BIOINFORMATICS

Vol. 22 no. 14 2006, pages e350–e358
doi:10.1093/bioinformatics/btl240

Efficient identification of DNA hybridization partners in a sequence database

Tobias P. Mann[1,*] and William Stafford Noble[1,2]

[1]Department of Genome Sciences and [2]Department of Computer Science and Engineering, University of Washington, Seattle WA, USA

ABSTRACT

Motivation: The specific hybridization of complementary DNA molecules underlies many widely used molecular biology assays, including the polymerase chain reaction and various types of microarray analysis. In order for such an assay to work well, the primer or probe must bind to its intended target, without also binding to additional sequences in the reaction mixture. For any given probe or primer, potential non-specific binding partners can be identified using state-of-the-art models of DNA binding stability. Unfortunately, these models rely on dynamic programming algorithms that are too slow to apply on a genomic scale.

Results: We present an algorithm that efficiently scans a DNA database for short (approximately 20–30 base) sequences that will bind to a query sequence. We use a filtering approach, in which a series of increasingly stringent filters is applied to a set of candidate *k*-mers. The *k*-mers that pass all filters are then located in the sequence database using a precomputed index, and an accurate model of DNA binding stability is applied to the sequence surrounding each of the *k*-mer occurrences. This approach reduces the time to identify all binding partners for a given DNA sequence in human genomic DNA by approximately three orders of magnitude, from two days for the ENCODE regions to less than one minute for typical queries. Our approach is scalable to large DNA sequences. Our method can scan the human genome for medium strength binding sites to a candidate PCR primer in an average of 34.5 minutes.

Availability: Software implementing the algorithms described here is available at http://noble.gs.washington.edu/proj/dna-binding

Contact: mann@gs.washington.edu

1 INTRODUCTION

Many fundamental methods in molecular biology rely on binding between complementary DNA molecules. For instance, the polymerase chain reaction (PCR) (Saiki *et al*., 1988) relies on the specific binding of short DNA primer sequences to the DNA of interest. PCR is used in a multitude of contexts (Innis *et al*., 1999), from disease diagnosis (Kaltenboeck and Wang, 2005) to gene expression measurement (Wong and Medrano, 2005). DNA microarrays (Schena *et al*., 1995) also rely on the specific hybridization of array probes to DNA sequences in a mixture in order to measure gene expression or determine sample genotypes (Stoughton, 2005).

Assays that rely on hybridization are compromised when primers or probes bind non-specifically to DNA molecules that are not their targets (Chou *et al*., 1992). In the presence of non-specific hybridization, measurement accuracy in quantitative assays can be severely compromised, especially when the hybridization target is present in low abundance. Even in the context of non-quantitative PCRs, non-specific binding can lead to the formation of undesired products that compete with the reaction of interest and reduce reaction yields. Therefore, assessing hybridization specificity is an important part of the design of these reactions.

The most straightforward approach to assessing hybridization specificity would be to query every potential binding site in the background DNA for binding affinity. In most experiments, the background DNA that comprises the reaction mixture consists of the genome of the organism being studied. Hence, for the human genome, this approach requires evaluating approximately six billion possible binding sites, corresponding to the two strands of each chromosome.

In practice, applying state-of-the-art DNA binding models on a genomic scale is not computationally feasible. These models use dynamic programming algorithms with a computational complexity of $O(nm)$ for two sequences of length m and n, respectively (Garel and Orland, 2004; Dimitrov and Zuker, 2004), and the complexity of querying an entire genome is $O(gmn)$, where g is the number of bases in the genome, m is the sequence length, and n is the size of the genomic subsequence queried at each position. In our experiments, scanning the complete human genome for binding sites to a 25-mer probe requires approximately 180 days of CPU time. For most primer or probe design applications, this is clearly too long to wait.

Current practical methods for predicting non-specific binding of a given DNA sequence rely on heuristic approximations. Perhaps the most commonly used method for identifying binding sites between a query DNA sequence and a target genome predicts binding sites based upon a pre-specified maximum number of mismatches between the probe's reverse complement and the target (Kent *et al*., 2002; Lowe *et al*., 1990; Wang and Seed, 2003; Xu *et al*., 2004). As we demonstrate below, this approach is inaccurate because sequences can stably bind in the presence of bulge loops, which correspond to insertions and deletions in an alignment. An alternative method for identifying non-specific binding sites relies on the BLAST algorithm or other alignment based criteria (Altschul *et al*., 1990; Haas *et al*., 2003; Zakour *et al*., 2004; Andersson *et al*., 2005). This approach, too, is inaccurate, primarily because BLAST is designed to detect statistically significant sequence homology, rather than sequence binding partners.

We propose a filter- and index-based method, shown in Figure 1, for rapidly identifying binding partners of a given query sequence. In

*To whom correspondence should be addressed.

