monte Carlo probability estimates appear to be biased even after 50,000 steps. The combination of Monte Carlo with analytic results was most effective in developing a clear picture of the posterior distribution on alignments.

Alignments	Multiplicity	Region				
		1	2	3	4	5
1-10	12	a	a, b,c,d,e	a	a	a,b
11-20	12	a	a, b,c,d,e	a	a	c,d
21-30	12	b	a, b,c,d,e	a	b	a,b
31-40	12	b	a, b,c,d,e	a	a	a,b
41-50	12	b	a, b,c,d,e	a	c	ab
51-60	60	c	a, b,c,d,e	a	a	a,b
61-70	60	c	a, b,c,d,e	a	a	c,d
71-80	12	b	a, b,c,d,e	a	a	c,d
81-100(120)	24	b	a, b,c,d,e	b, c	b	a, b

Table 2: Configuration of the alignments (the last group comprises 40 alignments).

The conditional probability of the C-prototype given an alignment is shown for the top 100 alignments in fig 7b. Theoretical probabilities were obtained according to (19). Monte Carlo estimates are also shown. It is interesting to note that, although the marginal (over alignments) posterior favors the C–prototype, the top 20 alignments all favor the G–prototype. The main point of our example is that failure to account for uncertainty in an alignment can lead to an incorrect inference.

In conclusion, this analysis demonstrates that Bayesian restoration methods can be used to assess the quality of DNA sequence alignments. Furthermore, the method can be used to make inferences that do not depend on choosing a single fixed alignment or a fixed set of error rate parameters. Perhaps surprisingly, we have demonstrated that inferences based on the conditional distribution of a prototype given the "best" alignment can be misleading.

5 Discussion

The example provided in section 4 deals with only a small segment of a much larger multiple sequence alignment. This was neccessary in part because the Bayesian restoration procedure is computationally intensive. The primary computational burden being the storage of many realization of the multiple alignment. We believe that with some creative bookkeeping, perhaps taking advantage of the fact that large blocks of alignments never move, larger problems could be tackled. We note, however, that there are many applications of MDI hidden Markov models where storage would not present such a significant problem. For example, protein sequence alignments (Krogh *et al.*, 1994; Baldi *et al.*, 1994) use MDI models with at most a few hundred main states. In typical DNA sequencing data, there will be a small number of DNA sequences that are all highly similar to one another. In protein sequence applications, it is more typical to have a large number of highly divergent sequences. A discussion of the protein analysis problems can be found in Krogh *et al.* (1994). Methods described here could be applied with some modifications to the protein alignment problem. In the DNA example, the assumed independence of the multiple realizations of the HMM is at least plausible. However in studies of naturally occurring sequences, evolutionary relationships will induce correlations among the sequences. Thus there are some challenging problems to be addressed.

We have tested the Bayesian restoration technique on other HMM architectures, including two–state recurrent and 3–state left–to–right models. We find that

multimodality of the posterior and consequent "sticking" of the Gibbs chain can occasionally present problems. Methods are available to improve the mixing behavior MCMC algorithms (*e.g.* Geyer and Thompson, 1994) and we are continuing to experiment with these methods.

The main advantage of the Bayesian approach is that it enables one to study the reliablity of the estimation of a complex discrete structure such as an HMM restoration. Our ability to summarize and visualize these distributions is limited, but with careful attention to particular examples, innovative and effective summaries of uncertainty can be developed. The algorithmic complexity of our approach is comparable to the Viterbi training (Merkav and Ephraim, 1991) but the Gibbs sampling approach has verifiable convergence properties. Furthermore, it allows for exploration of the full posterior distribution which can reveal interesting features that the Viterbi and maximum likelihood approaches to HMM restoration would miss.

REFERENCES

Allison L., Wallace C.S., (1994), "Posterior distribution of alignments and its application to parameter estimation of evolutionary trees and to optimization of multiple alignments", *J. Mol. Evol.*, 39, 418-430.

Baldi P., Chauvin, Y., Hunkapillar, T., McClure, M.A. (1994), "Hidden Markov models of biological primary sequence information," *Proceedings of the National Academy of Sciences USA*, 91, 1059-1063.

Baum, L.E. and Petrie, T. (1966), "Statistical inference for probabilistic of finite state Markov chains," *Annals of Mathematical Statistics*, 37, 1554-1563.

Baum, L.E., Petrie, T., Soules, G., Weiss, N., (1970), "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics*, 41, 164-171.

Casella, G., Robert, C.P. (1996), "Rao-Blackwellization of Sampling Schemes" , *Biometrika*, 83,81-94.

Churchill, G.A. (1989), "A stochastic model for heterogeneous DNA sequences," *Bulletin of Mathematical Biology*, 51, 79-94.

Churchill, G.A., (1992) "Hidden Markov chains and the analysis of genome structure," *Computers and Chemistry*, 16, 107-115.

Churchill G.A. (1995), "Accurate restoration of DNA sequences," *Case Studies in Bayesian Statistics* vol. II, eds. C. Gatsaris, J.S. Hodges, R.E. Kass, N.D. Singpurwalla, Springer-Verlag, New-York, 90-148.

Dempster, A.P., Laird, N.M., Rubin, D.B.(1977) "Maximum likelihood from incomplete data via EM algotithm", *J. Roy. Statist. Soc. B*, 39,1-38.

Eddy S.R (1995) "Multiple Alignment using hidden Markov Models", *Proc. of the Third International Conference on Intelligent Systems for Molecular Biology*, Menlo Park:AAAI Press, 114-120.

Eddy S.R.,(1996) " Hidden Markov Models", *Current Opinion in Structural Biology*, 6, 361-365.

Elliot R.J., Aggoun L., and Moore, J.B. (1994), *Hidden Markov Models: Estimation and Control*, Applications of Mathematics, Vol 29, Springer-Verlag, New-York.

