Optimal Spaced Seeds for Finding Homologous Coding Regions

Broňa Brejová and Daniel G. Brown
School of Computer Science, University of Waterloo
Waterloo ON N2L 3G1 Canada
{bbrejova, browndg}@math.uwaterloo.ca

September 28, 2002

Abstract

We study the problem of computing optimal spaced seeds for identifying homologous coding DNA sequences in large genomic data sets. We develop two models of DNA sequence alignment in coding regions, and using data sets from human/*Drosophila* and human/mouse comparisons, we compute optimal spaced seeds using a dynamic programming algorithm. The seeds we identify are more sensitive by far at identifying homologous regions than the seeds from BLAST or from PatternHunter, and also significantly improve on the sensitivity of WABA, which also uses a simple spaced seed. In particular, in human/*Drosophila* comparisons, we offer an 82% improvement in false negatives over BLAST and a 33% improvement over WABA. 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.

1 Introduction

Large scale similarity search is typically the first step of any study in comparative genomics. Regions with statistically significant sequence similarity are conjectured to be homologous. That is, they are conjectured to share evolutionary origin and to have been preserved by evolution. Comparative studies of homologous sequences can elucidate evolutionary history, function and structure of biological sequences. Efficient, sensitive tools for genome-scale pairwise local alignment are thus essential components of a wide range of bioinformatics tasks.

In particular, pairwise alignments of homologous coding regions of DNA sequences can aid gene finding [8], as they can help identify conserved exonic regions that otherwise would be missed by a gene finder without this information. Yet, as we will show, current techniques often miss homologous regions as they may lack a long enough region with no DNA mismatches.

Large scale similarity search is typically done by BLAST [1] or other similar programs. BLAST first finds short exact matches, called hits. A BLAST hit consists of several consecutive positions (the default is 11). For each, an alignment is built that extends the hit on both sides. If the alignment score exceeds a threshold, the alignment is reported. Some significant alignments do not contain 11 consecutive matches; thus, they are not discovered by BLAST.

Here, following ideas introduced in Ma et al. [6], we consider the use of spaced seeds, which generalize BLAST hits. A generalized hit consists of several non-consecutive matches in a prescribed configuration, called a seed. A seed can be represented as a binary sequence, in which a `1` denotes a position that is required to match and a `0` denotes a position that is not required to match. For example, the seed `110111` requires two consecutive matches followed by one position which may or may not match, and another 3 consecutive matches. The seed corresponding to a BLAST hit is simply `11111111111`.

In a model of similarity regions in which every position matches independently with constant probability, a randomly generated similarity region has higher probability of containing a hit of a well-chosen spaced seed with 11 ones than a hit of the trivial BLAST seed, even though the spaced seed is longer [6]. However, the expected number of hits occurring by chance depends only on the number of the ones in a seed and thus is the same. Ma et al. also report that their similarity search software, PatternHunter, is more sensitive than BLAST in an experiment with real data. In related work, Keich et al. [4] develop a dynamic programming algorithm which computes the optimal seed for regions of a given length at a given level of sequence similarity.

PatternHunter uses a seed optimized for sensitivity in a very simple probabilistic model of homologous sequences. However, this model is very general and does not reflect specific conservation patterns occurring in different types of homologous genomic sequences. In particular, protein coding regions make up most of the significant local alignments between two distant species, and are also specifically important. Thus, local alignment tools must perform well on these sequences.

Here, we extend the seed optimization idea to models better encapsulating sequence conservation in coding regions, and we find optimal seeds under such models. We compare the performance of these seed to other seeds. The main characteristic of coding sequences of which we take advantage is that their conservation patterns differ at different positions in codons. In particular, the third position in codons can often mutate silently, without altering the amino acid. Similarly, the first position often can mutate with no effect on the amino acid, or changing it to a similar amino acid. Hence, the second position is most likely to be conserved, followed by the first.

We have identified optimal seeds for both of the models we create, using a data set of homologous regions in human and mouse and in human and *Drosophila*. Using our methods, we discover seeds that reduce false negatives by 33% as compared to the seed used in sequence alignment program WABA [5], 82% as compared to BLAST, and 70% as compared to the seed optimized in PatternHunter. The runtime of PatternHunter with one of our seeds is within 10 % of its runtime when using the standard PatternHunter seed on our test set.

