Optimal Spaced Seeds for Hidden Markov Models, with Application to Homologous Coding Regions

Broňa Brejová, Daniel G. Brown, and Tomáš Vinař

School of Computer Science, University of Waterloo, Waterloo ON N2L 3G1 Canada
{bbrejova, browndg, tvinar}@math.uwaterloo.ca

Abstract. We study the problem of computing optimal spaced seeds for detecting sequences generated by a Hidden Markov model. Inspired by recent work in DNA sequence alignment, we have developed such a model for representing the conservation between related DNA coding sequences. Our model includes positional dependencies and periodic rates of conservation, as well as regional deviations in overall conservation rate. We show that, for hidden Markov models in general, the probability that a seed is matched in a region can be computed efficiently, and use these methods to compute the optimal seed for our models. Our experiments on real data show that the optimal seeds are substantially more sensitive than the seeds used in the standard alignment program BLAST, and also substantially better than those of PatternHunter or WABA, both of which use spaced seeds. Our results offer the hope of improved gene finding due to fewer missed exons in DNA/DNA comparison, and more effective homology search in general, and may have applications outside of bioinformatics.

1 Introduction

Consider a probabilistic process P that generates finite binary sequences. We wish to compute the probability that a given pattern s matches the random sequence, for a particular definition of pattern matching. Then, given that the pattern s comes from a small class of possible patterns, we wish to find the pattern s^* with the highest probability of matching the random sequence A.

Our form of pattern matching is based on *spaced seeds*. In this framework, one chooses a seed, represented by a binary string $s = (s_1, \ldots, s_k)$. In the seed, ones intuitively correspond to places in the random sequence which must also be ones, while zeros correspond to wildcards. Formally, for the seed s to match the randomly generated sequence A, there must be a position i in A such that $a_i \geq s_1, a_{i+1} \geq s_2, \ldots, a_{i+k-1} \geq s_k$.

Here, we wish to identify the optimal seed s^* when the random process is a hidden Markov model (HMM). For such models, we show that one can identify the probability that a given seed detects a sequence generated from an HMM efficiently. When k is small, one can examine all seeds efficiently to find the best one. We use our methods to find optimal seeds for finding homologous protein coding regions in long DNA sequences. The optimal seed for a reasonable HMM for this problem is more effective than seeds used by existing methods.

A launching point for our work is the recent study of DNA sequence homology by Ma et al. [8], who showed that spaced seeds can be useful in increasing the probability of detecting related DNA regions. The process of identifying related DNA sequences, large scale similarity search, is one of the most time consuming processes in bioinformatics, so procedures that improve it are needed. Ma et al.'s local alignment program PatternHunter, because of its use of spaced seeds tuned for a particular model, is more successful at detecting homologous regions, and thus is more likely to build useful local alignments between two sequences.

The model used by Ma et al., $PH(N, p)$, assumes that correct local alignments are of length N, and that in comparing two homologous sequences, the probability that they are the same at position i is p. Keich et al. [5] give a dynamic programming algorithm for computing the optimal seed in this simple probabilistic model. In $PH(64, .7)$, the seed of size less than or equal to 18, with 10 1's, that is most likely to detect homologous sequences is 111001001001010111.

Traditional programs for sequence alignment, like BLASTN [1], use in effect a seed of the form 1111111111, and only detect homologous sequences with a set of contiguous exact matches. The expected number of matches of the BLAST seed to a sequence generated by $PH(64, .7)$ is quite close to the expected number of matches of the PatternHunter seed, but the probability that an alignment has any match to the seed is substantially higher with PatternHunter's seed.

In Section 2, we review HMMs, and in Section 3, we consider the question of identifying the best seeds when the random process is more complicated, in particular an HMM. If the sequence A is given by a known HMM, we give an efficient algorithm that computes the seed s^* most likely to match A, given a distribution on the lengths of the sequence A.

One application of these techniques is increasing the probability of detecting homologous protein coding regions in DNA sequences. Protein coding regions form most significant local alignments between distantly related species, and are also specifically important. Alignments between homologous coding regions can, for example, aid in gene finding [7]. Alignments between homologous protein coding regions, however, are very poorly modeled by PH, as it does not take into account basic characteristics of protein alignments, such as their three-periodicity and the inhomogeneity of conservation rates within alignments.

