JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 12, Number 9, 2005

Pp. 1137–1152

Quick, Practical Selection of Effective Seeds for Homology Search

FRANCO P. PREPARATA,[1] LOUXIN ZHANG,[2] and KWOK PUI CHOI[3]

ABSTRACT

It has been observed that in homology search gapped seeds have better sensitivity than ungapped ones for the same cost (weight). In this paper, we propose a probability leakage model (a dissipative Markov system) to elucidate the mechanism that confers power to spaced seeds. Based on this model, we identify desirable features of gapped search seeds and formulate an extremely efficient procedure for seed design: it samples from the set of spaced seed exhibiting those features, evaluates their sensitivity, and then selects the best. The sensitivity of the constructed seeds is negligibly less than that of the corresponding known optimal seeds. While the challenging mathematical question of characterizing optimal search seeds remains open, we believe that our eminently efficient and effective approach represents a satisfactory solution from a practitioner's viewpoint.

Key words: homology search, sequence alignment, filtration technique, q-gram, spaced seeds, leakage model.

1. INTRODUCTION

THE NUMBER OF COMPLETELY SEQUENCED GENOMES increases at a phenomenal rate. This presents unprecedented opportunities for comparative genomics. By comparing orthologous genomic sequences, one can infer information on SNPs, translocation, tandem and segmental duplications, and intronic and intergenic regions with potential biological functions (Hardison *et al.*, 1997; Li, 2001). A usual starting point of this kind of study is to run homology search programs to detect similarities between segments of different genomic sequences or approximate repeats within a given sequence.

In the past twenty years, an extensive repertoire of homology search tools, such as FASTA (Lipman and Pearson, 1985), BLAST (Altschul *et al.*, 1990, 1997; Zhang, 2000), FLASH (Califano and Rigoutsos, 1995), MUMmer (Delcher *et al.*, 1999), QUASAR (Burkhart *et al.*, 1999), BLAT (Kent, 2002), PatternHunter (Ma *et al.*, 2002), and BLASTZ (Schwartz *et al.*, 2003), have been developed to meet the challenge to align biological sequences for different purposes in a fast and yet sensitive manner. These tools mainly belong to two categories: the "seed" approach as in the BLAST family, and the "suffix tree" approach as in MUMmer. We shall confine our attention to the seed approach. The seed approach is based on the principle of *filtration*, where consensus on a homology similarity is reached by first identifying short

[1]Computer Science Department, Brown University, Providence, RI 02912-1910.
[2]Department of Mathematics, National University of Singapore, Singapore 117543.
[3]Department of Statistics and Applied Probability, National University of Singapore, Singapore 117546.

identical segments (perfect matches), called *hits*, and then extending them on either side for approximate matches, called *alignments*. The resulting alignments are scored for acceptance. Independently, the filtration principle has also been applied to the approximate string matching problem (Karp and Rabin, 1987; Pevzner and Waterman, 1995; Burkhardt and Kärkkäinen, 2001).

The process can be modeled as follows. All sequences considered are constructed on the same alphabet, typically either nucleotides or amino acids, and are assumed to be generated by some Markov models. In this paper, we restrict ourselves to the zero-th order Markov model; that is, we assume that the sequence symbols are independently and identically generated (i.i.d.). Let S_1 and S_2 denote two aligned homologous sequences of length n, which may disagree only through substitutions. Positions of the two sequences are numbered $1, 2, \ldots, n$ from the left. In the standard approach, we define that there is a *hit between S_1 and S_2* occurring at position i (where $1 \leq i \leq n - w + 1$) if, for some chosen integer w, the two sequences are identical at positions $i, i + 1, \ldots, i + w - 1$. Using 1's and 0's to represent matches and mismatches between S_1 and S_2, respectively, such a hit can also be viewed as the detection of a substring of 1's, of length w (also denoted a q-gram), in a binary sequence S of length n. Binary sequence S is sometimes referred as a *similarity sequence*. The hit considered above could be viewed as due to a (simplest) *search pattern* $\pi = 1^w$ (w-mer). The *sensitivity* of a search pattern π is defined as the probability $p_\pi(n, p)$ of a hit with π over the ensemble of Bernoulli sequences of length n generated with probability p, where p indicates the level of similarity of the original two sequences. We will adopt this model throughout for discussion on the effectiveness of the seed approach.

BLASTN (Altschul *et al.*, 1990) uses a w-mer contiguous (or ungapped) seed. It was observed recently that *spaced* (or gapped) seeds are generally more sensitive than the corresponding contiguous seeds (Ma *et al.*, 2002). A spaced seed of length L and weight w is a sequence $1(0 \vee 1)^{L-2}1$ with exactly w 1's. The novelty of PatternHunter (Ma *et al.*, 2002) lies in the adoption of an "optimal" spaced seed for their search pattern. This innovation has since been adapted and applied to homology search for different purposes: DNA alignment (Ma *et al.*, 2002; Noè and Kucherov, 2003; Schwartz, 2003; Wheeler, 2003), coding regions (Brejovà *et al.*, 2004), and protein–protein, translated protein–DNA and translated DNA–DNA (Kisman *et al.*, 2005).

The fact that spaced seeds generally outperform the contiguous seeds of the same weight is informally explained in terms of relaxing the correlations existing among contiguous samplings. However, only partial progress has been made in the formal analysis of this behavior (Buhler *et al.*, 2003; Choi and Zhang, 2004; Gotea *et al.*, 2003; Keich *et al.*, 2004). It is known now that (1) for a given weight, contiguous seeds have higher sensitivity than uniformly spaced seeds $(10^k)^m1$, where $k, m \geq 1$ (Choi and Zhang, 2004; Keich *et al.*, 2004), and (2) nonuniformly spaced seeds are very likely to outperform contiguous seeds, but not much insight has been developed on the relative power of nonuniformly spaced seeds. In particular, Buhler *et al.* (2003) also propose a more refined structural modeling of the problem, which is the basis of an interesting asymptotic analysis in terms of the length n of the similarity sequence. Although very enlightening from a structural viewpoint, the asymptotic analysis may be of reduced practical significance, for it is a foregone conclusion that as the length n grows, the probability of detection approaches 1 so that an asymptotic analysis would compare such seeds in situations in which they are all equally effective (in fact, the gain in sensitivity of spaced seeds over contiguous seeds diminishes as n grows). On the contrary, situations which are of interest are those with relatively small n, the range of which varies with the parameter w. In these situations, the probability of a hit is well below 1 (typically, in the middle range), and an improvement of the order of, say, 50% or higher is of great practical significance.

The most interesting question is therefore the identification of the most sensitive seed(s) for a given weight w, level of similarity p, and moderate n (Keich *et al.*, 2004; Choi and Zhang, 2004; Kucherov *et al.*, 2004). A direct line of approach is through exhaustive search after the sensitivity of each spaced seed is computed. The sensitivity of a spaced seed can be computed by dynamic programming (Keich *et al.*, 2004) or a nested recurrence relation (Choi and Zhang, 2004). This direct approach, however, soon becomes impractical if not impossible, due to the number of spaced seeds growing at least exponentially in $L - w$. A first attempt to reduce the search set is the heuristic algorithm proposed by Choi *et al.* (2004) which uses the hitting probabilities of spaced seeds at $2L$ as an indication of their effectiveness.

