JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 9, Number 2, 2002

Pp. 389–399

Applications of Generalized Pair Hidden Markov Models to Alignment and Gene Finding Problems

LIOR PACHTER,[1] MARINA ALEXANDERSSON,[2] and SIMON CAWLEY[2]

ABSTRACT

Hidden Markov models (HMMs) have been successfully applied to a variety of problems in molecular biology, ranging from alignment problems to gene finding and annotation. Alignment problems can be solved with pair HMMs, while gene finding programs rely on generalized HMMs in order to model exon lengths. In this paper, we introduce the generalized pair HMM (GPHMM), which is an extension of both pair and generalized HMMs. We show how GPHMMs, in conjunction with approximate alignments, can be used for cross-species gene finding and describe applications to DNA–cDNA and DNA–protein alignment. GPHMMs provide a unifying and probabilistically sound theory for modeling these problems.

Key words: hidden Markov model, alignment, gene finding, comparative genomics.

1. INTRODUCTION

THE DISTINCT PROBLEMS OF *alignment* and *gene finding* have been, and continue to be, the impetus of much fertile research in computational biology. The theoretical framework of Needleman and Wunsch (1970) was a landmark in the application of mathematics to sequence analysis. In their paper, they showed that the dynamic programming method (DP) could be applied to the problem of finding similarity between sequences. Dynamic programming has since flourished in the context of biology and has been applied to a variety of problems with continued success. The problem of annotating translational and transcriptional features in genomes, the gene finding problem, has been one of the key beneficiaries of dynamic programming. The technique lies at the heart of all the successful gene finding programs to date (for a discussion of DP applications to gene finding see Salzburg [1998]; a partial list of users of DP is Burge and Karlin [1997], Henderson *et al.* [1997], Kulp *et al.* [1996], Pachter *et al.* [1999], and Wirth [1988]).

Despite the dynamic programming connection between alignment and gene finding, the two problems have traditionally been treated separately by computational scientists, and most programs are developed with only one of the problems in mind. Current methods for incorporating DNA and protein alignments into homology-based gene finders (Krogh, 2000; Kulp *et al.*, 1996; Pachter *et al.*, 1999) treat the alignments as separate evidence to be used for reweighting candidate gene annotations, rather than tackling the two problems jointly. On the other hand, the gene-finding process is generally not used in finding sequence

[1]Department of Mathematics, University of California Berkeley, Berkeley, CA.
[2]Department of Statistics, University of California Berkeley, Berkeley, CA.

alignments (see Gotoh [2000] for an exception). This is in sharp contrast to biologists who have always relied on comparisons for annotating biologically important features in genomes and for finding functionally significant domains. Indeed, the principle of comparison has been a leitmotif in biology.

A step in the direction of simultaneous alignment and gene finding was undertaken in the Procrustes project, initiated by Pevzner, Mironov, and Gelfand (1996). The Procrustes program was based on the observation that boundary detection for exons in a gene could be significantly enhanced if a protein homolog for the gene existed. Their method was to find the best alignment of the protein homolog to the DNA sequence subject to splice site constraints. Such constraints had previously played a role only in gene finding programs, not in alignment programs. Indeed, they coined the term *spliced alignment* for the problem they were solving. Recent approaches to gene finding are based on using a pair of orthologous DNA sequences from two organisms to simultaneously annotate both (Bofna and Huson, 2000; Batzoglou *et al.*, 2000; Korf *et al.*, 2001; Wiehe *et al.*, 1999). Despite the fact that the annotation of orthologous genes (and their correspondence) elucidates some of the alignment, in these approaches, alignments of the regions are performed first, and then they are used to enhance gene detection.

In this paper, we describe a unifying framework for alignment and gene finding. Our approach is to find the best alignment between two sequences while simultaneously annotating the regions. Thus, the identification of transcriptional and translational features aids in the alignment, and the alignment helps to find the genes. Our methods are based on hidden Markov models, which first found their way into computational biology in the context of sequence analysis (Churchill, 1989). One of the benefits of using hidden Markov models is that they are probabilistic, and so their output has a probabilistic interpretation. For example, using the GENSCAN program (Burge, 1997; Burge and Karlin, 1997), one obtains not just annotations but also probabilities of the predicted exons being correct, given the input sequence. Hidden Markov models are also useful for alignments (Searls and Murphy, 1995). In fact, the dynamic programming algorithm discovered by Needleman and Wunsch (1970) and extended/improved by Smith and Waterman (1981) is equivalent to the Viterbi algorithm for an appropriate hidden Markov model (Durbin *et al.*, 1998; Holmes, 1998). The first inkling of a marriage between alignment and gene finding HMMs appeared recently in the work of Kent and Zahler (2000).