In Section 4, we give an HMM modeling conservation within alignments of coding regions, and in Section 5, we use compute the optimal seeds under this model, using data, based on curated alignments between the human genome and the mouse and *Drosophila* (fruit fly) genomes. Comparisons between the seeds' predicted sensitivity and their actual sensitivity show that the HMM does more accurately model the probability that a seed is useful in practice than do simpler models, and similarly, we find that the seeds which our algorithms predict as having the highest probability of success are, in practice, optimal. In particular, the best seed with 10 ones and length less than or equal to 18, according to the algorithm, is the second best in our testing set, and detects 85% of alignments. This compares with 45% for the BLAST seed, 59% for the PatternHunter seed, and 82% for the seed 11011011011011, used by the program WABA [6].

We note that our methods were developed contemporaneously with other work by Buhler et al. [3], who extend Keich et al.'s algorithm [5] to the case of multiple seeds used for the same model. However, they consider neither HMMs, nor probabilistic models specifically developed for coding regions.

2 Probabilistic Models of Local Alignments

Local alignments can be represented as binary sequences, where `1` represents a match and `0` a mismatch. We model only ungapped alignments, as seed hits are found only in gapless regions. We can characterize a particular type of local alignment by giving a probabilistic model defining a probability distribution over all binary strings of a given length. Then, given a particular probabilistic model, we can find a seed s^* from a class of seeds C maximizing the probability of a hit in an alignment sampled from this model. For us, the class C is all seeds with W ones and length at most M. To avoid redundancy, we require that the first and last positions of a seed are `1`. The parameter W is the *weight* of a seed.

Ma et al. [8] introduced the concept of optimal seeds in PatternHunter. PatternHunter's model $PH(N,p)$ represents a similarity region of length N, where each position is a match independently with probability p. Formally, it is a sequence of N independent Bernoulli random variables $X_0, X_1, \ldots, X_{N-1}$, with $\Pr(X_i = 1) = p$ for each i. PatternHunter's seed optimizes this model with parameters $N = 64$ and $p = 0.7$. Keich et al. [5] give a dynamic programming algorithm to compute the probability that a given seed of length M has at least one hit in an alignment sampled from $PH(N,p)$. The running time is $O(2^{M-W}M(M^2+N))$ for each seed. One computes the probability for each seed in C and chooses the best.

Here, we explore the problem of finding the optimal seed under more general class of probabilistic models, namely hidden Markov models (HMMs). HMMs model dependencies between adjacent positions in the sequence and have successfully modeled biological sequences in a number of applications [4]. Here, we show that HMMs can capture several important properties of DNA alignments of protein coding regions.

An HMM is a simple generative process defined on a collection of states and transitions between the states. First, a starting state is chosen according to initial probabilities associated with each state. In each step, a single character is generated according to the emission probability table associated with the current state. Then, a new state is chosen according to the transition probabilities for the current state. The probability of a string in the model equals the probability that the string is generated by this process.

In the next section, we present an algorithm for computing the probability that a seed has a hit in a string sampled from an HMM. Our algorithm is an extension of the algorithm presented by Keich et al. [5]. In particular, the *PH* probabilistic model is equivalent to a single state HMM, and in this special case, our algorithm is identical to the one presented by Keich et al.

We also extend the probabilistic model so that the length of the alignment is also a random variable, chosen from some fixed distribution of alignment lengths. The probability of a seed match in such an extended model is the weighted average of the probabilities for models with fixed lengths.

3 Finding Optimal Seeds for HMMs

Assume that we are given a seed Q of length M and weight W and an HMM characterizing alignments. We will compute the probability that Q has at least one hit in a length N alignment.

Let S be the set of states of the HMM. Each step of the generative process consists of emission of one character in the current state, followed by transition to the next state. Let $\mathit{String}(s, x)$ be the event that the HMM generates string x, assuming that it starts in state s. Let $\mathit{State}(s, t, i)$ be the probability that after i steps the HMM will be in state t, assuming that it starts in state s. Notice that $\Pr(\mathit{State}(s, 0, s)) = 1$ and $\sum_{x \in \{0,1\}^i} \Pr(\mathit{String}(s, x)) = 1$, for all i.