While the characterization of optimal seeds for different values of the parameters w (the cost) and n (the target at hand) remains an intriguing and mathematically very challenging problem, its relevance to the practice of homology search is very much in doubt. In fact, some numerical experience with exhaustive

search reveals (see Section 6 for some examples) that, for fixed L, w, p, and n, almost all sensitivity values occur in the upper third of their range with a mode at about 80%. This suggests that any random seed far outperforms the corresponding contiguous seed. The reality of homology search also suggests that near-optimality, rather than absolute optimality, is what a practitioner may desire, if the corresponding seed is easily designed, considering that the similarity level is an undetermined factor.

It must be emphasized that the intent of this paper is not the validation of gapped-seed approaches to homology searches; some attention has already been devoted to this issue (for examples, Brejovà *et al.* [2004], Buhler *et al.* [2003], Choi *et al.* [2004], and Ma *et al.* [2002]) and more is needed. Our focus is on the issue of the selection of suitable gapped-seeds, and our main conclusion is that the search for "optimal" seeds is an elusive problem of formidable difficulty whose results can be almost equalled by direct design of near-optimal seeds.

Therefore, the objective of this paper is precisely the systematic, rapid design of seeds whose performance is expected to come remarkably close to that of the optimal seed(s). Towards this end, we shall elucidate an intuitively attractive mechanism (the *leakage model*), which underpins a spaced seed's sensitivity, and on this basis develop a prescription for effective seed design embodied by a very efficient procedure. The paper is organized in seven sections. Section 2 provides the needed definitions and notations, and Section 3 some general analysis. Sections 4 and 5 are the main portions of the paper: Section 4 presents the leakage model, which is a kind of dissipative Markov system, elucidating relationships between features of the seed sequence and the total probabilities of hitting sequences of varying lengths. Such an analysis naturally leads to a set of criteria for the selection of effective spaced seeds, presented in Section 5. In this section, the criteria are spelled out explicitly. In Section 6, we report a number of numerical results: first histograms of sensitivities for given L, w, p, and n, next some illustrations of the effects of various features of the seeds, and finally a comparison of our selected seeds with currently known optimal seeds found by exhaustive search. We conclude the paper in Section 7.

Finally, we note that some homology search tools, such as MegaBLAST (Zhang *et al.*, 2000) and SSAHA (Ning *et al.*, 2001), which are specially designed for locating large but highly similar regions, employ contiguous seeds with large weight. Recently, NCBI offered a version of MegaBLAST called Discontiguous MegaBLAST which uses the spaced seed idea as the nucleus for its alignments. At present, it provides only a limited choice of spaced seeds of weight 12. Therefore, it would be of interest to compare the sensitivity of our selected seeds with that of the contiguous seed with the same weight (see Table 3).

2. DEFINITIONS

Definition 1. *A search pattern or seed* $\pi = \pi_0\pi_1 \dots \pi_{L-1}$ *is a binary string with* $\pi_0 = \pi_{L-1} = 1$. *The weight* $w \leq L$ *is the number of 1's in* π.

The probability of the memoryless Bernoulli generator of the similarity sequence is denoted by p with $q = 1 - p$. If u and v are strings, then uv is their concatenation, $|u|$ is the length of u, and string $\overline{u}$ is the reverse of u. For a string $u = u_1 \dots u_n$, its j-prefix, denoted $u_{(j)}$, is the string $u_1 \dots u_j$; analogously, its j-suffix $u^{(j)}$, is the string $u_{n-j+1} \dots u_n$.

Definition 2. *Given a binary sequence* $v = v_1 \dots v_n$ *and seed* π *with* $n \geq |\pi|$, *a "hit" occurs at position* j *of* v *if* $v_j = 1$, $v_{j+L-1} = 1$ *and, for* $s = 1, \dots, L-2$, *if* $\pi_s = 1$ *then* $v_{j+s} = 1$.

Let b_j denote the probability of a hit at position j *for the first time from the left end*, i.e., a hit occurs at j, but no hit occurs at $1 \leq i < j$. The probability that a hit occurs *within* positions $1, \dots, j$ is denoted B_j. Clearly,

$$B_j = \sum_{i=1}^{j} b_i, \qquad 1 \leq j \leq n - L + 1.$$

In conformity with the definition of *hit* we have the following definition.

Definition 3. *The hitting set $\mathcal{F}(\pi)$ (abbreviated to $\mathcal{F}$) of seed π consists of the 2^{L-w} length-L binary strings obtained by assigning all possible (binary) values to the positions in the set $Z = \{j|\pi_j = 0\}$.*

For example, for seed $\pi = 101011$ we have $\mathcal{F} = \{101011, 101111, 111011, 111111\}$.

With the notion of hitting set, a hit at j means that the length-L substring starting at j belongs to the hitting set.

3. ANALYSIS

We now wish to construct the set $\mathcal{H}_j$ of the sequences of length $j + L - 1$ corresponding to a first hit at position j; i.e., for any sequence $v \in \mathcal{H}_j$ the L-suffix of v belongs to $\mathcal{F}$ and no other length-L substring of v belongs to $\mathcal{F}$. By definition, b_j is the total probability of set $\mathcal{H}_j$.

It is easy to realize that each sequence $v \in \mathcal{H}_j$ can be associated with a path in the standard shift-register (DeBruijn) diagram of order $L-1$. The states of the corresponding finite-state automaton are all the binary $(L-1)$-tuples, and a state u has transitions to states $u^{(L-2)}0$ (called a 0-transition) and $u^{(L-2)}1$ (called a 1-transition). Probabilities p and q are respectively assigned to 1- and 0-transition arcs of the DeBruijn diagram.

The path of a sequence $v \in \mathcal{H}_j$ starts from an arbitrary node (state) of the diagram (this state defines the $(L-1)$-prefix of v), consists of $j-1$ arcs avoiding all 1-transitions out of states u such that $u1 \in \mathcal{F}$ (otherwise a hit would occur for $i < j$), and terminates precisely in one of these states (which it leaves on the otherwise forbidden 1-transition). Such a sequence v results from concatenating to its $(L-1)$-prefix the labels of the traversed arcs and is completed with an additional symbol 1 (the final "forbidden" transition). The probability of sequence v is the product of the probabilities of its (statistically independent) symbols.

For example, for $\pi = 1101$, sequence $v = 011011101 \in \mathcal{H}_6$ corresponds to the sequence of visited states 011, 110, 101, 011, 111, 110, and a 1-transition from 110 completes v.

The analysis of the sequences of $\mathcal{H}_j$ appears very cumbersome. However, a crucial simplification is obtained if, rather than $v \in \mathcal{H}_j$, we consider its *reverse sequence*. Specifically, letting $v = v'1$, we consider in the same DeBruijn diagram of order $(L-1)$ the path of $\overline{v'}$ (the reverse of v'), which starts from a node $\overline{u}$ such that $u1 \in \mathcal{F}$ consists of $j-1$ arcs and terminates in an arbitrary state. We single out the set of starting states as follows:

Definition 4. *The set of states $\{\overline{u} : u1 \in \mathcal{F}\}$ is called the* initial set *$\mathcal{I}$ for pattern π.*

For example, for seed $\pi = 101011$, we have $\mathcal{I} = \{10101, 10111, 11101, 11111\}$.

The stated convention (analysis of reverse sequences) is adopted for the rest of this paper, and, for a correct understanding of the arguments, it is important that it be fully acknowledged by the reader. The intuitive reason behind the simplification is that each sequence $v' \in \mathcal{H}_{j+1}$ is obtained as a one-symbol extension of some $v \in \mathcal{H}_j$.