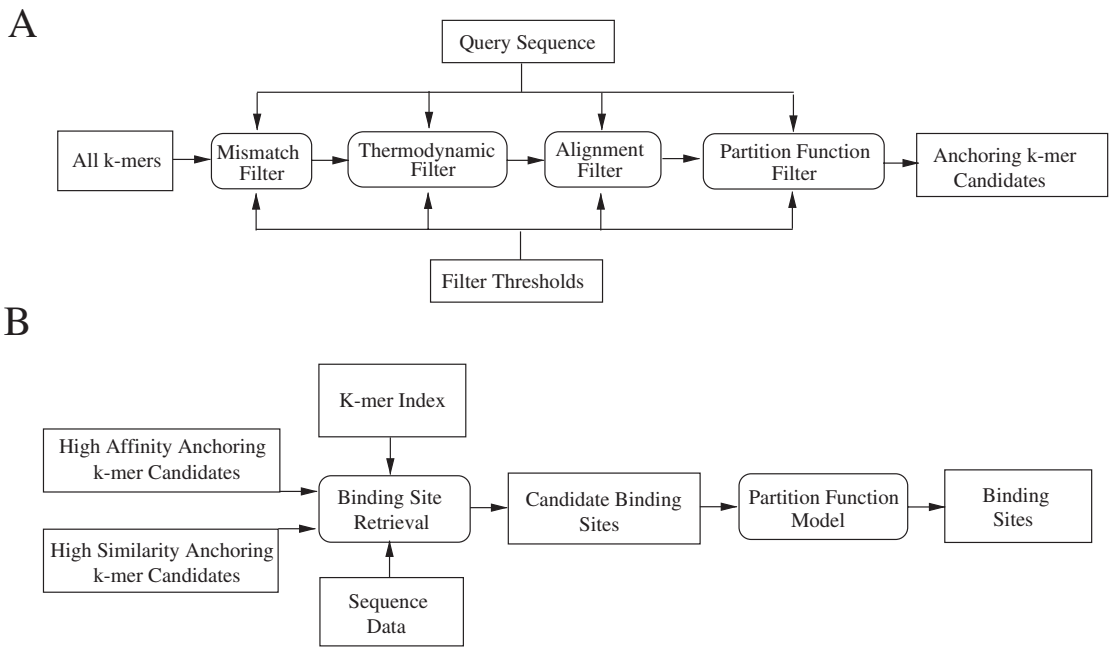

Fig. 1. Overview of filtering algorithm. **(A)** *k*-mer filtering. All *k*-mers for a specified value of *k* are input to the mismatch filter, along with a set of pre-chosen similarity thresholds. The four filters eliminate *k*-mers in turn, producing as output a list of candidate *k*-mers that could anchor a binding site. We subject all *k*-mers to two sets of thresholds, producing two sets of candidates binding site anchors. One set yields *k*-mers that have high thermodynamic affinity to the query, and the other set yields *k*-mers that have high sequence similarity to the query. (**B**) Candidate retrieval and evaluation. The *k*-mers that passed the filtering steps in (**A**) are located in the genome sequence using a precomputed index. We examine only those sites where a candidate *k*-mer from one group occurs with close proximity to a candidate *k*-mer from the other group. These candidate binding sites are then tested for binding affinity using the partition function model, and all sequences that bind to the query with greater than a target affinity are reported.

the initial stage (A), we consider all possible *k*-mers of a given length and identify *k*-mers that could anchor a binding site to the query sequence. This stage includes four filters that are designed to recognize various aspects of DNA binding stability. Two of the filters were developed for this application. The filters are applied in order of increasing computational complexity, so that most *k*-mers are excluded by the simplest filters. Using our approach and considering 10-mer anchors, typically more than 99% of target 10-mers are excluded from further consideration. In stage (B), we use a suffix array index to rapidly extract the sequence context of all occurrences of the *k*-mers obtained in the first step. These candidate binding sites are then evaluated using a model of DNA binding. Because so many *k*-mers are excluded at the outset, we can afford to apply an accurate model of DNA binding in the second stage of the algorithm.

Using our method, we achieve rapid and comprehensive identification of likely binding sequences. The first stage of the algorithm reduces the sequence search space by three orders of magnitude. The second stage is quick because many of the occurrences of the *k*-mers that pass the filtering stage can be eliminated by further filtering. Furthermore, our filter thresholds are set to achieve this speedup while retaining 100% accuracy, compared with considering every possible binding site in the target genome. Our approach reduces the amount of time to scan a sample 30 MB sequence from two days to under a minute for typical queries.

2 ALGORITHMS

We hypothesize that binding sites in genomic DNA can be comprehensively retrieved by first identifying short regions of agreement between the query sequence and the genomic DNA, and then examining the sequences containing these short regions of agreement with accurate models of DNA binding. We base this hypothesis on the observation that the thermodynamic instability of unbound bases in a DNA duplex (so-called 'loops') limits the amount of disagreement between a query sequence and any of its binding sites.

In particular, our method relies on a set of filters to identify *k*-mers that have good agreement with the query sequence, and could therefore anchor a binding site. In this section, we describe state-of-the-art models of DNA binding and then explain how our filters relate to those methods.

2.1 Partition function models of DNA binding

The overall goal of a model of DNA binding is to predict the *binding affinity* of a given pair of DNA sequences. The binding of two single stranded DNA molecules to form a dimer is a reversible reaction, and the binding affinity reflects the balance of association and dissociation reactions in a large population of molecules at thermodynamic equilibrium. When the binding affinity is large, then the dimer form is favored, and when the binding affinity is small, then the single stranded forms are favored. Currently, the most accurate models use thermodynamic reference data to approximate a quantity called the partition function. The partition function accounts for all ways in which two sequences can interact, and weights each interaction according to the energy of the interaction. The value of this function is proportional to the binding affinity.

In order to predict the binding affinity of two DNA sequences, partition function models of DNA binding stability consider physically realistic alignments between the two molecules, weighting each alignment according to its energy. The energy of a given alignment depends on a number of factors. The primary factor is the number of bases that are paired, and whether or not the paired bases are adjacent. In general, adjacent base pairs have higher binding energy than isolated base pairs due to so-called stacking interactions between adjacent base pairs. Conversely, runs of consecutive mismatches between the two strands, called loops, reduce the energy of the alignment. Extra energetic penalties are assigned to asymmetric loops. Bulge loops, corresponding to insertion and deletions in alignments, are also energetically unfavorable. Finally, the energetic stability of a single internal mismatch has been found to vary significantly according to the sequence context (SantaLucia, Jr, 1998; SantaLucia, Jr and Hicks, 2004), and these effects must also be taken into account.