Our approach is similar to that of Kent and Zahler [5], whose alignment program WABA uses the seed `110110110`. This seed requires matches on the first two positions within a codon, but not on the last, called the "wobble position." Kent and Zahler report a substantial improvement over BLAST in sensitivity in detecting homologous coding sequences. However, in contrast to WABA, we allow seeds which are not 3-periodic, which allows a higher probability of hits occurring in regions that do not have high conservation, and our optimized seeds have better theoretical and practical performance as a result.

The rest of the paper is organized as follows. Section 2 introduces probabilistic models of homologous coding regions. We modify the algorithm from Keich et al. [4] to search for optimal seed in these probabilistic models. Section 3 describes the data and procedures used to estimate the parameters of the models from biological sequences and, and shows our optimal seeds. In this section, we also characterize alignments in a more precise way that focuses on their most highly conserved segments, and compute seeds our test set, characterized in this way. Section 4, gives the results of our experiments, and Section 5 contains concluding remarks.

2 Models and seeds for local alignments of coding regions

PatternHunter's seed is optimal in that it has a higher probability of having a hit in a local alignment chosen from a specific probability distribution than any other seed with the same length and the same number of 1's. Here, we describe the probabilistic model used in PatternHunter and propose two extensions better suited for modeling homologous coding regions. We then show how to modify the dynamic programming algorithm of Keich et al. [4] to find optimal seeds for this model.

2.1 Probabilistic models of local alignments

To give probabilistic models of local alignments, we represent an alignment as a binary sequence, where `1` represents a match and `0` a mismatch. We model only ungapped alignments, as hits are found only in gapless regions.

PatternHunter's probabilistic 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 $p = 0.7$.

We introduce two generalizations of the model $PH(N, p)$ with more parameters that better capture the codon-based conservation structure of local alignments in protein coding regions.

First, the 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 position within codon, but the positions of the alignment are still independent. Formally, it is a sequence of N independent Bernoulli random variables $X_0, X_1, \ldots, X_{N-1}$ where $\Pr(X_i = 1) = p_{i \bmod 3}$.

Model $M^{(8)}(n, p_{\mathtt{000}}, p_{\mathtt{001}}, \ldots, p_{\mathtt{111}})$ represents a region of n codons ($N = 3n$ nucleotides) in which 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 specified by the parameters $p_{\mathtt{000}}, \ldots, p_{\mathtt{111}}$, and individual codons are independent of each other. 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 $\Pr(X_{3i} = a, X_{3i+1} = b, X_{3i+2} = c) = p_{abc}$.

2.2 Finding optimal seeds

Given a particular probabilistic model, we can find the seed from a certain class of seeds C that maximizes the probability of a hit in a local alignment sampled from this model, by computing the probability of a hit for each seed from C and choosing the best one. The class of seeds C in our experiments contains 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.

Keich et al. [4] give a dynamic programming algorithm to compute the probability that a given seed Q of length M has at least one hit in a local alignment X sampled from $PH(N, p)$. Their algorithm is based on the following idea. Let $X[i]$ denote the prefix of alignment X of length i. Let $A_{i,x}$ be the probability that seed Q produces at least one hit in $X[i]$, provided that $X[i]$ ends with x. Here i is an integer between 0 and N, and x is a binary string of length at most M. The probability that Q has at least one hit in the entire alignment X is then equal to $A_{N,\lambda}$ (λ denotes the empty string). The probability $A_{i,x}$ can be computed by the following recurrent formula:

$$A_{i,x} = \begin{cases} 0 & \text{if } i < M & \text{case (A)} \\ 1 & \text{if } |x| = M \text{ and } \mathit{matches}(x, Q) & \text{case (B)} \\ A_{i-1,y} & \text{if not } \mathit{matches}(x, Q) & \\ & \quad \text{and } y \text{ is } x \text{ with its last character removed} & \text{case (C)} \\ p \cdot A_{i,\mathtt{1}x} + (1-p) \cdot A_{i,\mathtt{0}x} & \text{if } |x| < M \text{ and } \mathit{matches}(x, Q) & \text{case (D)} \end{cases}$$

In this formula, $\mathit{matches}(x, Q)$ is the event that string $\mathtt{1}^{M-|x|}x$ is a hit for seed Q. That is, when we align x with the end of Q, it has 1 on all positions where Q has 1. Case (A) recognizes that seeds of length M cannot have hits in shorter regions. In case (B), the presence of string x requires a hit. In case (C), since the string x does not match Q, a hit will not end on the last position of $X[i]$, so we need only consider previous positions. Finally, case (D) provides a formula for combining probabilities for alignments with the last $|x|+1$ characters fixed to a probability for alignment with only $|x|$ fixed characters. This recurrent formula can be used to compute probabilities $A_{i,x}$ in order of increasing i and shrinking x.