The HMM we develop is both a generalized HMM (the type used for gene finding) and a pair HMM (the type used for alignment). We call such an HMM a generalized pair hidden Markov model (GPHMM). These HMMs are discussed in more detail in Section 2. In Section 3, we describe some applications of our GPHMM to a variety of annotation and alignment problems:

- DNA–DNA alignments. Here our GPHMM serves to perform both the task of alignment and gene finding simultaneously.
- DNA–cDNA alignment. This is a special case of the model used for DNA–DNA alignment. The advantage of using a GPHMM is that it is possible to obtain gene predictions that take into account exon lengths, splice site signals, and other features not incorporated in current programs.
- DNA–protein alignment. This is the spliced alignment problem described above. Again, our program has the advantage that the gene predictions are enhanced by taking into account not just splice sites but also exon lengths, transcriptional signals, and other biologically important information.

2. GENERALIZED PAIR HIDDEN MARKOV MODELS

Hidden Markov models (HMMs), the main reference often being in the context of speech recognition (Rabiner, 1989), have become increasingly popular for biological applications. The probabilistic framework underlying HMMs enables the seamless integration of various biological features into a unifying model in a way that is both flexible and mathematically rigorous. There are two different kinds of HMMs relevant to our problem: pair HMMs and generalized HMMs. Pair HMMs are well-suited for modeling alignment problems. Generalized HMMs are useful for gene finding. Further details and explanations of HMMs, these two particular types, and our extension of them are provided in this section.

2.1. Generalized hidden Markov models (GHMMs)

The idea of an HMM is that there is an underlying Markov chain which generates a sequence of states. In the gene-finding problem, the states typically include exons, introns, and intergenic regions. The Markovian

assumption implies that at any point the next state generated in the sequence will depend only upon the last. The term "hidden" refers to the fact that the sequence of states is not observed; instead, the observables are outputs from the states. In each state, the output will typically depend on the current state and may also depend on previous outputs. Referring again to the gene-finding problem, the output at each state is one of the four bases A, C, G, or T. The probability of each base will depend on the type of state.

A *generalized* HMM (Rabiner, 1989), also known as a hidden semi-Markov model, generalizes HMMs in that each hidden state may generate more than a single base. In a standard HMM, there is an output at every step in the underlying (hidden) Markov chain, even if that step leaves the chain in the same (hidden) state. Before moving to a new state, the chain will in general do a number of self-transitions, but after a random number of steps called the *duration time* in that state, it will eventually move to a different state. As a result, the duration time in any state turns out to have an exponentially decaying (geometric) distribution, and this may not be appropriate in all applications.

In a GHMM there is a duration distribution associated with each state, and a state generates output by first choosing the length according to this distribution, and then producing an output sequence of that duration. The generalization can improve performance by allowing for more accurate modeling of the typical duration of each hidden state, in particular in situations where the distribution of the length is important and significantly different from the exponential distribution.

2.2. Pair hidden Markov models (PHMMs)

One of the main problems in biological sequence analysis is to determine if two sequences, or parts of them, are functionally related. This is usually done by first aligning the sequences and then deciding whether the similarity in the alignment has occurred because the sequences are related, or just by chance. For this, a suitable scoring system is necessary to rank the alignments, and an algorithm is needed to find the optimal scoring alignment.

There is a probabilistic approach to this problem that makes use of the notion and algorithms of HMMs (Durbin *et al.*, 1988). One advantage of such an approach is that by appropriately weighting all alternative (suboptimal) alignments one can assign a similarity score to two sequences which is independent of any specific alignment. However, instead of generating a single sequence, as in a standard HMM or GHMM, we now generate an aligned pair of sequences, and we call this type of model a pair HMM (PHMM). A PHMM starts with an initial distribution and then cycles over the following two steps: 1) given the current state, pick the next state according to the transition probabilities; 2) pick a symbol pair in the new state according to an output distribution and add it to the alignment. Because we have probabilities for each step, we can also derive the probability of generating any particular alignment by taking the product of the probabilities at each step. One problem in sequence alignment is that when similarity is weak, it is hard to identify the correct alignment. In PHMMs we can calculate the probability that a given pair of sequences are related, independent of a specific alignment. This is done by summing over all possible alignments using a procedure called the forward algorithm, which is described below.

2.3. Generalized pair hidden Markov models (GPHMMs)

In this section, we present a model that merges the GHMM and PHMM in a framework we call the generalized pair HMM. The motivation behind the development of our model is to be better able to analyze biological sequences in pairs, a problem of increasing importance with the growing abundance of sequence data from different organisms. Our presentation of the theory is general, in that it can be applied to problems other than those of interest in biological sequence analysis.

Assume for a moment that our GPHMM is a sequence machine, generating output symbols as it cycles through a state space, corresponding to the physical output of the system being modeled. The GPHMM inherits the main features of the GHMM and the PHMM, generating generalized lengths of sequences in tandem.