Recently, efficient dynamic programming methods have been developed to compute the affinity of two DNA molecules (Garel and Orland, 2004; Dimitrov and Zuker, 2004). In this approach, a dynamic programming algorithm computes the sum of the exponentials of the energies of almost every alignment in which one molecule has at least one base pair with the other molecule. This sum is then proportional to the binding affinity. In this work, we use the HYBRID software (Markham and Zuker, 2004), which implements one such dynamic programming algorithm. However, our method does not rely on the specifics of the HYBRID software: our filters are designed to account for known, generic features of DNA affinity, and other models of DNA binding could be used in the final step to evaluate the filtered list of candidates. Indeed, although HYBRID and similar methods represent the state of the art in determining the affinity of two DNA sequences, they are known to systematically neglect some alignments that are important in some contexts.

2.2 An efficient algorithm for finding binding sites

Our goal is to identify all of the sequences in a database that bind to a query sequence according to a given partition function model of DNA binding. We do this in two stages, as described in Figure 1. First, we identify two groups of k-mers. One group of k-mers consists of k-mers with high sequence similarity to the query, and the other group of k-mers consists of k-mers with high thermodynamic affinity to the query sequence. Each group is defined as the set of k-mers that pass through a series of four filters described below; both groups are passed through the same filters but each group is identified by the use of different filter thresholds for each filter. In the second stage, each location in the sequence where there is a k-mer from the high affinity group within a pre-specified distance of a k-mer from the high similarity group is retrieved, along with flanking sequence. These candidate binding sites are then evaluated using the partition function model. The output of the algorithm is a list of binding partners for the query sequence.

In the first stage of our approach, we consider all k-mers of a given length, and we use a series of four filters to eliminate k-mers that have little affinity or similarity to the query. Each filter is designed to reject those k-mers that have little affinity to the query, and thus restrict the number of candidate binding sequences that must be considered. Furthermore, the filters are designed to be increasingly stringent, and are applied in order of increasing computational complexity; the first filter is very fast but will pass some k-mers with low affinity to the query, whereas the last filter is more expensive to compute but will reject all those k-mers with little thermodynamic affinity to the query. Each filter must be applied in conjunction with a threshold. The threshold for each filter is determined empirically by examining characteristics of binding sites predicted by the partition function model of DNA binding. These thresholds are chosen conservatively, so that each filter will pass some k-mers with low affinity to the query rather than discard k-mers that could anchor a binding site.

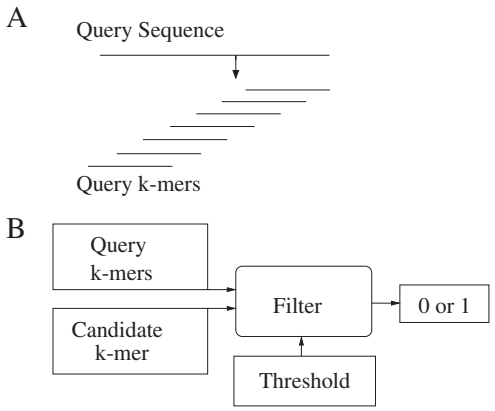

Fig. 2. Filtering k-mers. **(A)** Decomposition of the Query sequence into k-mers. The query sequence is decomposed into overlapping k-mers of a specified length. **(B)** Computation of the similarity of a k-mer to the query. Each filter identifies k-mers that could anchor a binding site, taking as input the k-mers derived from the query sequence, a candidate k-mer, and a pre-specified filter threshold. Each filter then reports whether the candidate k-mer had the specified level of similarity to at least one of the k-mers in the query sequence or not.

Each filter uses a function designed to compare two k-mers. In order to compare a candidate k-mer to a query sequence, we first decompose the query sequence into k-mers of the same length as the candidate k-mer, and then compare the candidate k-mer to each k-mer derived from the query (see Figure 2A). If any of the query derived k-mers meet the specified similarity to the candidate k-mer (Figure 2B), then the candidate k-mer is retained for further analysis. If none of the query derived k-mers meets the specified similarity, then the candidate k-mer is eliminated from further consideration.

The simplest filter—the mismatch filter—eliminates k-mers that differ from every k-mer in the query sequence by more than a specified number of bases. This filter is designed to reject k-mers that have little affinity to any part of the query sequence. The filter function computes the fraction of mismatches between a candidate k-mer K and the query sequence Q:

$$F_1(K,Q) = \max_{j \in s(Q,k)} \sum_{i=1}^{k} \frac{\delta(K_i = j_i)}{k},$$

where $s(Q,k)$ returns the set of all k-mers in Q, and δ is the Kronecker delta function.

The second filter rejects k-mers that contain destabilizing internal mismatches relative to the query. These destabilizing mismatches are identified using thermodynamic data on DNA binding stability (SantaLucia, Jr and Hicks, 2004). This filter's function is similar to

the mismatch filter, except that it takes into consideration the specific stabilities of dinucleotide stacks (pairs of adjacent, paired bases) and single internal mismatches. We implement this filter by encoding each k-mer K as a complex valued vector $\Phi(K)$, and we developed this filter so that the inner product of the conjugate of the encoding of one k-mer and another k-mer approximates the sum of the free energy of binding between the first k-mer and the reverse complement of the second k-mer, and vice versa. Details of this encoding are given in the appendix. The final value of this filter is a normalized dot product:

$$F_2(K,Q) = \max_{j \in s(Q,k)} \frac{\langle \Phi(K), \Phi(j) \rangle}{\sqrt{\langle \Phi(K), \Phi(K) \rangle \langle \Phi(j), \Phi(j) \rangle}}.$$