The algorithm can be made more efficient by storing and computing $A_{i,x}$ only for strings x such that $\mathit{matches}(x, Q)$ is true. For each length from 0 to M there are at most 2^{M-W} matching strings of this length, as only positions with zeroes in Q can vary. If we compute the probability $A_{i,x}$ for a matching x, we can use cases (A), (B) without change. Case (C) is not encountered. In case (D), however, we need the probability $A_{i,\mathtt{0}x}$ and string $\mathtt{0}x$ may be non-matching. In that case we apply case (C) repeatedly, until we obtain $A_{j,y}$ for some $j < i$ and y which is the longest matching prefix of $\mathtt{0}x$.

We have extended the previous algorithm so that it computes the probability of a hit for a seed Q under our generalized models $M^{(3)}$ and $M^{(8)}$. The only change required for model $M^{(3)}(N, p_0, p_1, p_2)$ is that in case (D) we replace p with $p_{(i-|x|-1) \bmod 3}$, which is the probability of a match on position directly preceding x in the alignment.

The modification for model $M^{(8)}(n, p_{\mathtt{000}}, \dots, p_{\mathtt{111}})$ is less straightforward, because the probability of a match on a given position depends on the other two positions within a codon. Therefore we only consider strings x whose length is a multiple of three. Case (D) is replaced with

$$A_{i,x} = \sum_{v \in \{\mathtt{0},\mathtt{1}\}^3} p_v A_{i,vx} \quad \text{if } |x| < M \text{ and } \mathit{matches}(x, Q) \qquad \text{case (D')}$$

The formula in case (D') works correctly if i is a multiple of three, so x is aligned properly with codon boundaries. For other values of i, we change the definition of $A_{i,x}$ as follows: $A_{i,x}$ is the probability that $X[i]$ contains a hit of seed Q provided that $X[3\lceil i/3 \rceil]$ ends with x.

We can thus compute $A_{N,\lambda}$ for each possible seed, and choose the seed whose probability of a hit is highest; this is the best seed for the model.

We have implemented the dynamic programming algorithm of Keich et al., and the modifications for the models $M^{(3)}$ and $M^{(8)}$ explained above, in the C programming language. The implementation is tuned to speed the computation as much as possible, by a variety of bit-based computation techniques. For each matching string x, the program precomputes the longest matching prefix y of $\mathtt{0}x$. This process takes $O(M^3 2^{M-W})$ time. Then, for each i, and each matching string x, the success probability $A_{i,x}$ can be found using only a constant amount of computation, if numbers of size between 0 and 2^{M-W} can be manipulated in constant time. (If not, there is a slowdown of a factor of $M - W$.) There are $NM2^{M-W}$ values that need to be computed, so the runtime is $O(2^{M-W} M(M^2 + N))$ for each of the $\binom{M-2}{W-2}$ possible seeds.

This algorithm, on its way to computing the probability of a match of seed Q on a region of length N, computes the probability of that seed matching regions of all lower lengths. We will use this fact to slightly extend our probabilistic models so that the length of the alignment is also a random variable, chosen from some fixed distribution of alignment lengths. Probability of a seed match in such extended model can be computed as a weighted average of probabilities for models with fixed lengths. In all of our experiments, we used these weighted averages in computing the optimal seeds.

3 Experimental data

Here, we describe our experimental data, and how we have estimated the parameters of the models before computing the optimal seeds for those sets of parameters. We focused on alignments between the human and fruitfly (*Drosophila melanogaster*) genomes, and between human and mouse. We limited our experiments to well-annotated proteins in all of these species, especially those for which it was straightforward to find DNA segments containing the genes. We aligned these sequences with BLASTP and extracted high-scoring pairs as an initial data set for our experiments. After a pre-processing step, described below, we had a data set of 19976 human-mouse RNA alignments and 1318 DNA alignments; for human-fly, we had 4304 RNA alignments and 507 DNA alignments. With our data sets, we estimated the parameters for the two models, $M^{(3)}$ and $M^{(8)}$ described above, and computed the best seeds to find hits in these alignments under those parameters.

```
           Protein alignment:
 M   E   K   T   E   L   I   Q   K   A
 +   +   K       E   L   +   Q   K   A
 V   D   K   -   E   L   V   Q   K   A

            mRNA alignment:
ATG GAG AAG ACT GAG  CTG ATC CAG AAG GCC
GTC GAT AAG --- GAG  CTG GTC CAG AAG GCT
```

Figure 1: An mRNA alignment corresponding to a protein alignment.