We denote $\mathcal{DB}(\pi)$ the order-$(L-1)$ DeBruijn diagram from which we have pruned all 1-transitions out of states u such that $1\overline{u} \in \mathcal{F}$. Again, we single out this set of states in the following:

Definition 5. *The set of states $\{u : 1\overline{u} \in \mathcal{F}\}$ is called the* reflecting set *$\mathcal{R}$ for pattern π.*

For example, for seed $\pi = 101011$, we have $\mathcal{R} = \{11010, 11011, 11110, 11111\}$. The reason for the chosen denotation ("reflecting") will be soon apparent.

Any path in this diagram describes a sequence whose reversal does not contain any string in the hitting set $\mathcal{F}$.

Diagram $\mathcal{DB}(\pi)$ is described by a $2^{L-1} \times 2^{L-1}$ transition matrix $M(\pi)$, which is substochastic, with entries 0, p and q where $M_{j,i}$ gives the transition probability from state i to state j (both i and j being integer readings of binary $(L-1)$-tuples). Letting $\mathbf{e}$ denote the 2^{L-1}-vector of all 1's and $\mathbf{v}$ an initial probability distribution vector, we immediately have

$$b_j = \mathbf{e}^T M(\pi)^{j-1}\mathbf{v}.$$

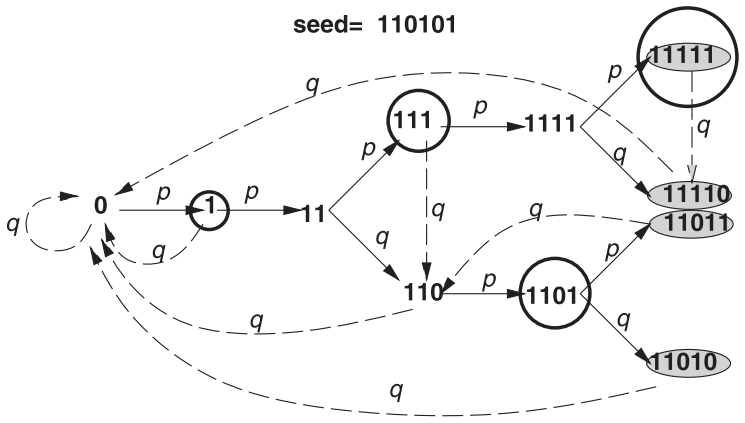

FIG. 1. The diagram of the minimal automaton for seed 101011. Return arcs are shown as broken lines. Nodes of $\mathcal{R}$ are shown shaded, and those of $\sigma(\mathcal{I})$ are circled. Notice: $\sigma(10101) = 1$, $\sigma(10111) = 111$, $\sigma(11101) = 1101$, $\sigma(11111) = 11111$.

In order to compare search seeds, we do not have to contend with the potentially enormous order of the matrix $M(\pi)$. Indeed, $M(\pi)$ is an extremely sparse matrix (with fewer than 2^L nonzero entries). As has been done by Buhler *et al.* (2003) and Choi and Zhang (2004), $\mathcal{DB}(\pi)$ can be replaced by an equivalent reduced finite-state automaton $\mathcal{P}(\pi)$, obtained by successively "merging" pairs of equivalent states,[1] as is commonly done in automata theory.

Since no state of $\mathcal{R}$ is equivalent to any other state,[2] the transition diagram of $\mathcal{P}(\pi)$ can be obtained by adjoining a set of arcs to the Patricia trie of the binary strings forming the reflecting set $\mathcal{R}$ for pattern π (Morrison, 1968). This Patricia trie is a balanced binary tree constructed in the following manner. For each string $u \in \mathcal{R}$ (of length $L-1$) the tree has a root-to-leaf path (from the root at level 0 to the leaves at level $L-1$) such that the node at level j is labeled by the length-j prefix of u. It follows that

1. the tree is balanced and has L levels;
2. its leaves are labeled by the strings of $\mathcal{R}$;
3. a node at level j is labeled by the length-j common prefix of the leaves of its subtree.

Given the Patricia trie of $\mathcal{R}$, automaton $\mathcal{P}(\pi)$ is completed as follows (refer to Fig. 1):

1. The original arcs of the Patricia trie are denoted *forward transitions*. Leaves (reflecting set nodes) have no forward transitions; internal nodes have either 1 or 2 forward transitions.
2. Denoting y (the label of) either a leaf or an internal node with a single forward transition, we construct the *return arc* (which is always a 0-transition) for v to the node whose label is the longest suffix of $y0$. All return arcs are referred to as *backward transitions*.
3. An arc corresponding to a 1-transition has label p, and an arc corresponding to a 0-transition has label q (the transition probabilities).

Whereas each member of the reflecting set is the label of a leaf of $\mathcal{P}(\pi)$, each $u \in \mathcal{I}$ is mapped to a unique node of $\mathcal{P}(\pi)$ such that its label $\sigma(u)$ is the longest suffix of u among the labels of the nodes of $\mathcal{P}(\pi)$. This set of nodes is denoted $\sigma(\mathcal{I})$.

These notions are illustrated in Fig. 1 with the diagram of the reduced automaton for $\pi = 101011$.

[1]States s_1 and s_2 with identical next-state transitions are merged into a state s with the same next-state transition, which is reached by the union of the transitions to s_1 and s_2.

[2]Indeed, since any $u \in \mathcal{R}$ has a single (0-transition) successor, it is potentially mergeable only with another state $u' \in \mathcal{R}$. But $u \neq u'$ implies $u0 \neq u'0$.

4. THE LEAKAGE MODEL (A DISSIPATIVE MARKOV SYSTEM)

The effectiveness of a seed is measured by its *hitting probability*. The comparison of two seeds is therefore based on their respective hitting probabilities for given p and n. In principle, the reduced automaton $\mathcal{P}(\pi)$ described above is the instrument upon which such comparisons can be based. Exhaustive search of the set of all seeds of given length L and weight w is currently the only available method for identifying an optimal seed. Such exhaustive search is a daunting task, because the set has size exponential in L and each $\mathcal{P}(\pi)$ has size at least exponential in $L - w$. Since the identification of effective, albeit not necessarily optimal, seeds remains a significant practical problem, there is a need for an intuitive elucidation of the mechanism that confers power to spaced seeds, with the objective to conceive simple but effective heuristics for seed design.

We shall view $\mathcal{P}(\pi)$ as a discrete-time system that stores a commodity (probability) at the nodes of its diagram. At each time unit, a nonleaf node redistributes the commodity it stored between its two successor nodes in proportions corresponding to the arc labels (p and q). A reflecting node (leaf), however, redistributes only a fraction q of its instantaneous probability and leaks out of the system a fraction p of this probability. We shall now show that this model, referred as the *leakage model*, can be initialized so that it correctly describes the hitting probabilities.

The time-evolution of our discrete-time system begins (at time-unit 1) in a configuration representing the sequences of $\mathcal{H}_1$. Specifically, each node of $\sigma(\mathcal{I})$, the image of the initial set $\mathcal{I}$, is "loaded" with the probability of the $(L-1)$-prefixes of the sequences of $\mathcal{H}_1$ that map to it. Thus, the system is initially loaded with total probability b_1/p. Since $\mathcal{H}_1 = \mathcal{F}$ and $\mathcal{I} = \{u : \overline{u}1 \in \mathcal{F}\}$, for each $u \in \mathcal{I}$, $\sigma(u)$ is loaded (accumulatively) with an amount $p(u) = p^{w(u)}q^{L-1-w(u)}$, where $w(u)$ is the weight of string u.