We will proceed by dynamic programming, where the subproblem is defined as follows: for any $i \leq N$, binary string x of length at most $\min\{i, M\}$, and $s \in S$, the subproblem $A_{i,x,s}$ is the probability that the string generated by the HMM in i steps starting in state s contains a match of the seed Q, provided that x is a prefix of the generated string. The probability that Q has at least one hit in the entire alignment then equals the sum of $A_{N,\lambda,s}$ over all states s, weighted by the initial probabilities of all states (λ denotes the empty string).

Let $\mathit{matches}(x, Q)$ be the event that string $x\mathtt{1}^{M-|x|}$ is a hit for seed Q. That is, when we align the starts of x and Q, x has `1` at all positions where Q has `1`. If $\mathit{matches}(x, Q)$, we say that x is a matching string. Let $\mathit{suffix}(x, Q)$ be the longest suffix z of x such that z is a matching string.

The probability $A_{i,x,s}$ can be computed by the following recurrent formula:

$$A_{i,x,s} = \begin{cases} 0 & \text{if } i < M \quad \text{(A)} \\ 1 & \text{if } |x| = M \text{ and } \mathit{matches}(x, Q) \quad \text{(B)} \\ \sum\limits_{t \in S} p_t \cdot A_{i-|y|,z,t} & \begin{array}{l} \text{if not } \mathit{matches}(x, Q)\text{, where} \\ \quad x = yz,\ z = \mathit{suffix}(x, Q), \\ \quad p_t = \Pr[\mathit{State}(s, t, |y|) \mid \mathit{String}(s, x)] \quad \text{(C)} \end{array} \\ \begin{array}{l} p \cdot A_{i,x\mathtt{1},s} \\ +(1-p) \cdot A_{i,x\mathtt{0},s} \end{array} & \begin{array}{l} \text{if } |x| < M \text{ and } \mathit{matches}(x, Q)\text{, where} \\ \quad p = \Pr[\mathit{String}(s, x1) \mid \mathit{String}(s, x)] \quad \text{(D)} \end{array} \end{cases}$$

Case (A) recognizes that seeds of length M cannot have hits in shorter regions. In case (B), string x guarantees a hit. In case (C), since the string x does not match Q, a hit will not start at the first position of the alignment, so only following positions need to be considered. In particular, the longest matching suffix z of string x corresponds to the first possible position where a hit can occur. We need to consider all possible states t in which the HMM can start to generate suffix z. Finally, case (D) provides a formula for combining subproblems where the last $|x|+1$ characters are fixed into a subproblem where only $|x|$ characters are fixed.

This recurrent formula can be used to compute probabilities $A_{i,x,s}$ in order of increasing i and shrinking x. Keich et al. [5] show how to efficiently organize the computation of a similar recurrent formula so that the running time is $O(M2^{M-W}(M^2+N))$ for each of the $\binom{M-2}{W-2}$ possible seeds (under the assumption that numbers of size between 0 and 2^{M-W} can be manipulated in constant time). We apply their methods to our modified algorithm yielding the following Lemma.

Lemma 1. *The probability that a length M, weight W seed Q has a hit in an alignment of length N generated by a given σ-state HMM can be computed in $O(M2^{M-W}(M^2+\sigma^3+\sigma^2 N))$ time.*

Proof. To complete the dynamic programming algorithm we need to demonstrate how to compute the probabilities needed in cases (C) and (D). First, the standard Forward algorithm for HMMs [9] can be used to compute values $B_{s,t,x}$ and $C_{s,x}$ defined as follows:

$$B_{s,t,x} = \Pr(\mathit{String}(s,x) \text{ and } \mathit{State}(s,t,|x|))$$

$$C_{s,x} = \Pr(\mathit{String}(s,x)) = \sum_{t\in S} B_{s,t,x}$$

The conditional probabilities required can now be computed using these two formulas:

$$\Pr(\mathit{State}(s,t,|y|) \mid \mathit{String}(s,\overbrace{yz}^{x})) = \frac{B_{s,t,y}\cdot C_{t,z}}{C_{s,x}} \tag{1}$$

$$\Pr(\mathit{String}(s,x1) \mid \mathit{String}(s,x)) = \frac{\sum_t B_{s,t,x}\cdot C_{t,1}}{C_{s,x}} \tag{2}$$

Let $\Gamma = \{x, x0 \mid x \text{ is a matching string}, |x| \le M\}$. The number of strings in Γ is $O(M2^{M-W})$. Observe that to compute values $A_{N,\Lambda,s}$, we do not need to compute values of $A_{i,x,s}$ for all x, but it is sufficient to compute the values for $x \in \Gamma$.