The third filter rejects k-mers that do not have good sequence agreement with the query, considering the possibility of asymmetric internal loops. For each candidate k-mer, this filter's function considers many alignments with respect to the query sequence, weighting each by the number of matches and the length and topology of loops. Asymmetric internal loops serve to separate regions of sequence agreement, and thus this filter will recognize sequence similarity even when regions of sequence agreement are separated by insertions or deletions in one sequence with respect to the other. We developed this filter function to be a coarse approximation of the partition function for one sequence binding to the reverse complement of the other, and we therefore consider only base pairing (and neglect the detailed thermodynamic reference data on dinucleotide stability) and internal loops of length three or less. In addition, we use loop stability values optimized for this application. The final value of the alignment filter is

$$F_3(K,Q) = \max_{j \in s(Q,k)} \frac{f(K,j)}{\sqrt{f(K,K) \cdot f(j,j)}}.$$

The alignment function $f(\cdot,\cdot)$ is described in the appendix.

The fourth filter applies the partition function model directly. In this step, we compute the binding affinity between the reverse complement of the k-mer and the query sequence. In order to normalize out the binding properties of the query sequence, we divide this binding energy by the binding energy of the k-mer in the reverse complement of the query sequence with the highest affinity to the query sequence. The final value is

$$F_4(K,Q) = \frac{g(\hat{K},Q)}{\max_{j \in s(Q,k)} g(\hat{j},Q)},$$

where a carat denotes reverse complement, and $g(.,.)$ is the partition function model of DNA binding. In practice, this filter is the most stringent and the most computationally complex.

We apply the four filters twice, with two sets of filter thresholds, to get two sets of candidate anchoring k-mers. We use filter thresholds so that the high similarity group of k-mers will be similar with respect to filters F1 and F2, and the high affinity k-mers will be similar with respect to filters F3 and F4. We then locate all occurrences of both candidate sets in the sequence database, and further consider only those locations in the sequence database where there is a k-mer from the high affinity group close to a k-mer from the high similarity group (see Figure 3).

After the four filtering steps, we must efficiently locate all occurrences of the high affinity and high similarity k-mers within the given sequence database. This is accomplished by using a modified suffix array (Gusfield, 1997; Manber and Myers, 1993) to index the database. In a suffix array, pointers to suffixes of a sequence are sorted lexicographically; in our modified suffix array, the pointers are sorted based on comparison of only the first k positions of the suffix, where k is the length of the filtered k-mers. We also build a hash table on the suffix array itself, so that the positions in the suffix array corresponding to a query k-mer can be quickly located (with a computational complexity of $O(k)$ per k-mer lookup). We use this sequence index, consisting of the modified suffix array and the hash table into the suffix array, to rapidly identify all locations where a candidate k-mer from one group occurs close to a candidate k-mer from the other group. These occurrences, along with their flanking sequences, comprise the list of candidate binding sites.

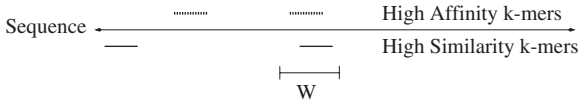

Fig. 3. Search for proximal hits. Our binding site search algorithm finds anchoring k-mers in the search sequence. We use two sets of filter thresholds, and obtain two sets of candidate anchoring k-mers; one set has high similarity to the query, and the other set has high affinity to the query (occurrences of k-mers from the high affinity set are drawn with dashes above the search sequence, and occurrences of k-mers with high similarity are drawn with solid lines below the search sequence). We locate all occurrences of both groups of candidate anchoring k-mers, and further examine only those sites where there is a candidate anchoring k-mer from the high similarity group occurring within a pre-specified distance w from a candidate anchoring k-mer from the high affinity group.

In the final step, each remaining candidate binding site is evaluated by the partition function model for affinity to the query sequence. As we show in Section 4, by using a set of fast, accurate filters, the filtering and indexing stages of the algorithm reduce the sequence search space by three to five orders of magnitude. Therefore, in the final step, we can afford to incorporate a relatively sophisticated, computationally expensive model of DNA stability. Thus, by coupling a pre-filtering step with accurate refinement of the candidate list, we achieve both efficiency and accuracy.

2.3 Choice of filter thresholds

Clearly, the success of our filtering strategy depends to a large extent on the thresholds that we use for each filter. If our thresholds are too stringent, then we risk eliminating true binding partners from our list. Conversely, if our thresholds are not stringent enough, then the efficiency of the search will decrease.

We compute these thresholds empirically by using the partition function model. First, with respect to a given set of experimental conditions and a target level of binding affinity, we scan a sequence database for binding sites to a set of query sequences using the partition function model, storing a list of all binding sites with stability better than a given threshold. We then choose filter parameters conservatively, so that if we re-searched the sequence using our filtering approach, we would obtain all of the binding sites obtained in the slow linear scan.