Many of the alignments we were using were not well characterized by any of the models, because the conservation was not uniform across the alignments. In particular, most alignments varied between high-identity regions and low-identity regions. (This is unsurprising, as the highly conserved regions are more likely to be functional parts of the proteins, for example [7]). Based on this observation, we identified the region in each alignment that is most likely to be aligned by a BLAST seed, again estimated the parameters of the models, and again computed the seed with the highest probability of finding matches in those regions. The underlying assumption is that if an alignment contains a hit for a given seed, the probability is high that there is a hit inside this highest-probability region.

3.1 Details of the data set

To produce our initial alignments, we applied BLASTP 2.0.8 [1] to a set of 7264 human, 1501 drosophila, and 4610 mouse protein sequences from SWISS-PROT [2] release 40, October 2001. We used the E-value threshold 1e-30 in these alignments, with the non-human sequences as the database. From these protein alignments, we then mapped as many as possible onto the corresponding mRNA sequences, which we extracted from GenBank (see Figure 1). In this way we have obtained an alignment of the two RNA sequences. These RNA alignments were matched to the genomic sequences encoding the proteins, though we limited our search to Genbank entries for genomic regions which were annotated to exactly match the SWISS-PROT protein sequence. Of the 36854 initial human-mouse protein alignments, we found proper mRNA alignments in 19976 of them, and DNA alignments in 1318 of them. (For human-drosophila, there are 4304 mRNA alignments and 507 DNA alignments.) We called the set of RNA alignments in mouse R_m, and in drosophila R_d, while the DNA alignment sets are D_m and D_d, respectively.

As the set of aligned regions is fairly small, we have chosen not to divide it into testing and training sets for our experiments. However, as we will be modeling each alignment using the probabilistic models, overfitting may be less of a concern in our experiments than it would be if we were training to the exact characteristics of the data sets. We have also discovered that optimal seeds are not very sensitive to small changes of parameters, so this also assuages this fear.

Table 1 summarizes statistical properties of the genes in our data sets, and also compares the human data to data from the entire human genome [3]. We see that our sample is biased towards genes with fewer, and longer, exons, as one would expect due to the bias toward more well-analyzed, shorter sequences.

	Human		Drosophila		Mouse		Human genome
	median	$n =$	median	$n =$	median	$n =$	median
exons per gene	4	1026	3	821	3	492	7
exon length	131	5732	212	2918	136	2185	122
mRNA length	1.3kb	4285	1.4kb	1026	1.3kb	3138	1.1 kb
gene length	3.3kb	1026	1.6kb	821	2.0kb	492	14 kb

Table 1: Statistical properties of genomic features of individual sets. The row labeled "mRNA length" gives the length of the coding sequence in nucleotides. Data for the human genome are taken from [3].

Data set	Model	p_{000}	p_{001}	p_{010}	p_{011}	p_{100}	p_{101}	p_{110}	p_{111}
R_m	$M^{(8)}$	0.1426	0.0573	0.1236	0.0660	0.0710	0.0364	0.2335	0.2696
R_m	$M^{(3)}$	0.0683	0.0514	0.1540	0.1158	0.1071	0.0805	0.2413	0.1815
R_d	$M^{(8)}$	0.1792	0.0668	0.1497	0.0660	0.0801	0.0337	0.2483	0.1762
R_d	$M^{(3)}$	0.1092	0.0569	0.1943	0.1013	0.1273	0.0664	0.2265	0.1181

Table 2: Parameters for model $M^{(8)}$ estimated from sets R_m and R_d. They are compared to probabilities of all triples obtained in model $M^{(3)}$ by multiplying probabilities at individual codon positions.