A value of $\mathit{suffix}(x0,Q)$ can be easily computed in $O(M^2)$. Then we compute values of $B_{s,t,x}$ for each pair of states s and t and for each $x\in\Gamma$. Using the fact that Γ is closed under prefix operation this can be done in $O(\sigma^3 M2^{M-W})$ time. It is possible to prove that only values of $B_{s,t,x}$ for $x\in\Gamma$ are needed to compute probabilities from formulas (1) and (2) for all $x\in\Gamma$. In particular, notice that if $x\in\Gamma$ and $x=yz$ where $z=\mathit{suffix}(x,Q)$, then both y and z are also in Γ.

Once all auxiliary tables are computed, each value $A_{i,x,s}$ can be computed in $O(\sigma)$ time. This gives total time $O(M2^{M-W}(M^2+\sigma^3+\sigma^2 N))$. □

4 Modeling Local Alignments in Protein Coding Regions

Now that we have demonstrated that the notion of optimal seeds can be extended to more general probabilistic models, namely HMMs, we apply this approach to the problem of detecting local alignments of homologous protein coding regions. These have specific properties not modeled well by the simple *PH* model. Also, because protein coding regions are very important, local alignment programs must perform well on these regions of genomes.

Protein coding regions encode an organism's proteins in its genome. A protein's amino acid sequence is encoded in DNA as a sequence of triplet codons, each representing one amino acid according to the genetic code. However, the code is redundant: some amino acids have multiple encodings. In particular, third positions in codons can often mutate "silently," not altering the amino acid, and even similar proteins can differ greatly in corresponding coding regions in DNA.

In higher organisms, a protein's coding DNA is often split into disjoint regions, called exons, separated by long stretches of non-coding DNA, called introns. Exons are cut out and spliced together during protein production. Here, we assume that the locations of the protein coding regions is unknown prior to local alignment. Because of the existence of introns, finding corresponding coding regions is difficult, since a protein alignment may correspond to several short alignments of exonic DNA, separated by long intron gaps.

Some local alignment programs, such as BLASTN [1] and PatternHunter [8], use seeds not reflecting properties of protein coding regions. Thus, they are not able to detect some related pairs of coding sequences. By contrast, Kent and Zahler [6] use the seed `110110110` in their alignment program, WABA. This seed represents the intuition that coding DNA has 3-periodic structure, and the third position of a codon is less conserved than the first two. Kent and Zahler report a substantial improvement over BLAST in sensitivity in detecting homologous coding sequences. We obtain better seeds for protein coding regions by modeling properties of alignments in these regions by probabilistic models. We describe three models increasing in complexity and accuracy.

Three-periodicity. The most obvious property of alignments in coding regions is their three-periodicity and that some of the positions of a codon are less conserved than others. A simple extension of the *PH* model, which we call $M^{(3)}$, encapsulates this idea. Model $M^{(3)}(N, p_0, p_1, p_2)$ represents a region of length N, where the probability of a position being a match depends on its relative codon position, but alignment positions are still independent. Formally, it is a sequence of N independent Bernoulli random variables $X_0, X_1, \ldots, X_{N-1}$ where $\Pr(X_i = 1) = p_{i \bmod 3}$. This model can be expressed as a simple 3-state HMM, as depicted in Figure 1, and we can use our algorithm to compute the probability that a seed matches an alignment sampled from this model.

Table 1 shows the parameters of the model $M^{(3)}$ estimated from our training set of alignments between human and fruit fly coding regions and between human and mouse coding regions (more details about the data sets can be found in

Data set	p_0	p_1	p_2
human/mouse	0.82	0.87	0.61
human/fruit fly	0.67	0.77	0.40

Table 1. Parameters of model $M^{(3)}$ estimated from our training set.

Section 5). In both cases, the third position is much less conserved than the others, and the first position is somewhat less conserved than the second.