Fredkin, D.R. and Rice, J.A. (1992a), "Maximum likelihood estimation and identification directly from single-channel recordings," *Proc. Roy. Soc. Lond. B*, 249, 125-132.

Fredkin, D.R. and Rice, J. (1992b), "Bayesian restoration of single channel patch clamp recordings," *Biometrics*, 48, 427-448.

Gelfand, A.E., Smith, A.F.M. (1990) "Sampling based approaches to calculating marginal densities". *Journal of the American Statistical Association*, 85, 398-409.

Gelman, A, Rubin D.B. (1992) "Inference from iterative simulation using multiple sequences", *Statistical Science*, 7(4), 457-511.

Geman, S., Geman, D., (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741.

Geyer, C.J., Thompson, E.A. (1995), "Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference," *Journal of the American Statistical Association*, 90, 909-920.

Huang, X. (1992) "A contig assembly program based on sensitive detection of fragment overlaps" *Genomics*, 14, 18-25.

Hunkapillar, T, Kaiser, R.J., Koop, B.F., Hood, L. (1991), "Large–scale automated DNA sequence determination," *Science* 254, 59-67.

Juang, B.H., and Rabiner, L.R. (1990), "The Segmental k-means Algorithm for Estimating Parameters of Hidden Markov Models", *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38, 1639-1641.

Juang, B.H., and Rabiner, L.R., (1991), "Hidden Markov models for speech recognition" *Technometrics*, 33: 251-272.

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

Lawrence, C.E., Altschul, S.F., Boguski, M.S., Lui, J.S., Neuwald, A.F., Wootton, J.C. (1993), "Detecting subtle sequence signals: A Gibss sampling strategy for multiple alignment", *Science* 262, 208-214.

Lazareva, B., Churchill, G.A., Casella G. (1997) "Sampling based methods for the estimation of DNA sequence accuracy", *in preparation.*

Leroux, B.G., (1992) "Maximum–likelihood estimation for hidden Markov models," *Stochastic Processes and their Applications*, 40, 127-143.

Liu, J.S., Lawrence, C.E. (1995a), "Statistical Models for Multiple Sequence Alignment: Unifications and Generalizations", *ASA Proceeding of Statistical Computing*, 1-8.

Liu, J.S., Neuwald, A.F., Lawrence, C.E. (1995b), "Bayesian models for local multiple sequence alignment and Gibbs sampling strategies", *Journal of the American Statistical Association*, 90(432), 1156-1170.

Merkav N. and Ephraim Y. (1991) "Maximum Likelihood Hidden Markov Modeling Using a Dominant Sequence of States", *IEEE Transactions on Signal Processing*, 39, 2111-2115.

Rabiner, L.R. (1989), "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, 77, 257-286.

Ripley, B.D. (1987) *Stochastic Simulation*, Wiley, New-York.

Robert, C.P., Celeux, G., Diebolt J. (1994) "Bayesian estimation of hidden Markov chains: A stochastic implementation", *Statistics and Probability Letters*, North-Holland, 16, 77-83.

Robert, C.P. (1994) *The Bayesian Choice*, Springer–Verlag, New York, NY.

Seto, D., Koop, B.F., Hood, L. (1993) "An experimentally derived data set constructed for testing large–scale DNA sequence assembly algorithms," *Genomics*, 15, 673-676.

Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Main, I.S., Haussler, D. (1996) "Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology", CABIOS 12: 327-345.

Stratanovich, R.L (1960) "Conditional Markov Processes" *Theory of Probability and its Applications*, 5,156-178.

Tanner, M., Wong W. (1987) "The calculation of posterior distribution by data augmentation", *JASA*, 82, 528-550.

Viterbi, J., (1967) "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions of Information Theory*, 13, 260-269.

Waterman, M.S. (1995) *Introduction to Computational Biology*, Chapman and Hall, London, UK.

White J.V., Stultz, C.M., Smith, T.F. (1994), "Protein classification by stochastic modeling and optimal filtering of amino–acid sequences", *Mathematical Biosciences*, 119, 35-75.

Appendix A: Proof of Theorem 3.1

First, we observe that

$$\Pr\left(B, \mathbf{s}_{[1,n]}, E \,|\, \mathbf{y}, \theta\right) = \Pr\left(s_n \,|\, \mathbf{y}, \theta\right) \cdot \Pr\left(s_{n-1} \,|\, s_n, \mathbf{y}, \theta\right) \cdots \Pr\left(s_1 \,|\, \mathbf{s}_{[2,n]}, \mathbf{y}, \theta\right),$$

where $\mathbf{y} = y_1, y_2, ..., y_n$. The proof follows from the observation that the conditional distribution $\Pr\left(s_{t-1} \,|\, \mathbf{s}_{[t,n]}, \mathbf{y}, \theta\right)$ depends only on s_t, $\mathbf{y}_{[1,t-1]}$ and θ. Looking at a general term in the expansion,

$$\begin{aligned}
\Pr\left(s_{t-1} = i \,|\, \mathbf{s}_{[t,n]}, \mathbf{y}, \theta\right) &\propto \Pr\left(s_{t-1} = i, s_{[t,n]}, \mathbf{y} \,|\, \theta\right) \\
&= \Pr\left(s_{t-1} = i, \mathbf{y}_{[1,t-1]} \,|\, s_t, \theta\right) \Pr\left(\mathbf{s}_{[t+1,n]}, \mathbf{y}_{[t,n]} \,|\, s_t, \theta\right) \Pr\left(s_t \,|\, \theta\right) \\
&\propto \Pr\left(s_{t-1} = i, s_t \,|\, \mathbf{y}_{[1,t-1]}, \theta\right) \\
&= \Pr\left(s_{t-1} = i \,|\, \mathbf{y}_{[1,t-1]}, \theta\right) \cdot \Pr\left(s_t \,|\, s_{t-1} = i\right) \\
&= f_i(t-1)\lambda_{is_t},
\end{aligned}$$

where the first equality holds because of conditional independence of $\mathbf{y}_{[1,t-1]}$ and $\mathbf{y}_{[t,n]}$ given s_t. Thus $\Pr\left(s_{t-1} \,|\, \mathbf{s}_{[t,n]}, \mathbf{y}, \theta\right)$ is a multinomial distribution with probabilities proportional to $f_i(t-1)\lambda_{is_t}$ and the "backward" sampling scheme follows from this.