3.2 Properties of alignments

We estimated the parameters of the $M^{(3)}$ and $M^{(8)}$ models from the RNA alignments in sets R_m and R_d. For each set we extracted all aligned codon pairs, discarding the pairs containing gaps. Then, each pair of codons was transformed to a triple of bits where `1` denoted a match and `0` a mismatch. We estimated the parameters of model $M^{(3)}$ as the relative frequencies of a match at each position in the triple, and the parameters of model $M^{(8)}$ as the relative frequencies of individual triples.

For model $M^{(3)}$, the parameters for data set R_m are $p_0 = 0.61, p_1 = 0.69, p_2 = 0.43$, and for data set R_d are $p_0 = 0.54, p_1 = 0.64, p_2 = 0.34$. As we would expect, the conservation is lowest in the third, "wobble" position, and highest in the second position. Also, as expected, the overall conservation is higher for human/mouse alignments than for human/drosophila alignments. Table 2 shows the parameters for model $M^{(8)}$, for both data sets. We also compare the amount of dependency between positions by comparing the probabilities of all triples in $M^{(8)}$ and in $M^{(3)}$ estimated on the same data. The two probability distributions differ substantially for some triples but the general trends seem to be the same. Hence, while model $M^{(8)}$ is more precise than model $M^{(3)}$, this added precision may not be needed, especially since it has four more parameters to estimate.

After mapping the RNA alignments onto the DNA sequences, we have divided the alignments into fragments at exon boundaries, with each fragment completely inside one exon in both species. Gaps at fragment ends were deleted. This smaller alignment fragments are regions that can be potentially found by a nucleotide similarity search tool when it is applied on the genomic sequences (assuming that the alignment tool cannot produce gaps of the length of the intron).

These fragments may themselves contain gaps, and we have divided the fragments accordingly, based on their gaps, into smaller, ungapped fragments. The entire region that makes up a hit must be inside an ungapped fragment, while the subsequent alignment algorithm may join the regions of

	D_d			D_m		
	gapped	ungapped	cores	gapped	ungapped	cores
Number of fragments	3615	6058	5105	4979	9637	8366
Mean length	144	82	30	230	115	38
Median length	106	58	21	138	73	21
Fragments shorter than 20	7%	18%	38%	10%	15%	39%

Table 3: Properties of the length distribution of gapped and ungapped fragments on data sets D_d and D_m, and on the core regions of alignment.

a gapped fragment.

Various parameters of the length distribution of alignment fragments on both genomic sets D_d and D_m are shown in Table 3. Typical gapped fragments have approximately two ungapped fragments. Because of differences in exon structure between species, the set D_d contains some very short fragments, with length less than 20, which are unlikely to be built into alignments.

3.3 More detailed consideration of the alignments

Finally, we examined the alignments and noted that their pattern of conservation was highly non-uniform. In particular, many alignments include short, highly conserved regions and many less well conserved regions. Our probabilistic models give a much lower probability of finding a hit in one of these regions than is correct. To address this problem, we have considered reducing each alignment to its most highly conserved region and used these regions in picking seeds.

For each alignment, we have computed the region within the alignment, at least 18 bases long, for which, if it were derived by the uniform-probability model, it would be most likely to contain a BLAST hit. (That is, if a given region is 30 bases long with 25 matches, we compute the probability of a match to the BLAST seed for the model $PH(30, 25/30)$.) Some properties of the length distribution of these regions is shown in Table 3; note that the number of cores is smaller than the number of ungapped fragments since some ungapped fragments have length less than 18. It is worth noting that most of these cores do turn out to be quite small, as their median size is 21. In computing our seeds in the next section, we have computed seeds in the model where we have assumed that all hits are required to occur in these maximum-probability regions, as well as seeds for the parameters obtained from the whole fragments.

3.4 Optimal seeds and predicted performance

Using the model parameters from Section 3.2, we have used the dynamic programming algorithm to find optimal seeds. We computed the optimal seeds for weight 10 and length at most 18 and for weight 11 with length at most 18 and at most 20. We also used the algorithms for both models $M^{(3)}$ and $M^{(8)}$, for the drosophila and mouse data sets. We also did all of these experiments for the core alignment regions identified in Subsection 3.3. This gives a total of $2 \times 2 \times 2 = 8$ tests for each of the three weight/length combinations. In our 24 tests, many of the seeds were optimal on more than one experiment; in addition, two tests with seeds of weight 10 gave the WABA seed `11011011011011` as the optimal one. We have chosen five of the seeds for further consideration, two for weight 10 and three of weight 11; they are listed in Table 4, with the PatternHunter, BLAST and WABA seeds of these weights, along with identifiers we used for these seeds in our experiments.