Dependencies within codon. Model $M^{(3)}$ models the different conservation levels within codons arising from the redundancy of the genetic code. However, dependencies exist among positions within codons. These arise in part from the differing prevalence of some amino acids and their corresponding codons. To model these dependencies, we use a new model, $M^{(8)}$.

The model $M^{(8)}(n, p_{\mathtt{000}}, p_{\mathtt{001}}, \ldots, p_{\mathtt{111}})$ represents a region of n codons ($N = 3n$ nucleotides). Each codon has conservation pattern $x \in \{\mathtt{0}, \mathtt{1}\}^3$ with probability p_x; the sum of $p_{\mathtt{000}}, p_{\mathtt{001}}, \ldots, p_{\mathtt{111}}$ is 1. In this model, positions within one codon have arbitrary dependencies, yet individual codons are independent. Formally, this model is a sequence of n independent triples of Bernoulli random variables $(X_0, X_1, X_2), \ldots, (X_{3n-3}, X_{3n-2}, X_{3n-1})$, such that $\Pr(X_{3i} = a, X_{3i+1} = b, X_{3i+2} = c) = p_{abc}$. The HMM shown in Figure 1 represents model $M^{(8)}$.

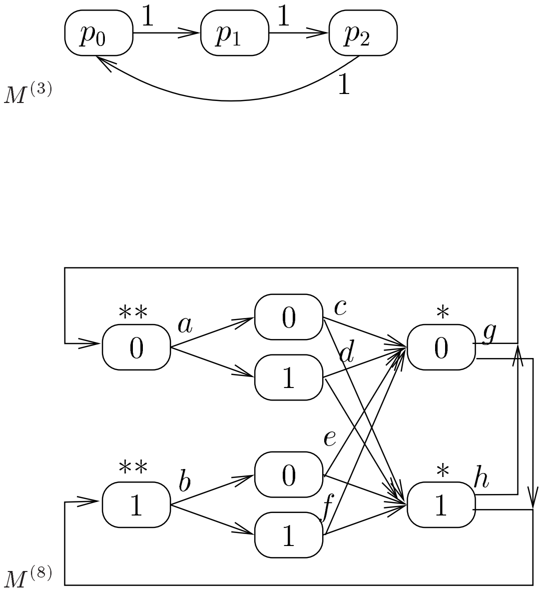
Table 2 demonstrates the existence of dependencies within codon. The first row of the table shows the probabilities of all triplets as estimated by model $M^{(3)}$, under the assumption that codon positions are independent. The second row shows the probabilities of all triplets in model $M^{(8)}$. In particular, triplets `000` and `111` occur in the data more often than expected based on model $M^{(3)}$.

Inhomogeneity of alignments. The previous models assume that alignments have roughly the same conservation rate throughout their length. In fact, the pattern of conservation of a typical alignment is highly non-uniform. Many alignments include short, highly conserved regions surrounded by less well conserved regions. (This is unsurprising, as highly conserved regions are more likely to be functional parts of the proteins [10].)

We address this problem with a new HMM. In this model, regions with high and low conservation can alternate. This model is an HMM consisting of 4 copies of the HMM for $M^{(8)}$ shown in Figure 1, each copy corresponding to a different level of conservation. To allow transitions between different conservation levels, we add transitions from all copies of both states labeled by a star in Figure 1 to all copies of both states labeled by two stars. In our experiments this model will be called *HMM*.

5 Experimental Data

Data and methods. We have conducted our experiments on two datasets of protein coding region alignments: human vs. fruit fly and human vs. mouse, each



$$a = \frac{p_{000} + p_{001}}{p_{000} + p_{001} + p_{010} + p_{011}}$$

$$b = \frac{p_{100} + p_{101}}{p_{100} + p_{101} + p_{110} + p_{111}}$$

$$c = \frac{p_{000}}{p_{000} + p_{001}}$$

$$d = \frac{p_{010}}{p_{010} + p_{011}}$$

$$e = \frac{p_{100}}{p_{100} + p_{101}}$$

$$f = \frac{p_{110}}{p_{110} + p_{111}}$$

$$g = h = p_{000} + p_{001} + p_{010} + p_{011}$$

Fig. 1. HMM representation of models $M^{(3)}(N, p_0, p_1, p_2)$ and $M^{(8)}(n, p_{000}, \ldots, p_{111})$. Each state is labeled with emission probability of '1'. In the HMM for $M^{(3)}$, each transition has probability 1. In the HMM for $M^{(8)}$, each state has two outgoing transitions. One of them has the transition probability shown in the picture; the other has probability such that they add to 1.