Let $S = \{s_1, \ldots, s_N\}$ denote the state space, and $X_1, \ldots, X_L$ the sequence of hidden states that the GPHMM follows as it generates the output. Examples of various state spaces will be presented in Section 3. The first state $X_1 = s_i$ is distributed according to some initial distribution $(\pi_i)_{i=1}^N$, and a jump to a new state s_j occurs according to a transition probability distribution a_{ij}. With each hidden state X_l, we associate a pair of duration times (d_l, e_l), generated from some joint distribution, representing the number of symbols in each sequence that are generated from that state. Let $p_l = \sum_{k=0}^{l} d_k$ and $q_l = \sum_{k=0}^{l} e_k$

denote the partial sums of the durations. In state X_l, the GPHMM generates the sequence pair $Y_{p_{l-1}+1}^{p_l}$ and $Z_{q_{l-1}+1}^{q_l}$, respectively, according to a joint distribution $b_{X_l}(Y_{p_{l-1}+1}^{p_l}, Z_{q_{l-1}+1}^{q_l} | Y_1^{p_{l-1}}, Z_1^{q_{l-1}})$ (we are using the notation Y_a^b to represent the subsequence of DNA $Y_a, \ldots, Y_b$). Note that the observations depend only on the current state, the current durations, and the previously observed data. Though it is mathematically valid to condition on all previous observations, it rarely makes sense to condition on more than a handful.

In practice, we observe only the sequences Y_1^T and Z_1^U, where T, U are the sequence lengths. The variables L, X_1^L, d_1^L, and e_1^L are hidden from us. We assume that we have all of the observations in both sequences by the time we reach the final state X_L, or in other words that $p_L = T$ and $q_L = U$. We can now write down the probability of a particular sequence of hidden and observed data:

$$\Pr(X_1^L, Y_1^T, Z_1^U, d_1^L, e_1^L) = \pi_{X_1} f_{X_1}(d_1, e_1) b_{X_1}(Y_1^{p_1}, Z_1^{q_1}) \prod_{l=2}^{L} a_{X_{l-1}, X_l} f_{X_l}(d_l, e_l)$$

$$* b_{X_l}(Y_{p_{l-1}+1}^{p_l}, Z_{q_{l-1}+1}^{q_l} | Y_1^{p_{l-1}}, Z_1^{q_{l-1}}).$$

The two main problems to be solved when applying the HMM theory are: 1) to compute the probability of the observed data $\Pr(Y_1^T, Z_1^U)$ and 2) given the observed sequence, to find the underlying hidden sequence $X_1, \ldots, X_L$ of states that best "explains" the observations. The first problem can be solved by a procedure called the *forward algorithm* and involves the recursive computation of the forward variables, defined as

$$\alpha(t, u, i) \equiv \Pr\Big(Y_1^t, Z_1^u, \{\text{some hidden state ends in } s_i \text{ at } (t, u)\}\Big)$$

$$= \Pr\Big(Y_1^t, Z_1^u, \cup_{l=1}^{t+u}\big(X_l = s_i, p_l = t, q_l = u\big)\Big).$$

State s_i can be reached at time (t, u) from the N possible states s_j, $1 \le j \le N$. The joint probability of the observed sequences ending in state s_j at time $(t - d, u - e)$, where d and e are the durations of the previous state in each sequence, respectively, is again a forward variable. By letting D be the maximum possible duration in any state, we can sum over all previous states and their durations and get

$$\alpha(t, u, i) = \sum_{j=1}^{N} \sum_{d,e=1}^{D} \Pr\Big\{Y_1^t, Z_1^u, \cup_{l=1}^{t+u}\big(X_l = s_i, p_l = t, q_l = u\big), \cup_{l=1}^{t+u}\big(X_l = s_j, p_l = t - d, q_l = u - e\big)\Big\}$$

$$= \sum_{j,d,e} \alpha(t - d, u - e, j)\ a_{j,i}\ f_{s_i}(d, e) * b_{s_i}(Y_{t-d+1}^t, Z_{u-e+1}^u | Y_1^{t-d}, Z_1^{u-e}).$$

(A more formal derivation of the above may be found in Cawley (2000)).

The solution to problem (1), the probability of the observed data, follows by definition, since

$$\alpha(T, U, i) = \Pr(Y_1^T, Z_1^U, X_L = s_i)$$

and hence $P(Y_1^T, Z_1^U)$ is just the sum of the $\alpha(T, U, i)$'s.

A convenient feature of HMMs is the ability to compute posterior probabilities of hidden states given the observations. This is made possible by using the forward variables along with closely related quantities known as the backward variables. The recursion for the backward variables is similar to that for the forward, the main difference being that they are computed by recursing in the reverse direction over the data.

The solution of the second problem, that of finding the "optimal" state sequence giving rise to the observed data, depends on the definition of "optimal." A natural estimate is the mode of the distribution of the hidden data conditional on the observed, or in other words, the single most likely sequence of states and

durations through the model. Maximizing this probability $\Pr(X_1^L, d_1^L, e_1^L | Y_1^T, Z_1^U)$ over $\{L, X_1^L, d_1^L, e_1^L\}$, which has its maximum at the same point as $\Pr(X_1^L, d_1^L, e_1^L, Y_1^T, Z_1^U)$, can be done by an efficient dynamic programming routine, most commonly known as the *Viterbi algorithm.* The procedure, based on the variables

$$\delta(t, u, i) = \max_{l, X_1^{l-1}, d_1^{l-1}, e_1^{l-1}} \Pr(Y_1^t, Z_1^u, X_1^{l-1}, X_l = s_i, p_l = t, q_l = u)$$

is essentially the same as in the forward algorithm with sums replaced by maxima. To actually retrieve the best underlying state sequence, we need to keep track of the argument that maximized $\delta(t, u, i)$; that is, we backtrack through $\operatorname{argmax}_{s_j, d, e} \delta(t - d, u - e, s_j)$, starting in $\max_{s_i} \delta(T, U, i)$.