Our thresholds are set by analyzing the binding sites identified using a linear scan, using the procedure illustrated in Figure 4. We decompose each binding site into its constituent k-mers, as in

Table 1. Query sequences

	Query sequence	Length	GC	ΔG PCR	ΔG MA
1	GAGCTGCGGCAGAGGCTGGCGCCC	24	0.79	−24.5	−36.8
2	GCCTGCACTGGCTTCAGGAAGCTGGAGCC	29	0.65	−25.3	−40.1
3	GGCCAGTTCCTGCAGCCCGAGGC	23	0.74	−21.6	−33.2
4	AGTGGCATGCCTCTCTCTACCCAGC	25	0.60	−19.7	−32.2
5	CCACCAAAAAGTAATTAAAGGGTTTGCCTCAT	32	0.38	−19.5	−35.6
6	CACGCAAATCATCCCCAGCCACATC	25	0.56	−19.1	−31.8
7	CAGGTGTCCCTGCTTCGGCTTCCAG	25	0.64	−20.6	−33.3
8	CGCGAAGTGACCTTCAGAGAGTACGCCAT	29	0.55	−22.3	−37.2
9	CTGGACTGCCAAGTCCAGGGCAGGCC	26	0.69	−23.0	−36.1
10	GTCACCCACCTGCTGGCCCCGG	22	0.77	−20.9	−32.0
11	GGGGCTCAATAAGTCTGCTTCCACCTT	27	0.52	−19.5	−33.0
12	GGGTGAGGCCCATTCATAAGACTGGC	26	0.58	−19.6	−32.7
13	CCAGTCATGTTGCCCCGTTTGTCAGAG	27	0.56	−20.4	−34.1
14	GGGAGGGCTGAAGAGGGCACTCC	23	0.70	−19.4	−30.9
15	GGATGCATATGGACTCTTAGGTGTTCTGCG	30	0.50	−20.6	−36.0
16	GAAAGGGCTGGCTATGATAAACTGTGGC	28	0.50	−19.4	−33.7

ΔG PCR: Free energy of binding, in kilocalories per mole, of the sequence to its reverse complement at 55 C in 50 mM NaCl and 2 mM MgCl2; ΔG MA: Free energy of binding, in kilocalories per mole, of the sequence to its reverse complement at 40 C in 1 M NaCl. Energies are computed using the HYBRID software.

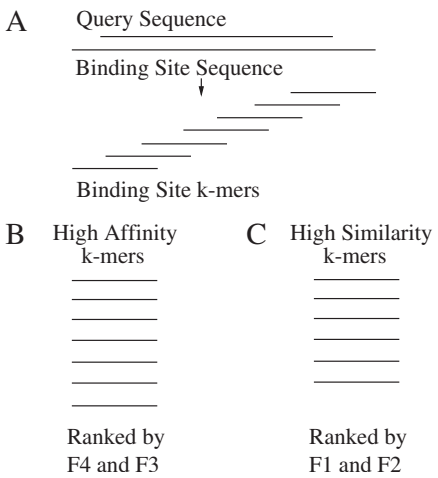

Fig. 4. Filter Analysis of a binding site. **(A)** Decomposition of binding site. Each binding site is decomposed into its constituent *k*-mers. **(B)** Ranking of binding site *k*-mers according to thermodynamic affinity. The binding site *k*-mers are ranked according to similarity to the query by F4; F3 is used to break ties. The similarity scores of the top ranked *k*-mer are added to the set of similarity scores used to determine filter thresholds for the high affinity group of candidate *k*-mer binding site anchors. **(C)** Ranking of binding site *k*-mers according to sequence similarity. All *k*-mers, except the top ranked *k*-mer in (**B**) are re-ranked according to sequence similarity to the query by F1; F2 is used to break ties. The similarity scores of the top ranked *k*-mer are added to the set of similarity scores used to determine filter thresholds for the high similarity group of candidate *k*-mer binding site anchors.

Figure 4(A). We then rank these *k*-mers according to similarity to the query sequence using the filter functions. For the high affinity group of filter parameters, we rank first by filter function F4 and break ties using filter function F3 as shown in Figure 4(B). For the high similarity group of filter parameters, we remove the top ranked *k*-mer and re-rank the remaining *k*-mers using the filter function F1; we break ties with filter function F2 as shown in Figure 4(C). We then compute the similarity of both top ranked *k*-mer to the query according to all filter functions.

After analyzing each binding site recovered from the linear scans, we integrate information from all binding sites as follows. Each binding site contributes two sets of similarity scores, one set for the top *k*-mer ranked according to thermodynamic affinity and one set for the top *k*-mer ranked according to sequence similarity. We accumulate all sets of similarity scores into two sets. One set contains the similarity scores for all top ranked *k*-mers according to thermodynamic affinity, and the other set contains the similarity scores for all top ranked *k*-mers according to sequence similarity. To obtain the final filter parameters, we find the minimum similarity score in a set over all binding sites for each filter function. This is thus a conservative method for obtaining filter parameters, and ensures that if the sequence were re-searched with our filtering approach, we would recover all of the binding sites identified with the linear scan.

Intuitively, the two sets of filter thresholds capture different characteristics of DNA binding: sequence agreement and *k*-mer binding affinity. These two notions of similarity are not the same: consider a query sequence that consists of several A bases followed by several G bases. A *k*-mer consisting entirely of A bases would have perfect sequence agreement to the left part the the sequence, whereas a *k*-mer consisting of all G bases with two consecutive internal A bases would have poor sequence agreement, but the reverse complement of that *k*-mer would have much higher binding affinity to the query sequence than the sequence consisting of all A bases. Our double filtering approach accounts for both situations.

3 METHODS

For validation purposes, we focus on the ENCODE regions of the human genome ENCODE Project Consortium (2004). These 44 regions together

Table 2. Rejection rates for the four filters

Sequence	F1	F2	F3	F4	Remaining
1	99.4	74.3	3.8	0.6	1589
2	99.2	69.0	7.3	2.8	2431
3	99.4	63.8	6.4	10.0	1862
4	99.3	54.0	2.9	24.0	2333
5	99.1	63.0	1.1	48.0	1887
6	99.3	51.7	3.1	29.7	2282
7	99.3	63.8	5.1	2.5	2322
8	99.2	65.0	7.3	14.5	2414
9	99.3	66.5	6.9	10.0	2059
10	99.5	58.5	2.9	13.6	1969
11	99.3	59.8	6.5	26.6	2164
12	99.3	63.2	5.7	47.1	1360
13	99.3	63.7	3.9	20.2	2181
14	99.4	75.4	2.8	2.7	1419
15	99.1	67.9	8.0	12.1	2375
16	99.2	68.5	4.0	47.4	1316
mean	99.3	64.3	4.9	19.5	1998

The table lists, for each of the query sequences in Table 1, the percentage of *k*-mers rejected by each of the four filters using the high affinityfilter thresholds, as well as the total number of *k*-mers that pass through all four filters. These results are for weak binding sites in standard PCR conditions.