It remains to notice that due to (10)

$$\frac{\mathbf{f}(t-1) \star \Lambda_{\cdot s_t}}{|\mathbf{f}(t-1) \star \Lambda_{\cdot s_t}|} = \frac{\mathbf{f}^*(t-1) \star \Lambda_{\cdot s_t}}{|\mathbf{f}^*(t-1) \star \Lambda_{\cdot s_t}|},$$

which completes the proof.

Appendix B: Monte Carlo Error

To assess the asymptotic variance of an estimator $\widehat{f}_M = \frac{1}{M}\sum_{i=1}^{M} f(\mathbf{S}^i, \theta^i)$ one can use the regenerative property of the chain (S^m, θ^m). (Ripley, 1987, Geyer and Thompson, 1994)). Consider the chain $\{\mathbf{S}^m;\ m = 1, 2, ..\}$. Choose one of its states $\mathbf{R}$, and define the sequence $(t_0, t_1, \ldots)$ such that t_i is the time of the $(i+1)$-th visit to the state $\mathbf{R}$. In practice, the state $\mathbf{R}$ should be chosen after some preliminary investigation to be one of the most frequently visited states. The Markov property of the chain implies that the interarrival times $\{\tau_k \equiv t_k - t_{k-1}\}$ for $k = 1, 2, \ldots$ form an *i.i.d.* sequence with $\mathrm{E}\tau_k < \infty$. Moreover, due to Gibbs sampler the *tours*

$$\left\{(\mathbf{S}^{t_{k-1}} \equiv \mathbf{R}, \theta^{t_{k-1}}), (\mathbf{S}^{t_{k-1}+1}, \theta^{t_{k-1}+1}), ..., (\mathbf{S}^{t_k-1}, \theta^{t_k-1})\right\},$$

and hence the random variables

$$F_k \equiv \sum_{i=t_{k-1}}^{t_k-1} f(\mathbf{S}^i, \theta^i)$$

are *i.i.d.* . It follows that the sample mean converges to the desired expectation,

$$\widehat{f(\mathbf{S}, \theta)} = \frac{(1/K)(F_1 + F_2 + ... + F_K)}{(1/K)(\tau_1 + \tau_2 + ... + \tau_K)} \xrightarrow{\text{a.s.}} \frac{\mathrm{E}F_1}{\mathrm{E}\tau_1} = \mathrm{E}f. \tag{20}$$

Finally, we introduce the centered random variables $F_k^* = F_k - \tau_k \mathrm{E}f$. When both $var(F_1)$ and $var(\tau_1)$ are finite, $var(F_1^*) < \infty$. It then follows, from the central limit theorem, that

$$\sqrt{K}(\widehat{f(X)} - \mathrm{E}f) = \sqrt{K}\frac{(F_1^* + F_2^* + ... + F_K^*)}{(\tau_1 + \tau_2 + ... + \tau_K)} \xrightarrow{D} N\left(0, \frac{var(F_1^*)}{(\mathrm{E}\tau_1)^2}\right). \tag{21}$$

The asymptotic variance of $\widehat{f(X)}$ can estimated by substituting the estimates

$$\mathrm{E}\tau_1 \approx \widehat{\tau},$$

and

$$var(F_1^*) \approx \widehat{F^2} - 2\widehat{F\tau}\widehat{F}/\widehat{\tau} + \widehat{F}^2\widehat{\tau^2}/\widehat{\tau}^2$$

into the right-hand side of (21), where $\widehat{\ }$ denotes the sample mean over $k = 1, 2, .., K$. We note that the above estimates can be updated as the Gibbs chain progresses.

Figure Captions

Figure 1 Two–state recurrent HMM architecture.

Figure 2 Left–to–right architecture for locating a pattern of size k embedded in a longer sequence.

Figure 3 Mutation–deletion–insertion (MDI) architecture with three M–states.