2.4. Approximate alignments

A naive implementation of the GPHMM described above takes $O(TUN)$ in memory storage and $O(D^4 N^2 TU)$ time to run, where D is the maximum duration in a hidden state, N the number of hidden states, and T and U the lengths of the sequences being analyzed. This means that the running time scales as some constant factor times the product of the lengths of the input sequences. One way of getting around this is to preprocess the data, producing an approximate alignment such that the problem grows linearly in the length of the observation sequences.

Under the assumption that there exists a "true" alignment of the sequences (but one which we may not be able to find), an approximate alignment allows us to state bounds on possible matches. Approximate alignments are necessary in the GPHMM framework described above because they allow for a reduction in memory (and computational) requirements, rendering large sequences on the order of hundreds of thousands of base pairs feasible.

An alignment of two sequences of lengths T and U, respectively, can be thought of as a path in $\mathbb{Z} \times \mathbb{Z}$ restricted to lie in the subset $R_{T,U} = [1, \ldots, T] \times [1, \ldots, U]$. The path must consist of steps from the step set $\{(0, 1), (1, 0), (1, 1)\}$. A path with a diagonal step entering a vertex (i, j) is an alignment in which position i in the first sequence is aligned with position j in the second; the vertical and horizontal steps correspond to gaps. A *global alignment path* is a path in $R_{T,U}$ which begins at a vertex $(r, 1)$ or $(1, r)$ for some r and ends at a vertex (T, s) or (s, U) for some s. A *local alignment path* is a contiguous subsequence of a global alignment path. A *global/local alignment set* refers to the set of points in $R_{T,U}$ appearing in an alignment path.

Definition. *An approximate alignment $\mathcal{A}$ is a nonempty subset $\mathcal{A} \subseteq R_{T,U}$ which is a union of global alignment sets.*

A preprocessing step that finds an approximate alignment between the two sequences allows us to significantly reduce the computation time. Without loss of generality, let Y_1^T be the longer of the two observation sequences. An approximate alignment that localizes each base of Y to a window of size h in Z allows us to set $\alpha(t, u, i) = 0$ if u does not fall within the window for base Y_t. It is straightforward to prove that the recursions for the forward and backward computations in the GPHMM remain correct. These reductions reduce the memory requirement to $O(hTN)$ and the number of computations to $O(hTN^2D^4)$. Smaller values of h reduce the size of the problem, but restrict the set of possible predicted matches more. The case $h = 1$ is equivalent to having a global alignment of the two sequences before processing. Small windows increase the dependency of the GPHMM result on the alignment; any relaxation allows for more robustness at the expense of computational time.

A natural method by which to find approximate alignments is to use an iterative/anchor method for finding a global alignment and stopping it before it finishes. An example of this is the GLASS program (Batzoglou *et al.*, 2000; Pachter, 1999), which uses fixed-length matches to iteratively build a global alignment. The program begins by finding all matches of some length k ($k \sim 35$ for BAC-sized sequences). These k-mer matches are aligned and anchored, after which the process is recursively repeated on the remaining intervening regions with a successively smaller value of k. To obtain a global alignment, the process is

terminated when $k = 1$. Terminating the process at a value of $k > 1$ results in an approximate alignment. The lower the terminal value of k, the more resolved the alignment.

It is interesting to note that approximate alignments generalize the notion of a banded Smith–Waterman problem (Gusfield, 1997). In particular, our approximate alignments can be thought of as generalized bands for an alignment problem. The approximate alignment problem is the problem of finding those constraints on the alignment that have the property that with high probability the optimal path lies inside the approximate alignment set.

3. APPLICATIONS

We focus here on the two closely related tasks of gene finding and of sequence alignment. As mentioned before, there is much overlap between the two problems, solving one helps substantially in the solution of the other. We show that the GPHMM framework is well suited to performing both tasks together.

One of the main strengths of GPHMMs for alignment and gene finding is the ease with which known biological signals and features can be incorporated into the model. Such features include introns, exons, promoters, and generally any type of signal for which there is a reasonable probabilistic model available. Another particularly convenient feature of using a GPHMM is the availability of natural likelihood-based methods for parameter estimation. The implementations of nonprobabilistic models usually run into the difficulty of having to heuristically determine parameters.

It was pointed out earlier that the ubiquitous Smith–Waterman algorithm for alignment can be seen as a particular implementation of a PHMM (Durbin *et al.*, 1998). The GPHMM generalizes such models, providing extensions of the Smith–Waterman algorithm to more general sequence alignment problems, which may involve different kinds of sequences originating from different organisms.