After the first transition, the system represents the distribution of the probabilities of the sequences in $\mathcal{H}_2$, whose total value is b_2/p. And in general, after j transitions, the system represents $\mathcal{H}_{j+1}$ with total (stored) probability b_{j+1}/p.

The crucial observation is that as the system evolves in time, probability is leaked out from the reflecting nodes, thereby reducing the amount of "stored" probability. This, in turn, reduces the probability of a first hit at any of the subsequent time units and correspondingly the hitting probability. Therefore, in order to maximize the hitting probability B_j, *the objective of seed design is the restraint of leakage*.

To gain additional insight into the mechanism, we note that commodity initially allocated to each node of set $\sigma(\mathcal{I})$ percolates towards the reflecting nodes (at level $L-1$), but only a fraction of it reaches this level, because a subset of the traversed nodes reflects backward a fraction q of the received commodity. Of the commodity reaching a reflecting node, a fraction p is leaked out, and a fraction q is again reflected back to some lower-level node. Clearly, the further upstream this destination node is, the longer the corresponding commodity will be retained (advantageously) within the system.

We now wish to relate the structure of the seed sequence to its behavior in the outlined model.

5. HEURISTIC CRITERIA FOR EFFECTIVE SEED DESIGN

The leakage model presented in the preceding section can now be used to provide some broad guidelines for effective seed design, thereby skirting the computationally intensive task of exhaustive evaluation, which may become totally impractical for seed sizes being presently contemplated (such as $L = 28$ in MegaBLAST [Zhang *et al.*, 2000]). It must be stressed that the considerations presented in this section are not precise optimization arguments, because the precise formal handling of the leakage mechanism is no simpler than that of the original problem. Rather, we intend to capitalize on the insights provided by the model to develop heuristics for beneficial design choices. The validation of this novel approach can only be provided by comparison with the currently known optimal seeds obtained by exhaustive search and reported in the literature.

Our approach is to focus on the salient features of the model and attempt to enhance their effect on the restraint of leakage. The qualitative analysis of the "leakage model" presented above points to two crucial features of a seed:

1. The initial commodity allocation, i.e., the distribution of the initial total probability p^{w-1} among the nodes $\sigma(\mathcal{I})$: it is desirable that only a small fraction be assigned to the reflecting nodes.
2. The levels of the return nodes, i.e., for each node with a backward arc, the level of the destination of this arc: it is desirable to reduce the number of arcs returning to high levels of the state diagram.

Remark. Incidentally, we note that for the contiguous seed 1^w, probability p^{w-1} is entirely allocated to the single reflecting node. Therefore, the system suffers an initial leakage p^w, which apparently is not offset by the fact that the reflecting node has a single return to level 0. Moreover, it is a simple exercise to verify that the uniformly spaced seed $(10)^{w-1}1$ has all initial allocations at levels $L-1$ and $L-2$, but only half of the reflecting nodes have returns to level 0. This provides intuitive justification of the proven inferiority of uniformly spaced seeds as compared with the contiguous seeds of the same weight (Choi and Zhang, 2004; Buhler *et al.*, 2003).

Concretely, since leakage occurs from the nodes of level $L-1$ of $\mathcal{P}(\pi)$, we shall focus on the top two levels, namely, $(L-1)$st and $(L-2)$nd levels, and analyze the following items.

- Item 1. Initial allocations to nodes at levels $L-1$ and $L-2$.
- Item 2. Reflections from the nodes of level $L-1$.

In order to relate the structure of π to Items 1 and 2, we now discuss a simple algorithm for determining the initial allocation and for the construction of the transition matrix of $\mathcal{P}(\pi)$. We begin with the observation that, starting from any node at level j, the percolating commodity that reaches level $L-1$ undergoes attenuation, which depends only upon j. This leads to the simplification that only the total commodity at any given level is of interest, and not its distribution among the nodes of that level.

Given the seed π, we construct the following three strings: $X(\pi)$, $Y(\pi)$, and $Z(\pi)$. First, $X(\pi)$ is obtained by replacing each 0 of π with Boolean variables $x_1, x_2, \ldots$. For example, for $\pi = 110101$, $X(\pi) = 11x_1 1x_2 1$. And $Y(\pi)$ is obtained by replacing the right-most 1 of $X(\pi)$ with 0. Finally, $Z(\pi)$ is obtained by replacing each 0 of π with a special Boolean symbol $*$. Symbol $*$ is defined by the property that $* \wedge x = x \wedge * = 1$ for $x = 0, 1$.

Next we define a sort of convolution of $Z(\pi)$ with $X(\pi)$ (refer to Table 1). Holding $X(\pi)$ fixed, suppose we shift $Z(\pi)$ j positions along $X(\pi)$. At this alignment of the two sequences, we compute the conjunction

TABLE 1. ILLUSTRATION OF THE DETERMINATION OF THE INITIAL ASSIGNMENT (UPPER-HALF) AND OF THE RETURN ARCS FROM THE REFLECTING LEVEL (LOWER-HALF) FOR SEED 110101[a]

$X(\pi)$	*1*	*1*	x_1	*1*	x_2	*1*	t_j	*Strings*	*Level j*
	*	1	1	*	1	*	$x_1 \wedge x_2$	**111111**	5
		*	1	1	*	1	x_1	**111101**, 111111	4
			*	1	1	*	x_2	**110111**, 111111	3
				*	1	1	x_2	—	2
					*	1	1	**110101**, 110111, 111101, 11111	1
						*	1	—	0
$Y(\pi)$	*1*	*1*	x_1	*1*	x_2	*0*	s_j	*Strings*	*Level j*
	*	1	1	*	1	*	$x_1 \wedge x_2$	**111110**	5
		*	1	1	*	1	0	—	4
			*	1	1	*	x_2	**110110**, 111110	3
				*	1	1	0	—	2
					*	1	0	—	1
						*	1	**111100**, **110100**, 110110, 111110	0

[a]In bold-face we show the assignments of strings to levels of $\mathcal{P}(\pi)$.

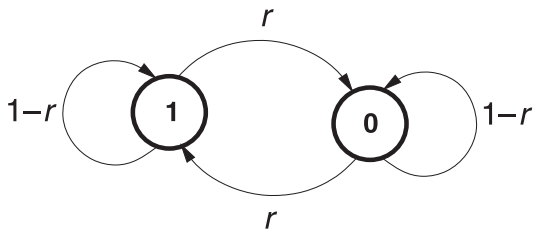

FIG. 2. A 2-state automaton generating the middle portion of a spaced seed. Here r is a design parameter.

of the ANDs of the aligned pairs of symbols (see upper half of Table 1): the resulting term (a conjunction t_{L-j}), when set equal to 1, provides a constraint on the variables $x_1, x_2, \ldots$ and therefore defines a subset of $\mathcal{I}$. Specifically, each member of this subset has the property that its $(L-j)$-suffix is the label of a node of $\mathcal{P}(\pi)$ at level $L-j$. In order to satisfy the requirement that for $u \in \mathcal{I}$, $\sigma(u)$ is the "longest" suffix, the strings of $\mathcal{I}$ assigned to level $i < L-1$ satisfy the (Boolean) condition

$$t_i \wedge \neg(\vee_{j=i+1}^{L-1} t_j) = 1.$$

This simple procedure enables us to determine the distribution of $\sigma(\mathcal{I})$ among the levels of $\mathcal{P}(\pi)$.