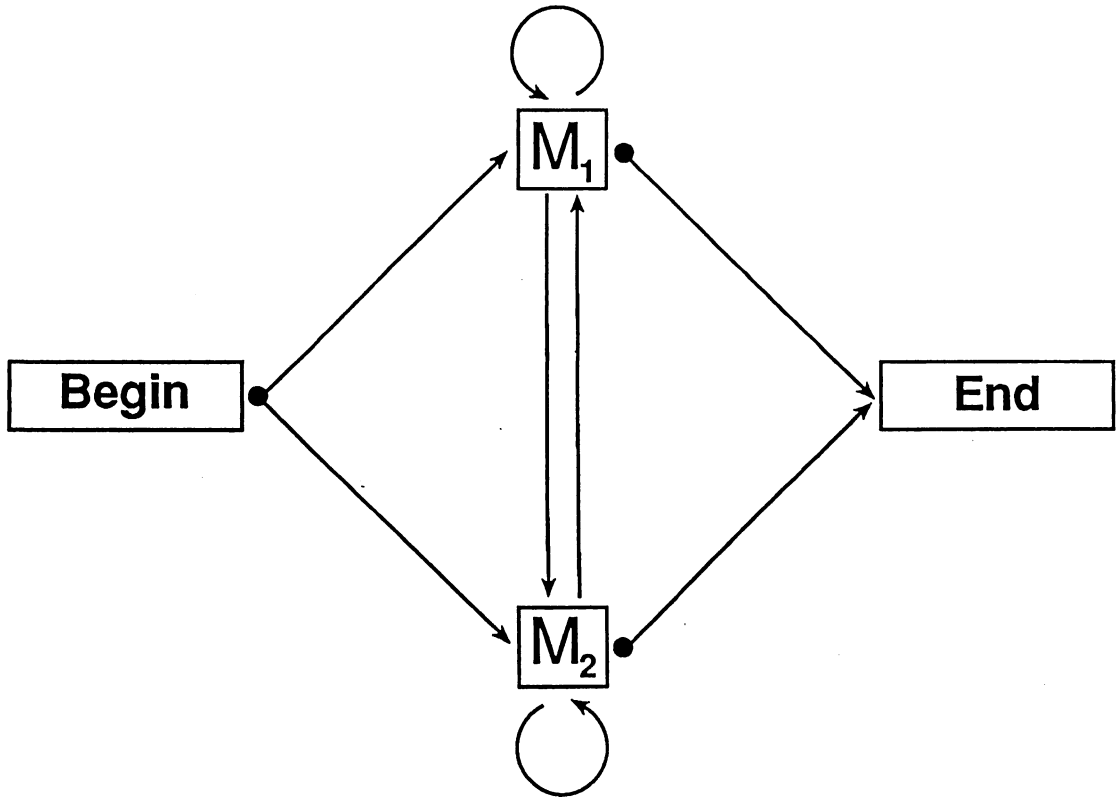
Figure 4 Pathgraph representation of all paths through an MDI model with three M–states that could have produced an output sequence with four elements. Diagonal transition correspond to M–states, horizontal transitions correspond to D–states and vertical transitions correspond to I–states.

Figure 5 The accumulation of probability mass for the two largest modes in each of the 53 state (thick lines) and 54 state (thin lines) models is shown as a function of Monte Carlo iterations on a log–log scale.

Figure 6 A multiple sequence alignment (6a) of the 6 six DNA sequences from table 1. Letters shown below the sequences are insertions and "–" show locations of deletions. This basic alignment served as the regeneration point for the Monte Carlo analysis. Variants of the multiple alignment are shown below the alignment (6b) and are referred to in table 2.

Figure 7 The unconditional posterior probability of alignments is shown for the top 100 alignments (7a). Analytic and Monte Carlo estimates are shown. Bars indicate 96% confidence intervals for the Monte Carlo estimates (see Appendix B). Analytically derived relative probabilities were normalized to have the same total mass as the 100 Monte Carlo estimated probabilities. The conditional probabilities

of the C–prototype given an alignment is also shown for the top 100 alignments (7b).