3.1. DNA–DNA

As more genomic DNA sequence becomes available from multiple organisms, the ability to identify conserved elements contained in long stretches becomes more important. Figure 1 shows the state space of a basic GPHMM for modeling genes in syntenic stretches of genomic DNA from two different organisms. Such a model can be used to exploit the information contained in the conservation of exon sequence to come up with more accurate gene predictions. The final product of the model is a global sequence alignment incorporating annotation of coding exons in both organisms simultaneously. Moreover, the probabilistic nature of the GPHMM allows for the determination of probabilities of the predictions being correct.

All states in the model generate bases or gaps in two organisms at a time; hence, it is a "pair" model. The intron and intergene states can be set up so that they always generate either a base in each organism, or a base in one organism and a gap in the other. The two sequences generated by such a model would each be geometrically distributed (with possibly different parameters), but the difference between the two lengths would in the limit converge to a Gaussian distribution, yielding highly correlated intron and intergenic lengths. An alternative approach is to generate introns and intergenic regions independently by using two consecutive self-transitioning states. The first state generates one sequence, and then the second state generates the other, resulting in independent geometric distributions for the sequence lengths. Regardless of the exact model that is used, it is convenient to represent these states as single states with a self-transition.

There are three types of intron states, one for each phase. An intron is said to be of phase 0 if it does not interrupt codons, a phase 1 intron interrupts a codon after its first base has been generated, and a phase 2 intron interrupts after the second base of the codon.

Geometric distributions are a poor fit for exon lengths, so the remaining exon states are all "generalized," each one generating two sequences (of possibly different lengths). The exon states each have two subscripts, characterizing their type. $E_{I,j}$ denotes an initial exon followed by a phase j intron, constraining its length to be equal to j mod 3 in each organism. Similarly, $E_{j,F}$ denotes a final exon following a phase j intron, and likewise for $E_{i,j}$. The exon states all have characteristic models at either end (start, splice, or stop signals) but are otherwise identical within.

The pair of sequences generated by the exon states is chosen from some joint distribution under which the lengths are constrained to be equal modulo 3 and are most likely to be the same. This has been found

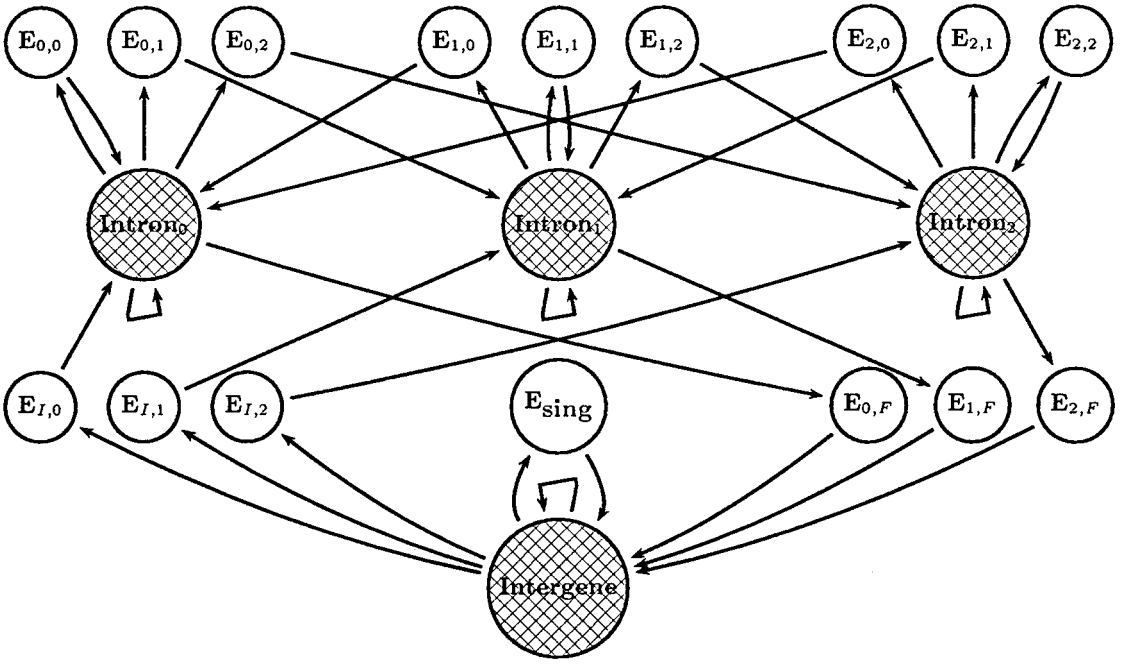

FIG. 1. A GPHMM for alignment and prediction of exons using genomic DNA from two different organisms. The shaded states are the typically less-conserved intergene and intron states, each producing either a single base or a gap in each organism. The use of self-transitions models their state durations as geometric. The unshaded states (all of which are exons) will all have duration one as they have no self-transitions; however, they are generalized and produce exon-pairs according to some predetermined joint distribution.

to be a reasonable model of the truth in studies so far (Batzoglou, 2000; Jareborg, 1999; Mironov, 1999). However, it is possible to modify the state space of the model to relax this constraint. PAM matrices (Burge and Karlin, 1997; Müller and Vingron, 1999) of an appropriate evolutionary distance can be used to model the pair of similar sequences produced. Quite a natural way to model exon states is to use a PHMM which for the most part generates codons in both organisms, but occasionally may produce a codon in one organism and a gap in the other (according to the appropriate PAM matrix).