Model	p_{000}	p_{001}	p_{010}	p_{011}	p_{100}	p_{101}	p_{110}	p_{111}
$M^{(3)}$	0.05	0.03	0.15	0.10	0.09	0.06	0.31	0.20
$M^{(8)}$	0.11	0.04	0.12	0.06	0.06	0.03	0.32	0.27

Table 2. Comparison of probabilities of conservation patterns within a codon in models $M^{(3)}$ and $M^{(8)}$. Parameters of models estimated from human/fruit fly alignments.

treated separately. First, we found statistically significant interspecies protein alignments. Then, we filtered these so that each protein occurs in at most one alignment. Next, we split the resulting sets into testing and training halves. The protein sequences were then mapped to their coding sequences in the genomes. In this way, we transformed alignments of pairs of proteins into alignments of pairs of DNA sequences. Note that one protein alignment can yield several DNA alignments because the coding regions for each protein can be interrupted by non-coding introns. We call the DNA alignments in this set *fragments*. We discarded weakly conserved and very short fragments because they cannot be detected by alignment programs using the spaced seed method.

The fragments may be gapped alignments. However, to find a fragment by an alignment program using the spaced seed method, the seed hit must be within an ungapped region. Thus, we broke gapped fragments in the training set into ungapped fragments, and again discarded weak and short fragments. In the testing set, we have kept gapped fragments, because a single hit is sufficient to discover the entire fragment by the spaced seed method.

This process ensures that our data sets contain biologically meaningful nucleotide alignments of protein coding regions. Only coding regions of related proteins are aligned, codon boundaries are always correctly aligned, and alignments do not extend to non-coding parts of the genome.

The initial data set consisted of all human, fruit fly and mouse proteins from SWISSPROT database [2], release 40.38, for which a correctly annotated coding regions could be found in the GenBank database. We created the initial alignments by BLASTP 2.0.8 [1] using E-value threshold 10^{-30}. The resulting set contained 339 human vs. fruit fly protein alignments and 675 human vs. mouse protein alignments. In the final data sets we filtered out gapped fragments with alignment score less than 16 (scoring scheme: match $+1$, mismatch -1, gap opening -5, gap extension -1) and ungapped fragments with less than 10 matches. The human vs. fruit fly dataset contains 972 ungapped fragments in the training set and 810 gapped fragments in the testing set. The human vs. mouse dataset contains 2171 ungapped fragments in the training set and 1660 gapped fragments in the testing set.

We used the training set of ungapped fragments to estimate the parameters of PH, $M^{(3)}$, $M^{(8)}$, and HMM. We estimated the parameters of PH, $M^{(3)}$, and $M^{(8)}$ by counting frequencies of corresponding conservation patterns, while we estimated the parameters of the HMM model by the Baum-Welch algorithm [4]. We estimated the length distribution of alignments from the set of ungapped fragments as well.

In our experiments we studied all 24310 seeds of weight 10 and length at most 18. We used the dynamic programming algorithm to compute the probability of a hit in a random alignment for each seed under each of the models.

To evaluate the seed performance, we have computed for each seed how many gapped alignments from the testing set contain a hit of the seed and can be thus potentially found by an alignment program using this seed. The results of these experiments follow.

Optimal seeds and their performance. Here, we present the optimal seeds under different probabilistic models of alignments, and we compare their sensitivity on the testing data set as well as sensitivity when using the seeds with alignment program PatternHunter.

Table 3 lists the seeds we selected for further examination, and presents the sensitivity of each seed on both testing sets. We also evaluated the seeds with the program PatternHunter and we list the fraction of testing fragments overlapping an alignment reported by PatternHunter.

On the human vs. fruit fly data set, the seed DATA-OPT has the highest sensitivity on the testing set. It was not discovered by any model. Models HMM and $M^{(8)}$ both discovered the seed HMM-OPT, which has almost the same performance as DATA-OPT. The seed M3-OPT is optimal under model $M^{(3)}$, but its performance is lower than that of the other seeds. We also include seeds used by WABA, PatternHunter and BLAST for comparison. Out of these, only WABA performs comparably with M3-OPT. We also include the seed globally worst on the testing set.