We return to Items 1 and 2 discussed above. With reference to Item 2, we wish to achieve returns from level $L-1$ to "low" levels. We note that if a seed terminates in a solid-1 suffix 1^f, $f > 1$, then each of the nodes at levels $L-1, L-2, \ldots, L-f+1$ has a return arc, due to the solid-1 suffix. On the other hand, none of these return arcs terminates on a node at levels $L-1, L-2, \ldots, L-f+1$. This beneficial effect (with reference to Item 2) suggests that the seed sequence be endowed with two end-runs of $w\beta$ consecutive 1s (β a design parameter to be determined).

Therefore we view the seed as the concatenation $e_1 m e_2$ of three strings, where e_1 and e_2 are, respectively, the left and right all-1 end-runs, and m is the middle portion. We assume that m is a random string generated by the 2-state automaton $\mathcal{T}$ given in Fig. 2, where probability r is a design parameter (this choice, rather than a simple Bernoulli sequence, provides additional flexibility). Automaton $\mathcal{T}$, which outputs 0 in state 0 and 1 in state 1, is started in state 0, and after $\nu - 1$ returns to state 0 (ν to be related to the design parameters), is halted before a transition from state 0 to state 1. Thus, string m is given by

$$m = 0^{z_0}(1^{u_1}0^{z_1})\ldots(1^{u_{\nu-1}}0^{z_{\nu-1}});$$

i.e., m, besides the prefix 0^{z_0} *initial 0-run*, is the concatenation of 0-1 *cycles* $1^{u_i}0^{z_i}$ ($u_i \geq 1, z_i \geq 1$), each consisting of a nonempty run of 1s followed by a nonempty run of 0s.

To see how the structure of m affects the allocation of probability to the top two levels of the system, we consider the sizes (numbers of literals[3]) of the conjunction terms pertaining to levels $L-1$ and $L-2$ (see Table 1). We make two observations.

1. The term of level $L-1$ has size ν in all cases, since each cycle (as well as the initial 0-run) contributes exactly one literal. This implies that only a fraction p^ν of the initial probability will be allocated to the top level. In the intent to maximize ν, one may be tempted to make ν equal to the number of 0s of m. However, this choice would imply $r = 1$ in $\mathcal{T}$ (i.e., $z_j = u_i = 1$ for all values of i and j) and would have a detrimental effect on t_{L-2}, as discussed below.
2. Thus, our objective is to select the value of r for $\mathcal{T}$ which minimizes the total initial leakage due to levels $L-1$ and $L-2$.

Since term t_{L-2} derives from a 2-position shift of the seed sequence (refer to Table 1) and each cycle has length at least 2, there is no interference between adjacent cycles. Therefore the contributions of the

[3]Note that the larger the size of the term the smaller the size of the set of strings it defines.

cycles are independent and can be added. The initial 0-run and the subsequent cycles will each contribute s literals as follows:

$$\begin{aligned} 0 \quad &\text{if} \quad u_j = z_j = 1 \\ s = 1 \quad &\text{if} \quad u_j = 1, z_j > 1 \ \text{ or } \ u_j > 1, z_j = 1 \\ 2 \quad &\text{if} \quad u_j \geq 2, z_j \geq 2 \end{aligned}$$

A simple calculation shows that the average number of literals contributed by the initial gap is $(2-r)$ and that the average contribution of the generic cycle is $2(1-r)$.[4] The expected value of the size of t_{L-2} is therefore

$$2 - r + (\nu - 1)2(1 - r). \tag{1}$$

Since a fraction p^i of the allocations to levels $L-i$ $(i = 1, 2)$ is leaked from the system, the objective function to be minimized is

$$\begin{aligned} \phi &= p^{2-r+(\nu-1)2(1-r)} \cdot p^2 + p^\nu \cdot p \\ &= p^{2-r+(\nu-1)2(1-r)+2} + p^{\nu+1}. \end{aligned} \tag{2}$$

The above expression contains the additional parameter ν, which can be eliminated. In our model of string m, the average number of 1s is readily found to be $(\nu-1)(r+2r(1-r)+3r(1-r)^2+\dots) = (\nu-1)/r$. Recalling that $2\beta w$ 1s have been assigned to the end-runs, we obtain the additional equation

$$\frac{\nu - 1}{r} = (1 - 2\beta)w \tag{3}$$

which we now use in Expression (2) to obtain

$$\phi = p^2 \left(p^{2-r+2(1-2\beta)wr(1-r)} + p^{(1-2\beta)wr} \right). \tag{*}$$

Analysis of this function is crucial for the selection of the parameter r as a function of w and p (for a chosen β). This will be discussed in the appendix.

We now turn our attention to parameter β. A reduction of β causes an increase of ν (Equation (3)) and therefore a decrease of the probability allocated to level $L-1$ (beneficial): This effect is correctly captured by $\phi_1(\beta) = p^\nu$, which is an increasing function of β. On the other hand, a reduction of β causes returns from top levels to occur at closer levels, thereby increasing the overall leakage over a chosen number of steps (detrimental). This effect may be captured by a number of functions expressing how the leakage of probability is related to the parameter β. Among these, we may consider the function $\phi_2(\beta) = p^{\beta w - 1}$, the attenuation occurring from the return level alluded above.[5]

These two contrasting phenomena point to an optimization problem whose detailed analysis seems extremely difficult. Since ϕ_1 and ϕ_2 are, respectively, increasing and decreasing functions of β, a reasonable heuristic is the minimization of $\phi_1 + \phi_2$. Differentiating with respect to β, we obtain

$$\begin{aligned} \frac{d(\phi_1 + \phi_2)}{d\beta} &= \left(-p^{(1-2\beta)wr+1} 2wr + p^{\beta w - 1} w \right) \ln p \\ &= w p^{\beta w - 1} \left(1 - 2r p^{2 + w(r - \beta(1+2r))} \right) \ln p. \end{aligned}$$

[4]The difference between the two cases is due to the fact that for the initial 0-run the case $u_0 = 1, z_0 > 1$ cannot occur.

[5]Incidentally, there is an additional interpretation of $p^{\beta w}$. Consider the time sequence of leakages due to a unit of probability allocated to the front level of the automaton $\mathcal{P}(\pi)$. It can be shown that $1/p^{\beta w}$ is the first moment of this sequence, a quantity we wish to maximize.

Equating to 0, we have

$$\beta = \frac{r}{1+2r} + \gamma$$

where parameter

$$\gamma = \frac{1}{w(1+2r)}\left(2 - \frac{\ln(2r)}{\ln(1/p)}\right).$$

Straightforward analysis reveals that $\gamma \in [-0.05, 0.08]$ for $r \in [0.6, 0.7]$, $p \in [0.5, 0.9]$, and $w \geq 10$, so that $\beta \in [0.24, 0.35]$ in this range of parameters.

The preceding considerations provide broad guidelines for the design of search seeds. We shall choose $\beta = 1/4$ and $r = 0.65$, to be adopted regardless of the value p. Thus, the design starts with the weight w, a measure of the search cost. Since both the computed values of end-run length and the number of runs of zeros in the middle portion may not be integers, discretion may be used in selecting floors or ceilings of noninteger quantities. Such discretion is used in the following algorithm.