Table 3. *k*-mer filtering performance

Sequence	PCR		Microarray	
	Weak	Medium	Weak	Medium
1	1589	15	15	15
2	2431	20	20	20
3	1862	14	14	14
4	2333	16	16	16
5	1887	19	20	16
6	2282	16	16	16
7	2322	16	16	16
8	2414	20	20	20
9	2059	17	17	17
10	1969	13	13	13
11	2164	18	18	18
12	1360	15	17	14
13	2181	18	18	18
14	1419	14	14	14
15	2375	21	21	21
16	1316	16	17	13
mean	1997.7	16.8	17.0	16.3

The table lists, for each of the query sequences in Table 1, the total number of *k*-mers that pass through all four filters using the high affinity thresholds.

Table 4. Proximity filtering performance

Sequence	PCR		Microarray	
	Weak	Medium	Weak	Medium
1	69	88	66	94
2	55	71	50	57
3	59	77	71	83
4	69	80	70	94
5	76	93	91	92
6	57	67	63	74
7	57	48	48	64
8	80	98	81	98
9	54	70	62	82
10	58	72	59	91
11	61	65	64	73
12	63	74	76	85
13	68	94	80	97
14	48	69	56	89
15	74	89	79	95
16	63	85	79	88
mean	63.19	77.50	68.44	84.75

The table lists, for each of the query sequences in Table 1, the percentage of sequence locations that are rejected by the proximity filtering step. The final row contains the column average.

comprise 1% of the human genome. The ENCODE regions were chosen to be representative of the entire genome, based on gene density, GC content, and density of conserved non-coding elements.

In addition, we chose a collection of sixteen query sequences to use in our experiments. We manually selected from the ENCODE regions exonic and intronic sequences that vary in length from 22 to 31 bases. Each selected sequence was analyzed using HYBRID, assuming standard PCR conditions (see below). A selected sequence *S* was added to the query set if the binding affinity between *S* and its reverse complement is greater than -19 kilocalories per mole. None of the selected query sequences overlaps a repeat sequence as annotated by RepeatMasker, and the percent GC of the queries range from 40% to 80%. The final list of query sequences is given in Table 1.

To generate a gold standard set of binding sites, we used HYBRID to scan every base of both strands of the ENCODE regions. The scan employed a window size of 35 bases, and was repeated with two different sets of experimental conditions, typical of PCRs and microarray experiments, respectively. For PCR conditions, we predict binding affinities at 55 C, with a concentration of 50 millimolar NaCl and 2 millimolar $MgCl_2$. For microarray conditions, we predict binding affinities at 40 C, with a concentration of 1 molar NaCl. In subsequent experiments, we used these lists of binding sites to verify that our algorithm correctly identifies all binding sites.

In selecting filter thresholds, we focus on two levels of binding site stringency, corresponding to weak and medium binding. We define a weak binding site as one where the equilibrium constant of the dimer formed by the binding site and the query sequence is at most six orders of magnitude less than the dimer formed by the query sequence binding to its reverse complement, under equal initial single strand concentrations. We define medium binding sites similarly, except we require only three orders of magnitude of difference. We used all binding sites recovered with the linear scans to choose filter thresholds.

4 RESULTS

In order to measure the efficiency and accuracy of our binding site prediction algorithm, we scan the ENCODE regions with a collection of query sequences, using HYBRID with and without the filtering and indexing pipeline. This experiment shows that our approach yields a significant improvement in running time, without missing any binding sites.

We begin by examining the behavior of each of the four filters for the thresholds designed to detect *k*-mers with high thermodynamic affinity to the query. Table 2 lists the percent of *k*-mers eliminated by the combined filters for each of the 16 query sequences. The mismatch kernel appears to provide the most value, since it has a

Table 5. Number of candidate sequences examined and accepted by the partition function model of DNA binding, and time for each run

Sequence	Weak PCR			Medium PCR			Weak microarray			Medium microarray		
	Candidates	Actual	Time	Candidates	Actual	Time	Candidates	Actual	Time	Candidates	Actual	Time
1	30712	25	6 m	2340	19	18 s	9543	21	35 s	1332	16	23 s
2	57587	23	20 m	11994	15	40 s	18702	16	76 s	3326	11	19 s
3	44628	100	8 m	4030	20	21 s	4882	21	17 s	3078	17	19 s
4	35218	45	11 m	5269	19	22 s	6152	20	24 s	1178	16	12 s
5	23235	29	6 m	1870	13	18 s	2791	14	23 s	1132	9	12 s
6	91220	108	13 m	7304	19	26 s	8112	21	28 s	6301	16	21 s
7	33780	48	10 m	6667	20	31 s	6667	22	24 s	4962	15	28 s
8	22310	26	5 m	179	14	10 s	3025	15	19 s	99	10	12 s
9	35396	45	12 m	6552	18	22 s	8390	19	35 s	4264	14	17 s
10	175109	336	12 m	5741	21	17 s	7976	25	29 s	1909	18	11 s
11	75547	40	16 m	5908	17	24 s	6181	18	32 s	4896	14	28 s
12	20887	70	6 m	3120	18	16 s	3369	20	19 s	1598	14	14 s
13	22934	31	6 m	907	19	14 s	3276	20	16 s	418	14	13 s
14	142717	361	13 m	5221	21	22 s	7081	37	34 s	1837	17	13 s
15	20106	26	8 m	1413	14	13 s	2839	16	17 s	153	10	12 s
16	17138	29	6 m	2988	17	17 s	3680	19	19 s	1639	13	17 s
mean	53032.8	83.9	9.9 m	4468.9	17.8	20.7 s	6416.6	20.3	27.9 s	2382.6	14.0	16.9 s