Results are similar for the human vs. mouse data set. Here, models $M^{(8)}$ and $M^{(3)}$ found the seed DATA-OPT, which is again optimal for the testing data. Model *HMM* reported the seed HMM-OPT again. In this case, there is much less difference between the best and the worst seeds, since alignments between human and mouse are much more conserved than alignments between drosophila and mouse, and are thus easier to detect by any seed. Still, there is a clear separation between performance of seeds tailored to detecting alignments in coding regions and other seeds.

Seed		Fruit fly		Mouse	
		Hit	PH	Hit	PH
11011011000011011	DATA-OPT	86%	85%	94%	92%
11011000011011011	HMM-OPT	85%	85%	94%	92%
11001011001011011	M3-OPT	82%	76%	93%	90%
11011011011011	WABA	81%	79%	92%	90%
111001001001010111	PH-OPT	59%	57%	87%	86%
1111111111	BLAST	45%	43%	82%	81%
101010101010101011	WORST	38%	39%	80%	79%

Table 3. Performance of selected seeds on both testing sets. Columns labeled Hit indicate how many fragments in the testing set have a hit of the seed. Columns labeled PH indicate how many of these fragments overlap an alignment discovered by the PatternHunter program using the seed.

In both data sets, the seeds that take into account the 3-periodic structure of the coding regions (WABA, M3-OPT, HMM-OPT and DATA-OPT), have higher sensitivity than other seeds. Further, the optimal seeds under models *HMM* and $M^{(8)}$ have significantly better performance than the WABA seed, currently used for local alignments in coding regions [6]. By using these seeds we were able to reduce the number false negatives by 29% in the fruit fly set and by 20% in the mouse set.

This increase in sensitivity does not come at a cost of greatly increased running time. The running time of PatternHunter increased by at most 3% compared to the original PH-OPT seed.

Our models as predictors of seed performance. For further validation, we studied how well the sensitivity predicted under each of the models corresponds to the sensitivity measured on the testing data set. Good models should assign higher probability to seeds which perform better in practice, and the probability predicted by the model should correspond to sensitivity on real data.

As illustrated in Figure 2, the predicted probability increases with the sensitivity of the seed on testing data in models $M^{(3)}$, $M^{(8)}$, and HMM. This is in contrast to the model PH where there is no clear correspondence between predicted and real sensitivity. Models HMM and $M^{(8)}$ exhibit better quality of ordering among the top seeds as well as among the worst seeds. In addition to that, HMM is clearly the best predictor of the real sensitivity; in fact, the correlation between the estimated sensitivity by HMM and the actual sensitivity is the highest, at $r^2 = 0.9687$. Sensitivity is consistently underpredicted in all models, since the training set consisted of ungapped fragments, which are slightly shorter than the gapped fragments in the testing set.

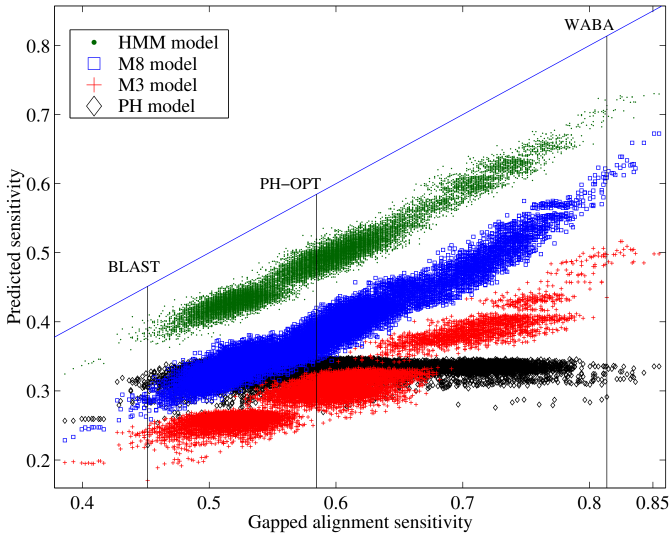


Fig. 2. Sensitivity of all seeds on the testing set versus their predicted sensitivity under different probabilistic models. Each dot represents one seed under one model. Vertical lines indicate sensitivity of the WABA, PH-OPT and BLAST seeds.