Begin
M1
M2
End

Begin
I_0
M_1
M_k
I_k
End

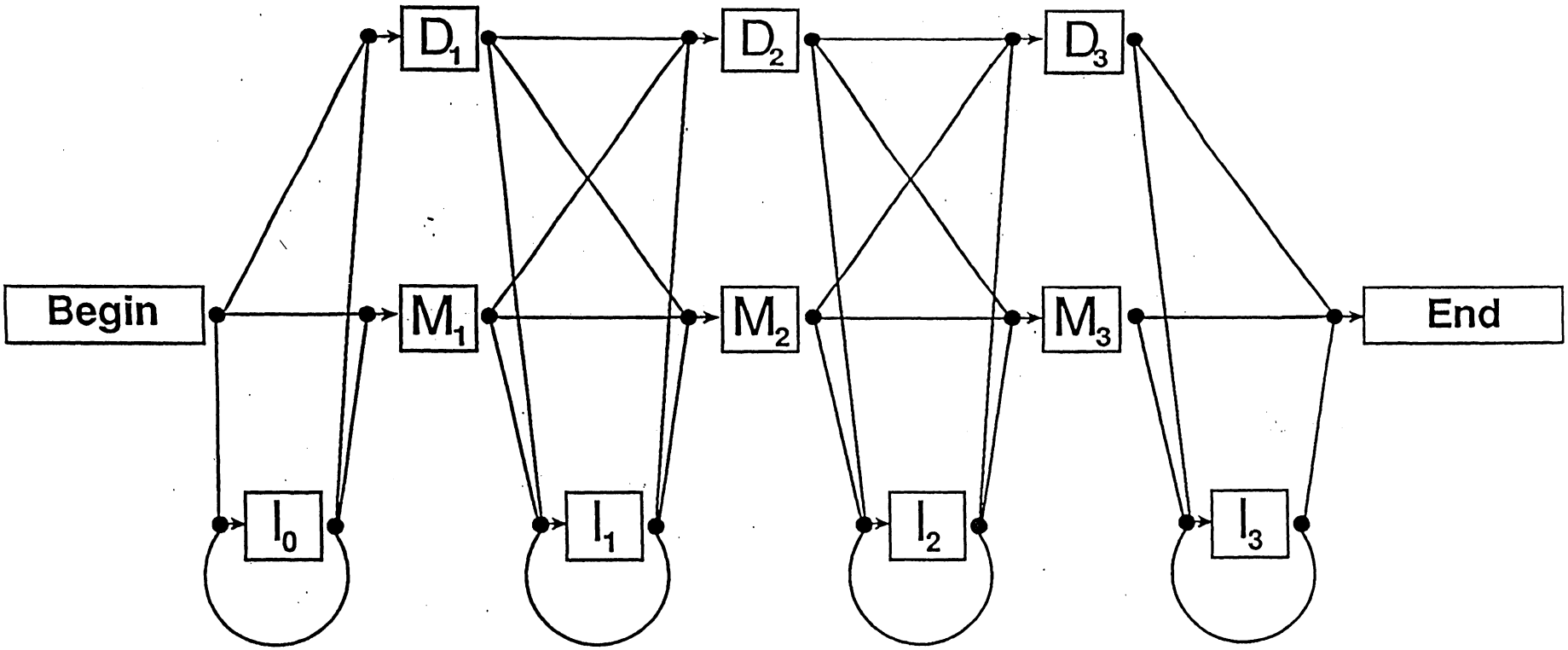


Fig 3