Seed design algorithm

1. Allocate $E = \lceil 2\beta w \rceil$ 1s to the end-runs e_1 and e_2. Specifically, $|e_1| = \lfloor E/2 \rfloor$ and $|e_2| = \lceil E/2 \rceil$.
2. The remaining $F = w - E$ 1s are assigned to the middle portion m. Since the initial 0-run of m has expected length $1/r$, the number of 0s of m is chosen as $Z = 1 + F$ (or $Z = 2 + F$). The number of runs ($\nu = (1 - 2\beta)rw + 1$) is chosen (using rounding) as

$$\nu = \lfloor 0.5 \times 0.65 \times w + 1 \rfloor \quad \text{or} \quad \nu = \lceil 0.5 \times 0.65 \times w + 1 \rceil.$$

3. Generate a set of t (for example, $t = 10$) sample binary sequences m beginning and ending with 0's, with F 1s, Z 0s, and $\nu - 1$ runs of 1s. Construct $e_1 m e_2$.
4. For each sample sequence, construct $e_1 m e_2$, and evaluate its hitting probability for selected p and n.
5. Select the sample seed achieving the highest hitting probability.

For example, according to Steps 1 and 2 above, we would obtain the following seed specifications.

Weight w	Endlengths	F	Z	ν	L
10	2-3	5	6	4	16
11	3-3	5	6	4	17
12	3-3	6	7	5	19
13	3-4	6	7	5	20
14	3-4	7	8	6	22
15	4-4	7	8	6	23

In the next section, we compare currently known optimal seeds with corresponding seeds (same w, p and n) synthesized with the above procedure.

6. EXPERIMENTAL RESULTS

The selection criteria outlined in the previous section reflect the fact that the effectiveness of a seed depends mostly upon the numbers of ones at the two ends of the seed and on the number of gaps in its middle portion. In this section, we shall validate the criteria by comparing the seeds obtained according to them to the corresponding optimal and contiguous seeds. First, we present an experimental analysis of the effects of the parameters on the sensitivity of a spaced seed. Then, we report the seeds selected on the basis of the stated criteria and compare them with the optimal seeds obtained through exhaustive search, or the contiguous seeds when the optimal seeds are unknown.

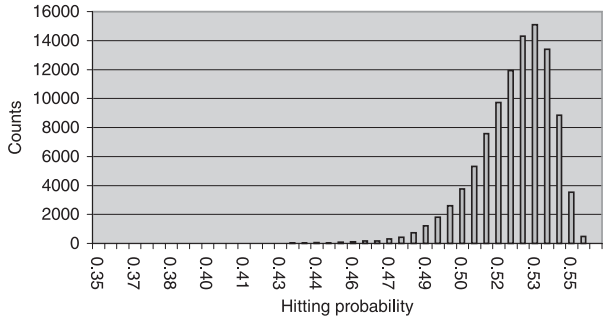

FIG. 3. Frequency histogram of sensitivities of spaced seeds of weight 15 and length 23. Here $n = 64$ and $p = 0.8$. The abscissa interval $[0.35, 0.56]$ is conventionally subdivided into 42 bins.

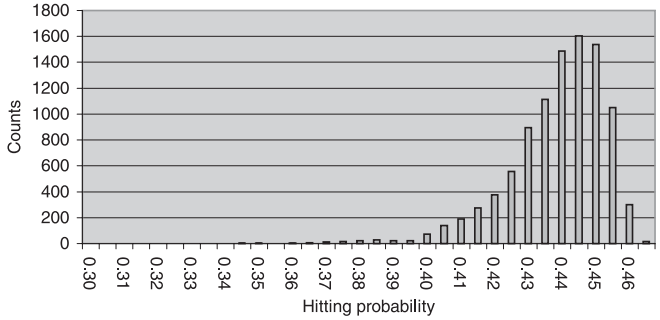

FIG. 4. Frequency histogram of spaced seeds of weight 11, for length increasing from 11 to 18. Here $n = 64$ and $p = 0.8$. The abscissa interval $[0.30, 0.47]$ is subdivided into 34 bins.

6.1. Sensitivity distribution of spaced seeds

There are 101,850 spaced seeds with weight 15 and weight 23. When $p = 0.8$ and $n = 64$, the average sensitivity is 0.5239; the standard deviation is 0.01583. Figure 3 is the histogram of the sensitivities of all the spaced seeds with weight 15 and length 23 when $n = 64$ and $p = 0.8$. The abscissa represents the sensitivity interval $[0.35, 0.56]$ that is uniformly divided into 42 bins. We notice that the histograms are decidedly skewed to the right, suggesting that a randomly picked spaced seed is likely to be fairly effective. Similar skewness was observed for other values of p and L. For instance, Fig. 4 displays the skewness pattern for all the spaced seeds of weight 11 but various lengths from 11 to 18 (with mean sensitivity 0.44128 and standard deviation 0.01465). The next two experiments deal, for fixed $n = 64$ and $p = 0.8$, with seeds of fixed weight ($w = 15$) and length ($L = 23$), but with varying features (end-run length and number of gaps in the middle portion). The histograms displayed in Fig. 5 pertain to seeds whose end-runs have lengths $s = 1, 2, \ldots, 5$. As the length s of the end-runs increases from 1 to 4, the bulk of histogram shifts towards the right, corresponding to an improvement in sensitivity. However, when this length has value 5, the histogram-bulk shifts back to left, revealing an optimum value around the value $\lceil w/4 \rceil$. Finally, in Fig. 6, we explore the effect of the number of gaps in the middle portion, while keeping fixed the length of the end-runs (at value 4). Note that the middle portion contains $15 - 8 = 7$ ones and $23 - 8 - 7 = 8$ zeros. The most effective seeds, as a set, have 5 or 6 gaps. These diagrams provide strong evidence that end-run length and number of gaps are critical features for the sensitivity of a spaced seed, and we note that the observed phenomena are in remarkable agreement with our analysis in the preceding section.

6.2. Validation of the selection criteria

To validate our proposed criteria for effective seed design, we obtained good spaced seeds for moderate weights using the sampling approach outlined in the last section and compared their sensitivities with that of the optimal seeds of the same weight reported by Choi and Zhang (2004). For each chosen w, L, p, and

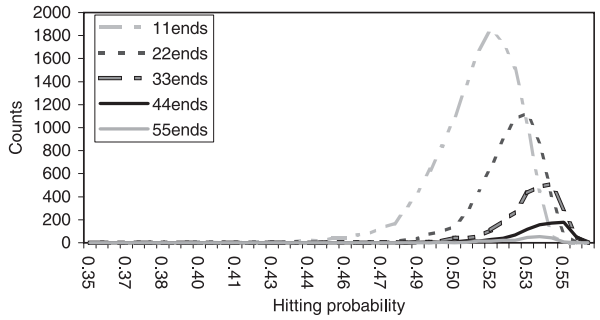

FIG. 5. Histograms displaying the effects of end-run length on seed sensitivity for weight 15 and length 23 (again, $n = 64$ and $p = 0.8$). Note that sensitivity improves as the end-run length increases to value 4 and deteriorates for value 5.

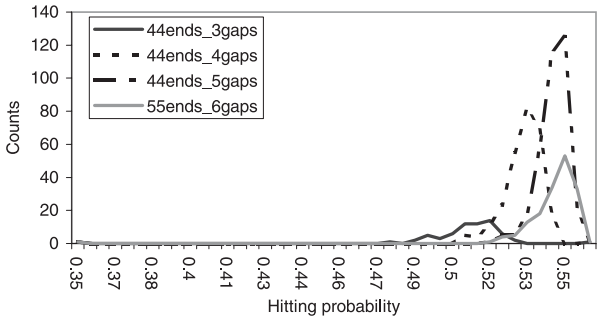

FIG. 6. Histograms displaying the effects of the number, ν, of middle-portion gaps on seed sensitivity, for fixed weight 15, length 23, and end-run length 4 (as usual, $n = 64$ and $p = 0.8$). Note that sensitivity improves as ν grows from 1 to 4.