Table 4 further demonstrates the ordering capabilities of each of the models. Models HMM and $M^{(8)}$ correctly identified the top 3 seeds, while the seed classified as the best in $M^{(3)}$ model ranks 15th on the testing data. Also, there is no clear correspondence between ranks in PH model and rank on the testing data. The BLAST seed (which is currently the most widely used) is among the worst seeds in any of the considered models.

6 Conclusion

We have given an algorithm to compute the optimal spaced seeds for detecting sequences generated by a hidden Markov model. By extending work of Keich et

Seed	Testing data		Rank under a prob. model				Name
	Rank	Sensitivity	HMM	$M^{(8)}$	$M^{(3)}$	PH	
11011011000011011	1	0.855	2	2	17	9746	DATA-OPT
11011000011011011	2	0.851	1	1	16	9746	HMM-OPT
11000011011011011	3	0.843	3	3	42	19124	
11011011001011001	4	0.836	15	15	15	23212	
101101100001101101	5	0.835	42	18	37	13208	
11001011001011011	15	0.824	4	6	1	17945	M3-OPT
11011011011011	22	0.814	17	27	129	24187	WABA
111001001001010111	11258	0.585	10427	10350	3254	1	PH-OPT
1111111111	24270	0.451	24285	24233	24310	24310	BLAST
101010101010101011	24310	0.386	24310	24310	24298	24306	WORST

Table 4. Ranks of seeds under different probabilistic models. The table shows the rank of each of the chosen seeds in the testing data set as well as under each of the considered probabilistic models.

al.[5], we have given an efficient algorithm for computing the probability that a sequence generated by such a model matches a given seed; then, by examining all such seeds, one can find the best.

We have adapted our methods to the subject of detecting homologous coding sequences in DNA. Our HMM for local alignments of these regions models their 3-periodic nature, dependence among positions within triplets, and inhomogeneity in conservation. This HMM is much better at predicting the sensitivity of a given seed than more naive models. In particular, in our experiments, the 3 best seeds according to this model were, in fact, the best overall.

Also, these optimal seeds are substantially more sensitive to alignments than seeds used by existing sequence alignment methods, while alignments based on them require essentially no additional time. Our experiments show that they substantially outperform the naive BLAST seed, and the PatternHunter seed, developed for a different model. They also outperform the simple 3-periodic seed used in WABA, matching 20-29% of the alignments missed by that seed.

Other questions for future work remain open. First, is it possible to develop similar seed-finding algorithms for other probabilistic models more complicated than HMMs, while still keeping the reasonable runtime? Alternatively, can one approximate the probability of a seed match without the expensive dynamic programming we have described? Also, do other pattern-finding applications of this sort exist, where one searches for patterns in the output of HMMs, and if so, how readily can these fit within our framework? Finally, can still more accurate models for coding alignments be developed, or can similar techniques can be used to study patterns of conservations in sequence elements other than protein coding regions?

Acknowledgements. We would like to thank Bioinformatics Solutions, Inc. (`http://www.BioinformaticsSolutions.com/`) for providing us with a version of PatternHunter customized to our needs and Ming Li for providing access to the manuscript [5].

References

1. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
2. A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45–48, 2000.
3. J. Buhler, U. Keich, and Y. Sun. Designing seeds for similarity search in genomic dna. In *Proceedings of the 7th Annual International Conference on Computational Biology (RECOMB)*, 2003. To appear.
4. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis.* Cambridge University Press, 1998.
5. U. Keich, M. Li, B. Ma, and J. Tromp. On spaced seeds. Unpublished.
6. W. J. Kent and A. M. Zahler. Conservation, regulation, synteny, and introns in a large-scale C. briggsae-C. elegans genomic alignment. *Genome Research*, 10(8):1115–1125, 2000.
7. I. Korf, P. Flicek, D. Duan, and M. R. Brent. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17 Suppl 1:S140–8, 2001.
8. B. Ma, J. Tromp, and M. Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, March 2002.
9. L. R. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.
10. Z. Yang. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution*, 10(6):1396–1401, 1993.