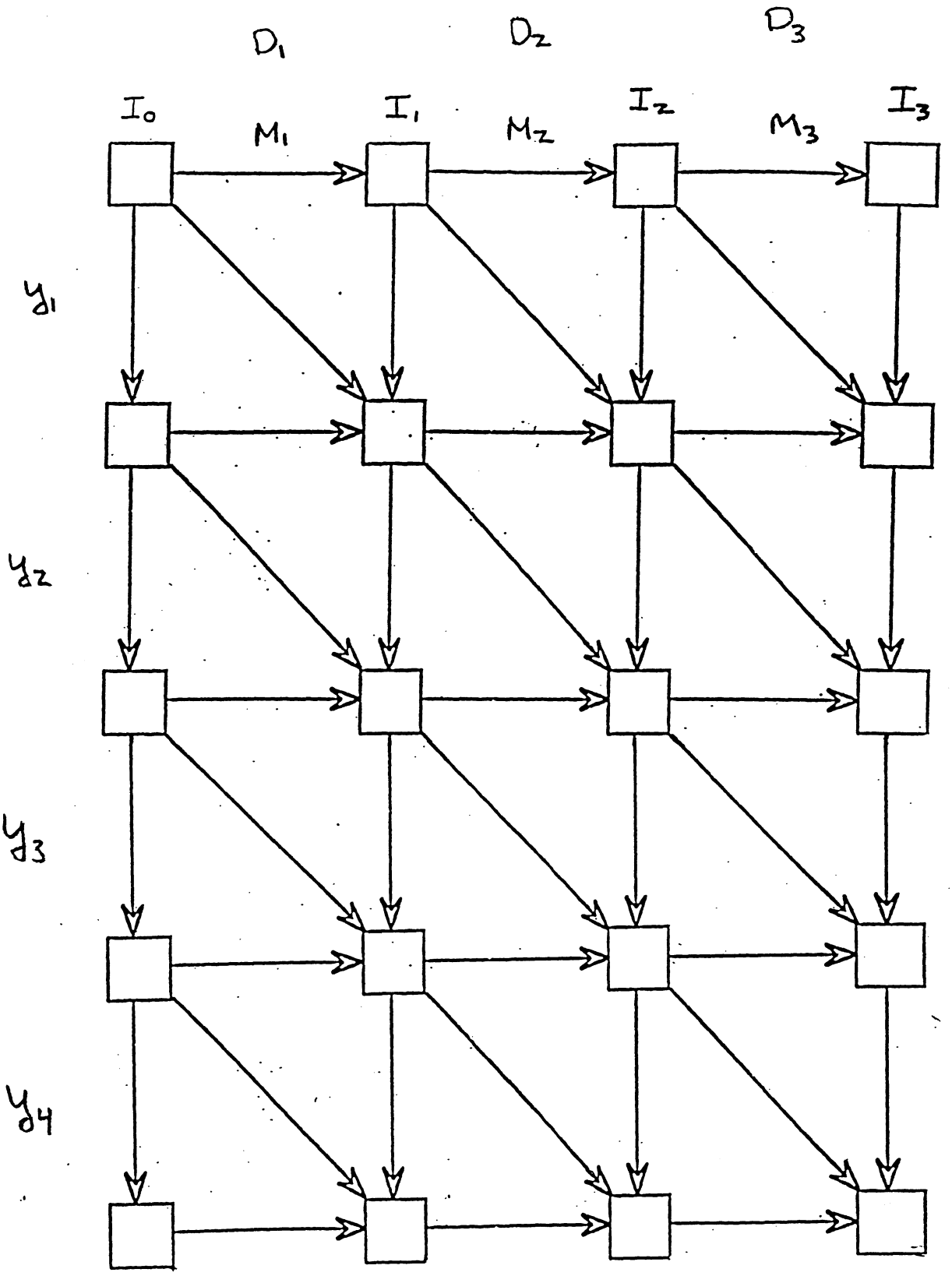
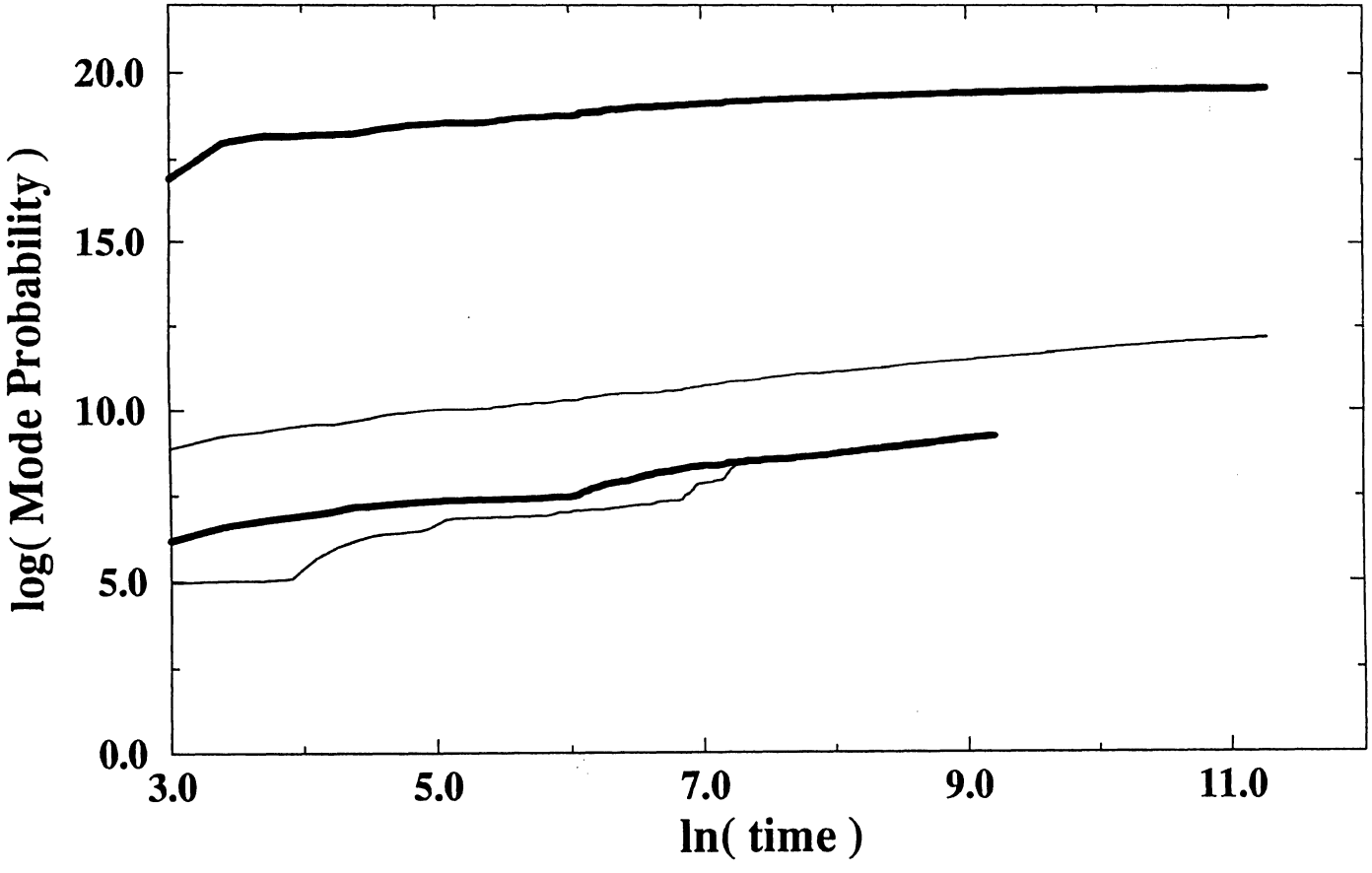
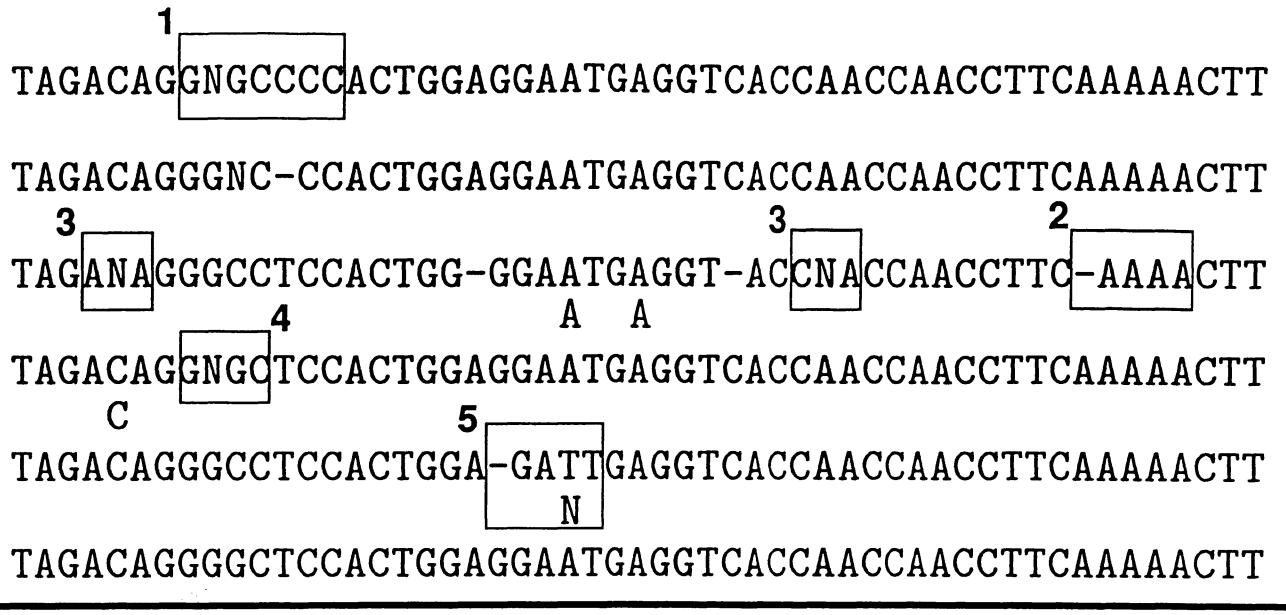