The table lists, for each experiment, the total number of candidate sites produced by the filtering and indexing pipeline, the number of those sites that are considered by HYBRID to be true binding sites, and the total wall clock time required to identify the sites.

rejection rate over 99%; however, this high rejection rate is primarily a result of its placement first in the filter pipeline. In practice, the more computationally expensive filters are also more exclusive. In each case, the filters reduce the complete set of $4^{10} = 1,048,576$ k-mers to less than 2500 k-mers. Also, note that the setup in Table 2 (weak binding sites in PCR conditions) is the most permissive and hence yields a relatively large number of k-mers. Table 3 lists the total number of k-mers that successfully pass through all four filters in each experiment: strong and weak binding, and PCR and microarray conditions. With the exception of the weak binding/PCR conditions, the algorithm typically produces on the order of 20 k-mers for further consideration. The results in Tables 2 and 3 use the filter thresholds selected using the high affinity filter parameters. Results for the high similarity set of thresholds are similar.

After obtaining both groups of candidate binding site anchors, we then identify locations in the sequence where a k-mer from the high affinity group occurs near a k-mer from the high similarity group. Table 4 lists, for all four experiments, the percentage of sites identified by the high affinity group of binding site anchor candidates that are not close enough to a k-mer occurrence from the high similarity group of binding site anchor candidates. On average, this step reduces the list of candidate sites by between 63% and 85%, depending upon the experiment.

The final stage of the analysis involves running HYBRID on the filtered list of candidate binding sites. Table 5 lists, for each experiment, the number of candidate binding sites that were evaluated by the HYBRID software. Clearly, this stage is very important, since the number of sites considered is typically several orders of magnitude larger than the number of sites that HYBRID identifies as binding partners. In this sense, our filters are conservative: they do not very closely approximate the computation performed by HYBRID. However, these conservative thresholds lead to high accuracy. For all 16 primers that we tested, our filtering and indexing pipeline identifies 100% of the binding sites that were identified by HYBRID in the much more computationally expensive linear scan of the entire ENCODE regions. Furthermore, as shown in Table 5, the entire pipeline is very efficient. For medium binding strength and standard PCR conditions, HYBRID was only required to evaluate an average of 4467 sites, and scanning the entire ENCODE database required 20.7 seconds on average. By comparison, a linear scan of the ENCODE regions using HYBRID takes approximately two days.

5 DISCUSSION

We have presented a method for rapidly identifying binding partners for a given query DNA sequence within a genome-sized DNA database. Our approach combines a k-mer filtering method, which identifies k-mers that could nucleate binding sites to the query, with an efficient indexing method, which rapidly locates these nucleating k-mers in a sequence database. The combination of these two methods speeds up the DNA binding site search by at least three orders of magnitude.

We note that not all predicted binding sites will be relevant to every hybridization reaction. Some dimers may be slow to reach equilibrium concentrations, especially if the dimer has internal loops. Thus, in a PCR, some dimers may not have time to form and thus may not be a problem. However, in microarray hybridization experiments, conditions are much closer to equilibrium, and secondary binding sites may be more of a concern.

Among the four tasks that we considered, finding weak binding partners for PCR primers is the most difficult search task, and the one for which we obtain the least improvement. However, this task may be the most important for experimentalists, because even weak binding sites can drive high yields on undesired background reactions. This is because in PCR, the primers are present in vast excess,

and the excess concentration of primer in the initial stages of the reaction drives high levels of weak binding site occupancy, even though the binding affinity is low.

The major bottleneck in our method is evaluating the final list of sequences. Even though we reduce the number of sequences that must be considered by several orders of magnitude, the partition function model is still sufficiently slow that it introduces a significant computational burden. It is important to recognize, however, that we can typically place an upper limit on this burden: once we identify a pre-specified number of binding partners for a given query, the search can terminate, since that particular query is not a tenable primer or probe candidate.

6 FUTURE WORK

Conceptually, searching a RNA database for binding sites to a RNA sequence is similar to the problem addressed in this paper. Although the same partition function model can be used to compute the binding affinity of one RNA molecule for another, the parameters are different due to the chemical differences between RNA and DNA Mathews *et al*. (1999). We are currently beginning experiments to evaluate the computational complexity of this version of the binding site search problem. Further, it may also be of interest to search for DNA binding partners of an RNA molecule, or RNA binding partners for a DNA molecule. Because the data for these heterogeneous dimers is much less complete than the data for DNA/DNA or RNA/RNA dimers, our method is not applicable to these binding site searches.

Our method depends critically on the filter parameters, and clearly the similarity of the anchoring k-mers in a binding site to a query is not known in advance. We are therefore increasing the size of our database of predicted binding sites, so that we can estimate the sensitivity of our method for a wider variety of query sequences.

7 CONCLUSIONS

We have shown that DNA binding site search of genomic scale DNA sequences is tractable for realistic experimental conditions, for primer length DNA sequences. Our filters work together to reduce by at least three orders of magnitude the number of sequences that must be examined by a partition function model of DNA binding, reducing search time from two days to scan the ENCODE regions to under a minute for typical queries. This filter- and index-based method will be useful in the design of PCR primers and short oligonucleotide probes.