Table 5 shows the theoretical performance of these eleven seeds on homology regions whose conservation patterns are derived from the models PH, $M^{(3)}$ and $M^{(8)}$ and whose lengths are from

Name	Seed	Experiments where optimal
M8-10-17	`110 110 110 000 110 11`	$M^{(8)}$ weight 10, length ≤ 18, full drosophila and mouse regions
M3-10-14	`111 010 110 110 11`	$M^{(3)}$ weight 10, length ≤ 18, full mouse regions
PH-10	`111 001 001 001 010 111`	PatternHunter model, weight 10
WABA-10	`110 110 110 110 11`	WABA and $M^{(3)}$ weight 10, length ≤ 18, core regions, mouse and drosophila
BLAST-10	`111 111 111 1`	BLAST
M8-11-17	`110 110 110 010 110 11`	$M^{(8)}$ weight 11, length ≤ 18, full and core drosophila regions
M8-11-14	`111 110 110 110 11`	$M^{(8)}$ weight 11, length ≤ 18, core mouse and drosophila regions and M^3, core mouse regions
M8-11-20	`110 110 010 110 000 110 11`	$M^{(8)}$, weight 11, length ≤ 20, full mouse and drosophila regions
PH-11	`111 010 010 100 110 111`	PatternHunter model, weight 11
WABA-11	`110 110 110 110 110 1`	WABA
BLAST-11	`111 111 111 11`	BLAST

Table 4: Seeds considered for further investigation. The experiments for which a seed was optimal are indicated.

the same distribution as the real aligned fragments. We see that in all models, the advantage of the optimal seeds over the naive BLAST seeds are quite significant, and the improvement over the WABA seed is still nontrivial in some cases. For example, the weight 11 seed M8-11-20 has a 39% improvement over the weight 11 WABA seed in theoretical sensitivity by the $M^{(8)}$ model. It is also worth noting that the new seeds optimized for the coding sequence models do not perform much worse than the PatternHunter seed for the PH model, which suggests they should still do well at identifying non-coding homology regions.

The final column of Table 5 shows the result of looking for regions matching the seeds in the ungapped fragments in D_d. Recall that, in order to find a hit in a gapped fragment, there must be an ungapped fragment that includes a match for the seed. Also, as the distributions of length and identity were identified from these ungapped fragments, the expectation is that they should have a match to the seeds approximately as often as the columns in the table indicate. We see that the closest similarity is to the column corresponding to the core regions and model $M^{(8)}$, suggesting that this model best approximates the actual alignments. We also already see the likelihood that the optimized seeds may be preferable to the other seeds such as WABA, though the results are not, by any means certain. Note that for the core regions, the theoretical sensitivity of the WABA-10 seed is highest; this seed was optimal for model $M^{(3)}$ in this case.

4 Experiments

We examined the usefulness of our five test seeds and our 6 reference seeds at identifying the alignments in our data sets. We first examined the gapped fragments in our sets to see which contain ungapped fragments with hits for the seeds. We have also tested the usefulness of the

		full fragment		core region		actual
Seed name	PH	$M^{(3)}$	$M^{(8)}$	$M^{(3)}$	$M^{(8)}$	fragments
M8,10,17	4.6%	7.4%	16.2%	14.1%	19.4%	25%
M3,10,14	4.7%	5.7%	12.4%	14.6%	19.3%	22%
WABA,10	4.0%	6.3%	13.9%	15.2%	20.6%	24%
PH,10	4.7%	4.3%	6.5%	7.2%	8.1%	17%
BLAST,10	3.0%	2.3%	6.0%	7.9%	11.6%	14%
M8,11,20	2.3%	4.5%	10.6%	7.2%	10.0%	22%
M8,11,17	2.3%	4.4%	9.9%	10.7%	14.8%	21%
M8,11,14	2.3%	2.9%	8.1%	10.1%	14.9%	19%
WABA,11	2.0%	3.4%	7.6%	9.7%	13.4%	20%
PH,11	2.4%	2.3%	5.1%	6.1%	8.2%	15%
BLAST,11	1.5%	1.2%	3.9%	5.4%	8.9%	12%

Table 5: Predicted sensitivity of the seeds under different probabilistic models, for models with length distributions, homology fractions and parameters from drosophila data sets R_d and D_d. For each seed, the predicted probability of the seed matching a homology region in the model is shown. The third and fourth columns correspond to the core regions of highest homology identified using the techniques of Subsection 3.3. Finally, the fifth column represents the actual fraction of ungapped fragments from set D_d that match the seed.

seeds in practice, using a version of PatternHunter that the seed to be set for different runs. Our results show that our optimized seeds provide significantly improved performance in finding protein homology regions. Our experiments were performed on a 1.7GHz Pentium 4.