As mentioned above, the model in Fig. 1 is very basic; it models genes on only one strand and produces the same exon sequence in each organism. Some of the restrictions are relatively simple to relax. Genes can be modeled on the reverse strand by the addition of a mirror image of the state space below the intergene state. The insertion and deletion of exons can be allowed for by the introduction of extra exon states which produce an exon in one organism and a gap in the other. Sequencing errors leading to exon pairs of lengths not equal modulo 3 can be allowed for by a simple adjustment of the exon model. Some of the restrictions are more rigid however. The assumption of the overall conservation of the order and direction of genes and their exons cannot easily be avoided by this model. In cases of large inversions and other rearrangements, some preprocessing of the data might restore the order. Insertions of introns within exons in one sequence are not easily handled. Another difficulty is exon pairs where the lengths differ modulo 3 due to introns of different phases (as opposed to sequencing errors). It is not clear how often this occurs in nature (it is estimated to happen in about 1% of genes for *H. sapiens* versus *M. musculus* [Pachter, 1999]).

We note that the Wobble Aware Bulk Aligner (WABA) of Kent and Zahler (2000) takes an initial step in the direction of the model we describe. Their model is simpler, not being generalized and using mainly the signal of the third base wobble to find coding sequences.

3.2. DNA–cDNA

Another useful application for GPHMMs is to the problem of aligning a full-length cDNA to genomic sequence (Florea *et al.*, 1998). Figure 2 presents the state space of such a GPHMM.

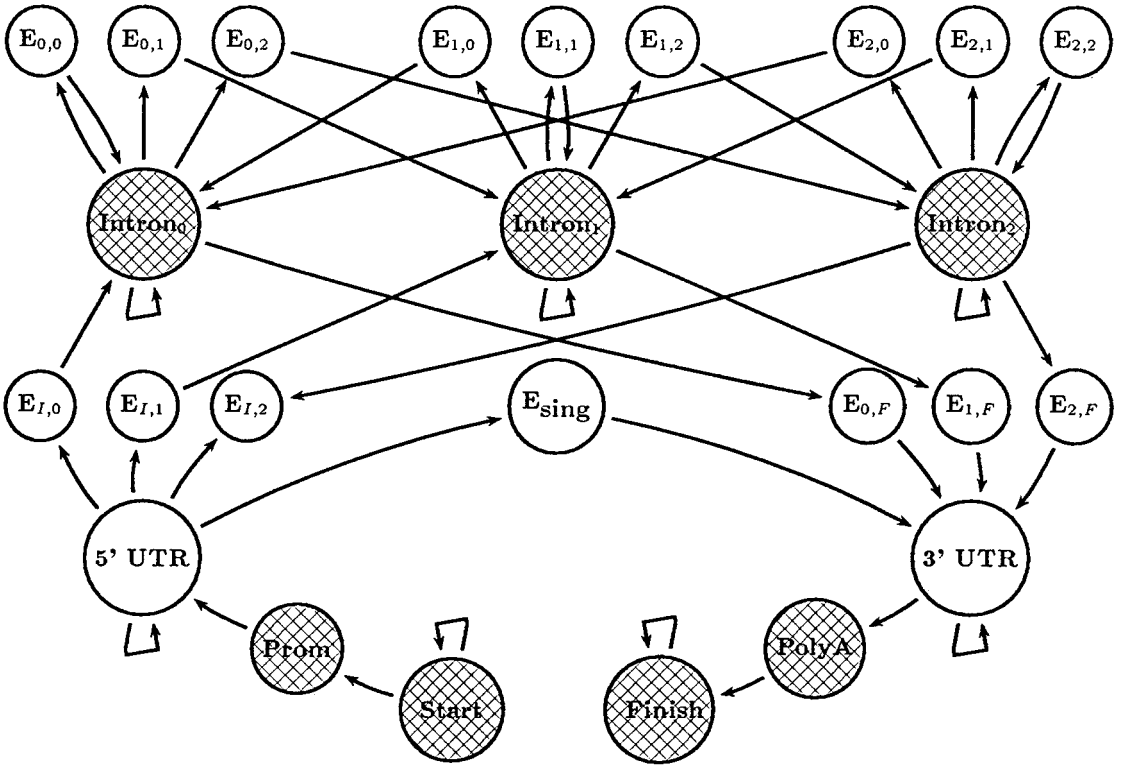

FIG. 2. A GPHMM for alignments between a (possibly long) stretch of genomic DNA and a full-length cDNA. The shaded states generate output in only one of the two sequences. The PolyA state generates a poly(A) tail in the cDNA; all the other shaded states generate sequences in the genomic DNA only. The unshaded states generate output in both the genomic and the cDNA sequences. As in Fig. 1, the states allowing self-transitions produce either single bases or gaps in each sequence, thereby having geometric lengths. The remaining states are generalized.