$n = 64$, we choose at random 10 seeds with desired end-runs and number of gaps and selected the best.[6] The experimental results are summarized in Table 2. As can be seen, the selected seed has near optimal sensitivity for each weight and similarity.

Since the exhaustive search approach is very time consuming, optimal spaced seeds with weight greater than 18 have not been identified. Hence we compare our selected seed with a specified weight from 19 to 24 with the corresponding contiguous seed. Table 3 lists their sensitivities when $n = 128$ and $p = 0.7, 0.8, 0.9$. The comparison indicates that the sensitivity could be improved over 100% for aligning highly similar sequences whose similarity is over 70%.

We also compared the selected seeds to the optimal spaced seeds identified by Choi *et al.* (2004) in connection with real DNA sequences. We conducted a series of homology search experiments using PatternHunter (academic version 2.0) on the following four datasets, where the first named genomic sequence was used as the query sequence:

(i) *Haemorhiza influenza* (NC_000907, 1.83 Mb) and *Escherichia coli* (NC_000913, 4.63 Mb) genomes;
(ii) *Bifidobacterium longum* (NC_004307, 2.23 Mb) and *Haemorhiza influenza* genomes;
(iii) *Archaeoglobus fulgidus* (NC_000917, 2.16 Mb) and *Haemorhiza influenza* genomes;
(iv) a 232 kb segment in human chromosome 22 and a 4.95 Mb segment in mouse chromosome 11.

Since we want to test the performance of the selected spaced seeds for homology search in a large DNA database, we deliberately choose these four datasets without using any genomic information (such as GC content and coding regions) on these sequences. We test only the seeds selected for weight from 10 to 13 and

[6]The random sampling was done as follows: all binary sequences of length $Z + F - 2$ were generated and those satisfying the requirements were selected and stored in a list, from which we carried out uniform random sampling.

TABLE 2. COMPARING THE SENSITIVITY OF OUR SELECTED SEEDS WITH THAT OF KNOWN OPTIMAL SEEDS (AS REPORTED IN CHOI *et al.* [2004])

w	*Similarity (%)*	*Best spaced seeds found by sampling*	*Sensitivity*	*Optimal sensitivity*	*Sensitivity ratio (our seed/optimal seed)*
10	65	1101100010110111	0.37487	0.38093	98.4%
	70	1101100010110111	0.58736	0.59574	98.6%
	75	1101100011010111	0.80112	0.80112	100 %
	80	1100111010010111	0.93569	0.93685	99.8%
	85	1101011011000111	0.98880	0.99010	99.8%
11	65	11100111010010111	0.26487	0.26721	99.1%
	70	11100010110110111	0.46351	0.46712	99.2%
	75	11101001001110111	0.69397	0.69596	99.7%
	80	11100101100110111	0.87792	0.88240	99.4%
	85	11101001001110111	0.97542	0.97601	99.9%
12	65	1110011110100010111	0.17916	0.18385	97.4%
	70	1110010110001110111	0.34823	0.35643	97.6%
	75	1110100001110110111	0.57692	0.58709	98.2%
	80	1110010010011110111	0.80182	0.81206	98.7%
	85	1110100111000110111	0.94798	0.95212	99.6%
13	65	11101110010101001111	0.12219	0.12327	99.1%
	70	11100110110010101111	0.26443	0.26475	99.8%
	75	11101101101010001111	0.47964	0.48210	99.4%
	80	11100110110010101111	0.73001	0.73071	99.9%
	85	11101100101011001111	0.91581	0.91747	99.8%
14	65	11101001110011010001111	0.07966	0.08179	97.4%
	70	11101011001110100011111	0.18954	0.19351	97.9%
	75	11100101110001011011111	0.38074	0.38805	98.1%
	80	11101010011001110011111	0.63225	0.66455	95.1%
	85	11100100110100111011111	0.86566	0.87223	99.2%
15	65	111101010001110011011111	0.05253	0.05340	98.3%
	70	111100010100110111011111	0.13630	0.13867	98.3%
	75	111101011100100011011111	0.30078	0.30546	98.5%
	80	111100011010110011011111	0.55086	0.55623	99.0%
	85	111100011010100011011111	0.80986	0.81601	99.2%
16	65	1111001100110010110101111	0.03413	0.03495	97.6%
	70	1111001110100100110101111	0.09697	0.09894	98.0%
	75	1111001101011010001101111	0.23408	0.23781	98.4%
	80	1111001110101001100101111	0.46966	0.47319	99.2%
	85	1111000101100110101101111	0.75056	0.75339	99.6%
17	65	11110100110011010110011111	0.02207	0.02270	97.2%
	70	11110011101010100110011111	0.06833	0.07000	97.6%
	75	11110101110011001001011111	0.18189	0.18347	99.1%
	80	11110011010100110011011111	0.39448	0.39817	99.1%
	85	11110101110010010110011111	0.68031	0.68807	98.9%
18	65	111100101001100110111100011111	0.01389	0.01463	94.9%
	70	111100111001101110101000011111	0.04741	0.04915	96.4%
	75	111101101001110000100110111111	0.13651	0.14011	97.4%
	80	111101101011100011001001111	0.32485	0.33050	98.3%
	85	111101101101110001000101111	0.60813	0.62022	98.1%

TABLE 3. COMPARING THE SENSITIVITY OF OUR SELECTED SEEDS, BASED ON OUR SEED DESIGN ALGORITHM AT $p = 0.7$, WITH THAT OF THE CONTIGUOUS SEEDS

w	*Similarity (%)*	*Best spaced seeds found by sampling*	*Sensitivity*	*Sensitivity of the contiguous seed*
18	70	111101101011100011001001111	0.12137	0.05430
	80		0.63461	0.35760
	90		0.99511	0.92960
19	70	1111100110101001110010101111	0.08745	0.03791
	80		0.55589	0.29321
	90		0.99071	0.89851
20	70	11111001100011101001010111011111	0.06257	0.02641
	80		0.48296	0.23861
	90		0.98470	0.86177
21	70	111110100111010111001001001111111	0.04398	0.01838
	80		0.40740	0.19304
	90		0.97343	0.82029
22	70	1111101001101101001110010100111111	0.03123	0.01277
	80		0.34730	0.15545
	90		0.96170	0.77520
23	70	11111101010010011011000110110111111	0.02167	0.00887
	80		0.28518	0.12473
	90		0.93996	0.72770
24	70	111111001010011000111001101011011111	0.01534	0.00616
	80		0.24057	0.09980
	90		0.92197	0.67891

$p = 0.7$. The selected seed of weight 10 (1101100010110111) outperforms the corresponding optimal seed 1101100011010111 on the dataset (iv); the selected seed of weight 11 (11100010110110111) outperforms the corresponding optimal seed 111010010100110111 on the datasets (i), (ii), and (iii); the selected seed of weight 12 (1110010110001110111) outperforms the corresponding optimal seed 111010110100110111 on the dataset (iv); the selected seed of weight 13 (11100110110010101111) outperforms the corresponding optimal seed 11101011001100101111 on the datasets (i), (iii), and (iv). This shows that the seeds selected according to the criteria proposed here are also quite good when applied to genomic sequences.