4.1 Fragment searches

We examined all gapped fragments in both D_m and D_d, to identify what fraction contained a hit for each of the 11 seeds. The results of these experiments are shown in the first and second columns of Table 6 for human/mouse and in the sixth and seventh for human/drosophila. We assign a PatternHunter score to each of the fragments, and we display result for fragments with score at least 16 separately, as it is least likely that a DNA homology search program will find the regions discovered initially by BLASTP, whose nucleotide conservation does not sufficient to achieve this score.

Here, we see that while all of the seeds except the BLAST seeds are quite sensitive, the best seeds for both weights and for both species are from our set of optimized seeds. In particular, the seed M8-10-17 has roughly half of the false negative rate of WABA-10 (4% versus 7%) on the high-scoring human mouse hits and two-thirds of the false negative rate (6% versus 9%) for human-drosophila. While this improvement may seem incremental, finding these missed regions of homology can be extremely helpful in predicting gene structure, or simply in properly identifying important regions of a coding sequence.

4.2 Actual alignments

We have also used the nucleotide alignment program PatternHunter to align the genomic regions in our test sets, using all eleven test seeds. We masked all non-exonic sequences in our experiments, to avoid complications with intronic matches. This may have made all seeds less likely to find

	Human/mouse					Human/drosophila				
Seed	Seed hit		PH hit		Exons	Seed hit		PH hit		Exons
	All	≥ 16	All	≥ 16		All	≥ 16	All	≥ 16	
M8-10-17	60%	96%	55%	95%	63%	38%	94%	32%	93%	69%
M3-10-14	59%	93%	53%	92%	65%	33%	90%	29%	88%	72%
WABA-10	58%	93%	54%	93%	63%	36%	91%	30%	90%	69%
PH-10	51%	90%	50%	90%	62%	25%	80%	22%	77%	67%
BLAST-10	49%	85%	46%	85%	59%	21%	67%	18%	64%	64%
M8-11-20	54%	92%	51%	91%	52%	33%	90%	28%	89%	57%
M8-11-17	55%	92%	49%	88%	56%	32%	88%	26%	85%	61%
M8-11-14	53%	90%	51%	90%	59%	29%	86%	25%	84%	64%
WABA-11	52%	89%	50%	88%	56%	29%	85%	25%	84%	61%
PH-11	48%	88%	47%	86%	55%	22%	75%	20%	70%	59%
BLAST-11	44%	81%	43%	81%	53%	18%	61%	16%	58%	55%
Sample	4987	2191	4987	2191	5732	3615	768	3615	768	5732

Table 6: Performance of individual seeds in experiments on human/mouse and human/drosophila. Columns are explained in the text; the results are shown for fragments with a PatternHunter score at least 16 and for all fragments, regardless of score.

matches, as splice sites are often well conserved, but we assume that all seeds are equally affected by this decision. The PatternHunter scoring system we used is: match = +1, mismatch = -1, gap open = -5, gap extend = -1, and we searched for matches with score was at least 16. The results of our experiments for human/mouse are in the third and fourth columns of Table 6 and in the eighth and ninth columns for human/drosophila. The first in each pair is the fraction of all fragments that overlap an alignmnet found by the PatternHunter, while the second is for those fragments that have score at least 16 exists.

Here, again, the optimized seeds perform better than the BLAST seeds or the seeds optimized for the PatternHunter model, and there continues to be an improvement over the results of the WABA seeds. While some of the regions which do have a possible alignment of score at least 16 are not detected, even when they do have a starting hit, these event seems equally common for all seeds, and the advantage of our new seeds continues to be prominent.

Figure 2 shows the sensitivity of the seeds as a function of fragment PatternHunter score for seeds WABA-10, BLAST-10 and M8-10-17. We see that M8-10-17 is preferred for essentially all scores, and is particularly useful for regions with moderate score, between +10 and +30, while BLAST in all cases is inferior.