Unlike the model in Fig. 1, there is only one possible state to begin with, the state labeled "Start." All of the shaded states model regions that are not expected to apply to the cDNA sequence, and so they generate output in only the genomic sequence.

The exon and intron states work much as they did in the DNA–DNA example, except that the introns generate nothing in the cDNA sequence. In aligning cDNA to genomic DNA, there would be very little expected difference between the corresponding regions, so the exon and UTR states would have joint distributions with very high probability of producing identical sequences. The probability of producing identical sequences would depend chiefly upon the quality of the sequencing and assembly steps which produce the data.

The promoter (Prom) and polyadenylation (PolyA) states allow for the incorporation into the alignment of signals typically found upstream and downstream of the gene. The PolyA state generates a poly(A) tail, so it generates sequence in the cDNA only. Other states which could be introduced to improve performance would be splice site and intron states on a loop from each of the UTR states, which would allow for the prediction of introns between untranslated exons, and a poly(A) signal state which occurs towards the end of the 3'UTR state.

3.3. DNA–protein

GPHMMs are also well suited to the problem of aligning protein with genomic DNA. In fact, it can be modeled with the same state space as that of the DNA–cDNA model in the previous section. All that is required is the modification of the output distributions from each of the states.

The start, promoter, intron, polyadenylation, and finish states all work exactly as before. Since the protein sequence would only be expected to align with coding exons, the UTR states now only produce output in the genomic DNA sequence. The only remaining difference is that in the coding exon states a pair of sequences is produced over two different alphabets, amino acids and nucleotides. Construction of a joint distribution that produces a sequence of DNA and its translation is relatively simple.

There is a slight difficulty in treating output generated on either side of an intron of phase other than zero. The convention is that the next amino acid is produced with the third base of a codon. When finishing a codon spanning an intron of phase 1 or 2, the model does not allow for keeping track of the previous bases of the codon.

As in the case of DNA–cDNA alignment, it has been established that the incorporation of the biological signals assists DNA–protein alignment. Usuka and Volker (2000) showed that the inclusion of splice site strength led to better alignment in *Arabidopsis thalania.* Gotoh (2000) developed a model handling coding potential, translation initiation, termination, and splicing signals in addition to sequence similarity, leading to improved performance. The GPHMM approach can handle all of these features and more. As mentioned before, the GPHMM has the added benefit of its parameters being easy to estimate in a theoretically sound manner. We note that there are other methods by which to align DNA sequences to protein sequences. For example, the program GeneWise (Birney and Durbin, 2000) tackles the DNA–protein problem by using an HMM to find optimal alignments between DNA and a protein profile-HMM. Since the profile-HMMs it uses can be thought of as representing families of proteins, it tackles a slightly different kind of alignment problem to the one we describe here.

4. CONCLUDING REMARKS

The GPHMMs described here were developed in the course of building a probabilistic two-organism cross-species gene-finding program. The results can be applied to that problem, although numerous technical issues need to be resolved (Alexandersson *et al.*, 2001). Parameters for the models can be estimated from counts using the methods outlined by Durbin *et al.* (1998). We are currently finishing software for the implementation of the two-organism gene-finding GPHMM. Extension of the theory to more than two organisms is possible, although the computational complexity increases dramatically and the use of approximate alignments becomes more important. It is interesting to note that the GPHMM we describe for DNA–DNA alignment and gene finding directly generalizes the existing HMM gene finders such as GENSCAN, GENIE and HMMGene (Burge and Karlin, 1997; Krogh, 2000; Kulp *et al.*, 1996) and can be used for single-organism gene finding as a special case (assuming one uses the same splice site and other signal models). This is accomplished by forcing each state to always generate a pair of identical outputs and by restricting the joint transition and length distributions to be univariate.

The computational burden associated with GPHMMs is mitigated by the use of approximate alignments. It is an interesting computational problem to develop fast algorithms for finding approximate alignments that substantially reduce the search space, but at the same time contain the "true" alignment with high probability.

ACKNOWLEDGMENTS

We thank Terry Speed for valuable comments. M.A. was supported by STINT, the Swedish Foundation for International Cooperation in Research and Higher Education.

REFERENCES

Alexandersson, M., Cawley, S., and Pachter, L. 2002. SLAM: Cross-species gene finding and alignment with a generalized pair hidden Markov model. Submitted.

Bafna, V., Huson, D.H. 2000. The conserved exon method for gene finding. *Proc. Int. Conf. Intelligent Systems for Molecular Biology.*

Batzoglou, S., Pachter, L., Mesirov, J., Berger, B., and Lander, E.S. 2000. Comparative analysis of mouse and human DNA and applications to exon prediction. *Genome Res.* 10(7), 950–958.

Birney, E., and Durbin, R. 2000. Using GeneWise in the *Drosophila* annotation experiment. *Genome Res.* 10(4) 547–548.