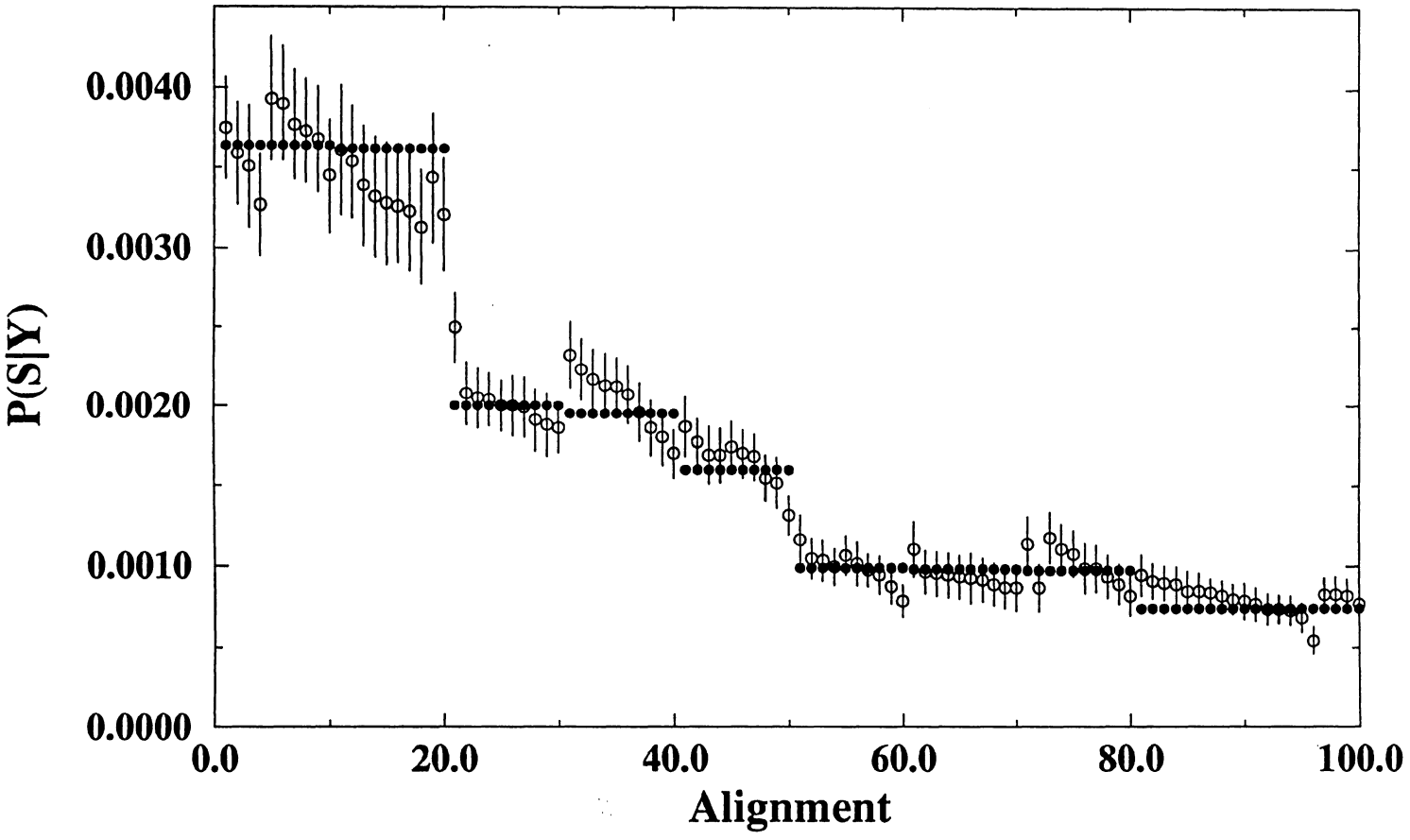
Fig 4

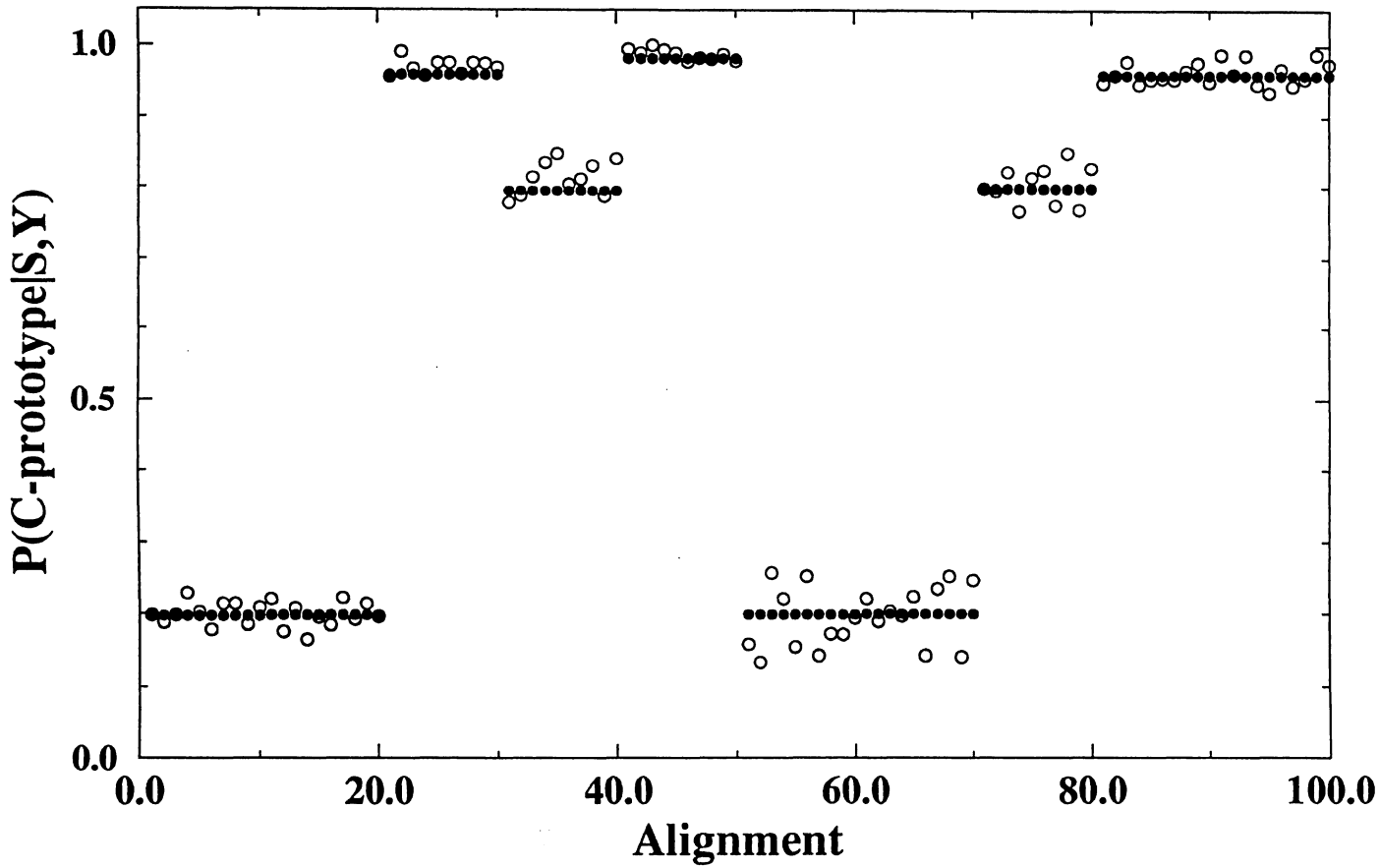

20.0
15.0
10.0
5.0
0.0
log(Mode Probability)
3.0
5.0
7.0
9.0
11.0
ln(time)

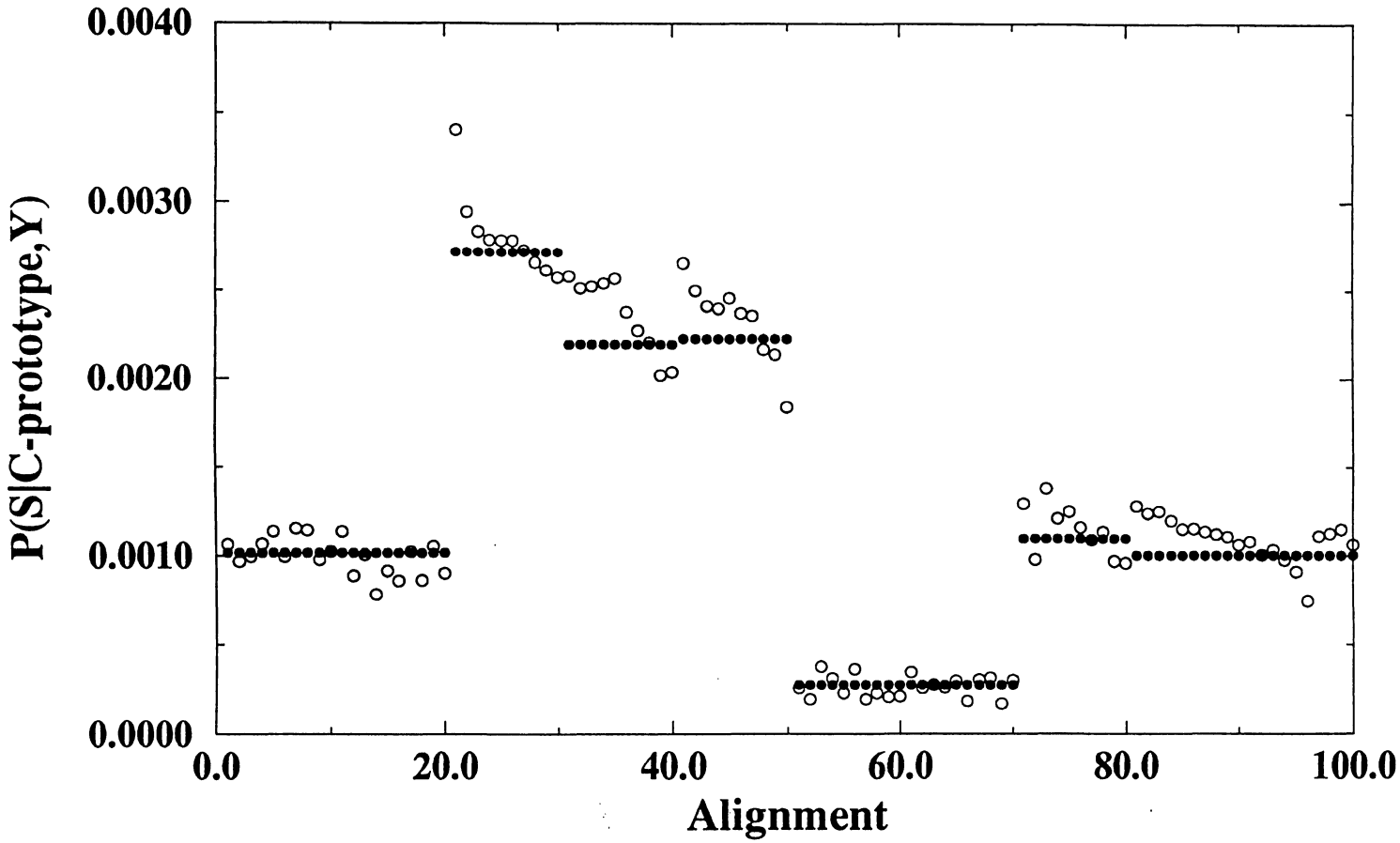


TAGACAGGGGCTCCACTGGAGGAATGAGGTCACCAACCAACCTTCAAAAACTT

	1	2	3	4	5
a	GNGCCCC	-AAAA	ANA...CNA	GNGC	-GATT N
b	GGCC-CC N	A-AAA	ANA...C-A N	GG-C N	G-ATT N
c	GNGC-CC C	AA-AA	A-A...CNA N	GGC- N	-GANT T
d		AAA-A			G-ANT T
e		AAAA-			


0.0040
0.0030
0.0020
0.0010
0.0000
P(S|Y)
0.0
20.0
40.0
60.0
80.0
100.0
Alignment


1.0
0.5
0.0
P(C-prototype|S,Y)
0.0
20.0
40.0
60.0
80.0
100.0
Alignment


0.0040
0.0030
0.0020
0.0010
0.0000
P(S|C-prototype,Y)
0.0
20.0
40.0
60.0
80.0
100.0
Alignment