4.3 Exon sensitivity

We finally considered the usefulness of our seeds at actually identifying exonic regions as being regions of nucleotide conservation. For each of the seeds, we computed what fraction of these 5732 human exons in our test sets overlapped at least one of the PatternHunter hits we found with that seed.

Here, we saw that the formerly best seeds suddenly fell off in sensitivity. For example, in the human/mouse comparisons, the seed M8-10-17 had a match in only 63% of these exons, as compared to 65% for the nominally less sensitive seed M3-10-14 or also 63% for the WABA-10 seed. The difference is easily accounted for, however. The shorter seed M3-10-14 is capable of matching 16-base-pair long regions of 100% identity, which have PatternHunter score of 16. However, if this is

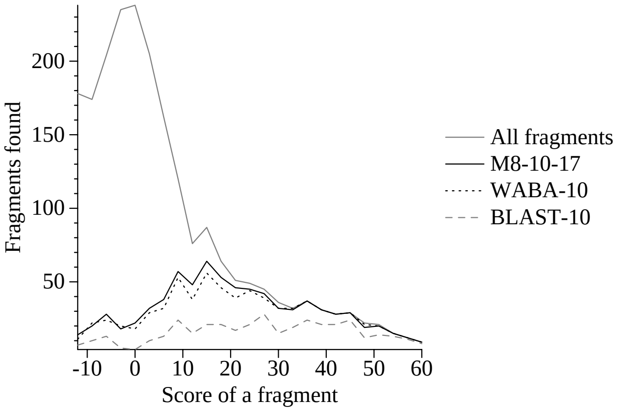

Figure 2: The sensitivity of seeds as a function of region score. The number of BLASTP fragments for each PatternHunter score are indicated, along with the sensitivity of three seeds at detecting them. The optimized seed M8-10-17 is consistently more effective than the other two, especially in the range of scores from 10 to 30.

the entire homology region, it clearly cannot match the 17 base-pair long seed M8-10-17. When we looked at the score 16 alignments, we found that they accounted for all of the difference; 160 exons with best score 16 only matched M8-10-17, while 270 of them only matched M3-10-14; this difference of 110 is the 2% of the 5732 human exons in our set that is the difference.

We also verified that the runtime of PatternHunter did not change much as a result of using our new seeds. In general, WABA seeds had slightly faster runtimes for comparisons, such as 32 s for human/mouse using WABA-10 versus 34 s for M8-10-17, possibly because WABA finds more seeds closer to each other and PatternHunter is well suited for not wasting runtime if this occurs.

In general, the results of our experiments are encouraging; they show that optimized seeds can be significantly more sensitive than naively chosen seeds at identifying regions of homology in exonic sequences, and particularly, may help at identifying medium-length (at least 17 base pairs) regions with low homology.

5 Conclusion

We have proposed two simple probabilistic models for sequence conservation in homologous coding regions and found optimal seeds under these models for real data sets. We have compared the performance of these seeds with previously used seeds, both in the probabilistic model and on data from real alignments. They outperform existing seeds in finding preliminary hits and in producing alignments, and they also require very little additional search time. Further experiments are needed to conclusively decide which seed is best. Comparisons should be extended to other species and bigger data sets.

Many questions still remain open. It is certainly possible to construct more complicated models of alignments, such as Markov chains, or profiles, in which the probability of a match changes with its distance from the center of the alignment. Our preliminary attempt to replace each alignment with its highest probability core suggests this may be a productive direction. It is also interesting whether similar techniques can be used to study patterns of conservations in sequence elements other than protein coding regions. Finally, it is interesting to ask whether there are techniques for approximating the probability of a match as a function of the model generating it without the exponential-time dynamic programming algorithm we have described.

6 Acknowledgement

We would like to thank Ming Li for providing access to the manuscript [4] and Bioinformatics Solutions, Inc. (`http://www.BioinformaticsSolutions.com/`) for providing us with a version of PatternHunter where we could change the seed.

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] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, February 2001.

[4] U. Keich, M. Li, B. Ma, and J. Tromp. On spaced seeds. Unpublished.

[5] 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.

[6] Bin Ma, John Tromp, and Ming Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, March 2002.

[7] 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.

[8] R. F. Yeh, L. P. Lim, and C. B. Burge. Computational inference of homologous gene structures in the human genome. *Genome Research*, 11(5):803–806, 2001.