Burge, C. 1997. Identification of genes in human genomic DNA. PhD Thesis, Stanford University, Stanford, CA.

Burge, C., and Karlin, S. 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 268, 78–94.

Cawley, S. 2000. Statistical models for DNA sequencing and analysis. PhD. Thesis, Department of Statistics, University of California, Berkeley.

Churchill, G.A. 1989. Stochastic models for heterogeneous DNA sequences. *Bull. Math. Biol.* 51, 79–94.

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1978. A model of evolutionary changes in proteins. In M.O. Dayhoff, ed., *Atlas of Protein Sequence and Structure*, vol. 5, supp. 3, 345–352, National Biomedical Research Foundation, Washington, D.C.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. 1998. *Biological Sequence Analysis*, Cambridge University Press, Cambridge, UK.

Florea, L., Hartzell, G., Zhang, Z., Rubin, G.M., and Miller, W. 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.* 8, 967–974.

Gelfand, M.S., Mironov, A., and Pevzner, P.A. 1996. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. USA* 93, 9061–9066.

Gotoh, O. 2000. Homology-based gene structure prediction: Simplified matching algorithm using a translated codon (tron) and improved accuracy by allowing for long gaps. *Bioinformatics* 16(3), 190–202.

Gusfield, D. 1997. *Algorithms on Strings, Trees and Sequences.* Cambridge University Press, Cambridge, UK.

Henderson, J., Salzberg, S., and Fasman, K. 1997. Finding genes in human DNA with a hidden Markov model. *J. Comp. Biol.* 4(2), 127–141.

Holmes, I. 1998. *Studies in Probabilistic Sequence Alignment and Evolution.* PhD. Thesis, University of Cambridge, UK.

Jareborg, N., Birney, E., and Durbin, R. 1999. Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Res.* 9(9), 815–824.

Kent, W., and Zahler, A. 2000. Conservation, regulation, synteny, and introns in a large-scale *C. briggsae–C. elegans* genomic alignment. *Genome Res.* 10(8), 1115–1125.

Korf, I., Flicek, P., Duan, D., and Brent, M. 2001. Integrating genomic homology into gene structure prediction. *Bioinformatics* 1(1), S1–S9.

Krogh, A. 2000. Using database matches with HMMGene for automated gene detection in drosophila. *Genome Res.* 10(4) 523–528.

Kulp, D., Haussler, D., Reese, M.G., and Eeckman, F.H. 1996. A generalized hidden Markov model for the recognition of human genes in DNA. *Proc. 4th Int. Conf. Intelligent Systems for Molecular Biology*, 134–141.

Makalowski, W., Zhang, J., and Boguski, M.S. 1996. Comparative analysis of 1196 orthologous mouse and human full-length mRNA and protein sequences. *Genome Res.* 6, 846–857.

Mironov, A.A., Fickett, J.W., and Gelfand, M.S. 1999. Frequent alternative splicing of human genes. *Genome Res.* 9, 1288-1293.

Müller, T., and Vingron, M. 1999. Modeling amino acid replacement. *J. Comp. Biol.* To appear.

Needleman, S.B., and Wunsch, C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453.

Pachter, L. 1999. Domino tiling, gene recognition, and mice. PhD. Thesis, Department of Mathematics, Massachusetts Institute of Technology.

Pachter, L., Batzoglou, S., Spitkovsky, V.I., Banks, E., Lander, E.S., Berger, B., and Kleitman, D.J. 1999. A dictionary based approach for gene annotation. *J. Comp. Biol.* 6, 419–430.

Rabiner, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77(2), 257–286.

Reese, M.G., Kulp, D., Tammana, H., and Haussler, D. 2000. Genie—gene finding in *Drosophila melanogaster. Genome Res.* 10(4), 529-538.

Salzberg, S.L. 1998. Decision trees and Markov chains for gene finding. In Salzberg, Searls, Kasif, eds. *Computational Methods in Molecular Biology*, 187–203.

Searls, D.B., and Murphy, K. 1995. Automata-theoretic models of mutation and alignment. *Proc. 3rd Int. Conf. Intelligent Systems for Molecular Biology*, 341–349.

Smith, T.F., and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195-197.

Usuka, J., and Volker, B. 2000. Gene structure prediction by spliced alignment of genomic DNA with protein sequences: Increased accuracy by differential splice site scoring. *J. Mol. Biol.* 297(5), 1075–1085.

Wiehe, T., Burset, M., Abril, J., Gebauer-Jung, S., and Guigo, R. 1999. Comparative genomics: At the crossroads of evolutionary biology and genome sequence analysis. Poster at Meeting of the European Society for Molecular Biology and Evolution, Barcelona.

Wirth, A. 1998. *A* Plasmodium Falciparum *Genefinder.* Honours Thesis, Department of Mathematics and Statistics, University of Melbourne.

Address correspondence to:
Lior Pachter
University of California, Berkeley
Department of Mathematics
970 Evans Hall
Berkeley, CA 94720

E-mail: lpachter@math.berkeley.edu