7. CONCLUSIONS

In this paper, we have investigated the mechanism that confers gapped seeds their superiority over corresponding ungapped seeds. The homology search process has been formulated in terms of the time-evolution of a network receiving an initial allotment of probability and slowly leaking it to the exterior: the slower the leakage, the higher the performance. In this model, we have identified features of the seed which affect the leakage phenomenon and have reduced them to end-run lengths and structure of the seed middle-portion. On this basis, we have formulated a seed selection procedure yielding seeds whose performance is practically indistinguishable from that of optimal seeds, whenever known. It may be argued that seed selection, being a one-time event, could bear the cost of exhaustive search; it is, however, of significance to know that such investment yields a negligible gain. While the deep, daunting, and mathematically significant question of the characterization of optimal seeds remains unanswered, we believe that our proposal provides a satisfactory solution of the problem from a practitioner's viewpoint.

APPENDIX

Here we analyze the the function $\phi(r)$ in Expression (∗). Recall the function

$$\phi(r) = p^2\left(p^{2-r+2(1-2\beta)wr(1-r)} + p^{(1-2\beta)wr}\right)$$

which we wish to minimize. For notational convenience, we let $A = (1-2\beta)w$ and $\psi(r) = \phi(r)/p^2$. Differentiating ψ with respect to r, we obtain

$$\frac{d\psi(r)}{dr} = \left[(1+4Ar-2A)p^{2+2Ar(1-r)} - Ap^{(A+1)r}\right]p^{-r}\ln(1/p).$$

Let

$$g_{A,\,p}(r) := (1+4Ar-2A)p^{2+2Ar(1-r)} - Ap^{(A+1)r},$$

then

$$g'_{A,\,p}(r) = \left[2A(2r-1)(1+4Ar-2A)p^{2+2Ar(1-r)} + A(A+1)p^{(A+1)r}\right]\ln(1/p) + 4Ap^{2+2Ar(1-r)}$$
$$> 0$$

for $r \in [1/2, 1)$. That is, $g_{A,\,p}(r)$ is increasing in r on $[1/2, 1)$. Furthermore, it can be verified directly that

$$g_{A,\,p}(1/2) = p^{2+A/2} - Ap^{(A+1)2} < 0,$$
$$g_{A,\,p}(1) = (2A+1)p^2 - Ap^{A+1} > Ap^2(1-p^{A-1}) > 0.$$

Hence there exists a unique $r^*(A,\ p)$ (abbreviated to r^*) such that $g_{A,\,p}(r^*) = 0$, and therefore $\psi'(r^*) = 0$, $\psi'(r^*-) < 0$ and $\psi'(r^*+) > 0$ verifying that the extremum is indeed a minimum. The value r^* can be found numerically. Sample values of r^* are reported in the table below.

	Value of r^* when p is			
$2\beta w$	0.2	0.4	0.6	0.8
5	0.712	0.714	0.716	0.719
10	0.642	0.652	0.665	0.685
15	0.61	0.621	0.636	0.662

ACKNOWLEDGMENTS

F.P. Preparata was partially supported by the Kwan Im Thong Chair at the National University of Singapore, and L. Zhang and K.P. Choi were partially supported by BMRC Research Grant BMRC01/1/21/19/140.

REFERENCES

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. 1997. Gapped Blast and Psi-Blast: A new generation of protein database search programs. *Nucl. Acids Res.* 25, 3389–3402.

Brejovà, B., Brown, D., and Vinař, T. 2004. Optimal spaced seeds for homologous coding regions. *J. Bioinfor. Comput. Biol.* 1, 595–610.

Buhler, J., Keich, U., and Sun, Y. 2003. Designing seeds for similarity search in genomic DNA. *Proc. 6th Intl. Conf. Comput. Mol. Biol. (RECOMB '03)*, 67–75.

Burkhardt, S., Crauser, A., Ferragina, P., Lenhof, H.-P., Rivals, E., and Vingron, M. 1999. q-gram based database searching using a suffix array. *Proc. 3rd Intl. Conf. Comput. Mol. Biol. (RECOMB '99)*, 11–14.

Burkhardt, S., and Kärkkäinen, J. 2001. Better filtering with gapped q-grams. *Proc. 12th Ann. Symp. on Combinatorial Pattern Matching*, 73–85.

Califano, A., and Rigoutsos, I. 1995. FLASH: Fast look-up algorithm for string homology. Technical report, IBM, T.J. Watson Research Center.

Choi, K.P., Zeng, F., and Zhang, L. 2004. Good spaced seeds for homology search. *Bioinformatics* 20, 1053–1059.

Choi, K.P., and Zhang, L. 2004. Sensitivity analysis and efficient method for identifying optimal spaced seeds. *J. Comput. Sys. Sci.* 68, 22–40.

Delcher, A.L., Kasif, S., Fleischmann, R.D., Peterson, J., White, O., and Salzberg, S.L. 1999. Alignment of whole genomes. *Nucl. Acids Res.* 27, 2369–2376.

Gotea, V., Veeramachaneni, V., and Makalowski, W. 2003. Mastering seeds for genomic size nucleotide BLAST searches. *Nucl. Acids Res.* 31, 6935–6941.

Hardison, R., Oeltjen, J., and Miller, W. 1997. Long human–mouse sequence alignments reveal novel regulatory elements: A reason to sequence the mouse genome. *Genome Res.* 7, 966–969.

Karp, R., and Rabin, M.O. 1987. Efficient randomized pattern-matching algorithms. *IBM J. Res. Develop.* 31, 249–260.

Keich, U., Li, M., and Ma, B. 2004. On spaced seeds for similarity search. *Disc. Appl. Math.* 138, 253–263.

Kent, W.J. 2002. BLAT: The BLAST-like alignment tool. *Genome Res.* 12, 656–664.

Kisman, D., Li, M., Ma, B., and Wang, L. 2005. tPatternHunter: Gapped, fast and sensitive translated homology search. *Bioinformatics* 21, 542–544.

Kucherov, G., Noé, L., and Roytberg, M. 2004. A unifying framework for seed sensitivity and its application to subset seeds. Technical report 5374, INRIA, France.

Li, W.-H., Gu, Z., Wang, H., and Nekrutenko, A. 2001. Evolutionary analysis of the human genome. *Nature* 409, 847–849.

Lipman, D.J., and Pearson, W. 1985. Rapid and sensitive protein similarity searches. *Science* 227, 1435–1441.

Ma, B., Tromp, J., and Li, M. 2002. PatternHunter—faster and more sensitive homology search. *Bioinformatics* 18, 440–445.

Morrison, D.R. 1968. PATRICIA—practical algorithm to retrieve information coded in alphanumeric. *J. ACM* 15, 514–534.

Ning, Z., Cox, A.J., and Mullikin, J.C. 2001. SSAHA: A fast search method for large DNA databases. *Genome Res.* 11, 1725–1729.

Noé, L., and Kucherov, G. 2003. YASS: Similarity search in DNA sequences. Technical report 4852, INRIA, France.

Pevzner, P., and Waterman, M.S. 1995. Multiple filtration and approximate pattern matching. *Algorithmica* 13, 135–154.

Schwartz, S., Kent, W.J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R.C., Haussler, D., and Miller, W. 2003. Human-mouse alignment with BLASTZ. *Genome Res.* 13, 103–107.

Wheeler, D.L., Chappey, C., Lash, A.E., Leipe, D.D., Madden, T.L., Schuler, G.D., Tatusova, T.A., and Rapp, B.A. 2003. Database resources of the National Center for Biotechnology. *Nucl. Acids Res.* 31, 28–33.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. 2000. A greedy algorithm for aligning DNA sequences. *J. Comp. Biol.* 7, 203–214.

Address correspondence to:
Franco P. Preparata
Computer Science Department
Brown University
115 Waterman Street
Providence, RI 02912-1910

E-mail: franco@cs.brown.edu