ACKNOWLEDGEMENTS

This work was funded by NIH awards R33 HG003070, T32 HG00035 and R01 GM071923.

REFERENCES

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. 215: 403–410, 1990.

A. Andersson, R. Bernander, and P. Nilsson. Dual-genome primer design for construction of DNA microarrays. *Bioinformatics*, 21(3): 325–332, 2005.

Q. Chou, M. Russell, D. E. Birch, J. Raymond, and W. Bloch. Prevention of pre-PCR mis-priming and primer dimerization improves low-copy-number amplifications. 20(7):1717 –1723, 1992.

R. A. Dimitrov and M. Zuker. Prediction of hybridization and melting for double stranded nucleic acids. *Biophys. Journal*, 87(1): 215–226,2004 .

ENCODE Project Consortium. The ENCODE (ENcyclopedia of DNA Elements) project. *Science*, 306: 636–640, 2004.

T. Garel and H. Orland. Generalized Poland-Scheraga model for DNA hybridization. *Biopolymers*, 75(6): 453–467, 2004.

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

S. A. Haas, M. Hild, A. P. H. Wright, T. Hain, D. Talibi, and M. Vingron. Genome-scale design of PCR primers and long oligomers for DNA microarrays. *Nucleic Acids Res*., 31(19): 5576–5581, 2003.

M. A. Innis, D. H. Gelfand, and J. J. Sninsky. *PCR Applications: Protocols for Functional Genomics*. Academic Press, 1999.

B. Kaltenboeck and C.M. Wang. Advances in real-time PCR: Application to clinical laboratory diagnostics. *Adv. in Clin. Chem*., 40:219–259, 2005.

W.J. Kent, C. W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, and D. Haussler. The human genome browser at UCSC. *Genome Res*., 12(6): 996–1006, 2002.

T. Lowe, J. Sharefkin, S. Q. Yang, and C. W. Dieffenbach. A computer program for selection of oligonucleotide primers for polymerase chain reactions. *Nucleic Acids Res*., 18(7):1757 –1761, 1990.

U. Manber and E. Myers. Suffix arrays:a new method for on-line search. *SIAM J. Comput*., 2: 935–948, 1993.

N. Markham and M. Zuker. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acid Res*., 33: W577–W581, 2004.

D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol*., 288: 911–940, 1999.

R. K. Saiki, D. H. Gelfand, S. Stoffel, S.J. Scharf, R. Higuchi, G. T. Horn, K. B. Mullis, and H.A.Erlich. Primer-directed enzymatic amplification of DNA with a thermostable polymerase. *Science*, 239(4839): 487–491, 1988.

J. SantaLucia, Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl Acad. Sci. USA*, 95: 1460–1465, 1998.

J. SantaLucia, Jr and D. Hicks. The thermodynamics of DNA structural motifs. *Annu. Rev. Biophys. Biomol. Struct*., 33: 415–440, 2004.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. 270: 467–470, 1995.

R. B. Stoughton. Applications of DNA microarrays in biology. *Annual Rev. Biochem*., 74: 53–82, 2005.

X. Wang and B. Seed. A PCR primer bank for quantitative gene expression analysis. *Nucleic Acids Res*., 31(24): e154, 2003.

M.L. Wong and J.F. Medrano. Real-time PCR for mRNA quantitation. *Biotech*., 39: 75–85, 2005.

W. Xu, W. J. Briggs, J. Padolina, W. Liu, C. R. Linder, and D. P. Miranker. Using MoBioS' scalable genome join to find conserved primer pair candidates between two genomes. *Bioinformatics*, 20: i355–i362, 2004.

N. Ben Zakour, M. Gautier, R. Andonov, D. Lavenier, P. Veber M-F. Cochet, A. Sorokin, and Y. Le Loir. GenoFrag: software to design primers optimized for whole genome scanning by long-range PCR amplification. *Nucleic Acids Res*., 32(1): 17–24, 2004.

APPENDIX: FILTERS

We use four filters. The simplest counts the number of mismatches between two k-mers, and the most complicated computes the binding energy of the reverse complement of a k-mer binding to the query according to the partition function model. The other two filters are described in the next two subsections. We use A and B to represent the sequences input to the filter; these sequences have length m and n, respectively. We use A_i to represent the ith element of sequence A.

Free energy filter

The free energy filter is defined first by mapping sequences A and B to complex valued vectors $\Phi(A)$ and $\Phi(B)$, and then taking their inner product. We developed the mapping Φ and present it here for the first time.

The mapping function has the property that if A and B are identical, then

$$\langle \Phi(A)^*, \Phi(B) \rangle = \Delta G(A, \hat{B}) + \Delta G(B, \hat{A}) \text{-} \Delta G_i$$

where a carat denotes reverse complement, and $\Delta G(A, \hat{B})$ is the free energy of binding of A to the reverse complement of B, and ΔG_i is a duplex initiation energy parameter. This computation of the binding energy between two sequences approximates the free energy computations presented in SantaLucia, Jr and Hicks(2004).

The inner product $\langle \Phi(A), \Phi(B) \rangle$ has the property that the angle between $\Phi(A)$ and $\Phi(B)$ increases with the number of mismatches. The angle is also sensitive to the identity of the mismatching bases, and will increase more for strongly destabilizing mismatches (such as C—C) than for mildly destabilizing mismatches (such as G—G).

The inner product can be computed as

$$\langle \Phi(A), \Phi(B) \rangle = \sum_{k=1}^{m-1} [\Delta G_s(A_k, A_{k+1}) \Delta G_s(B_k, B_{k+1})] + [\delta(A_k = B_k) \delta(A_{k+1} = B_{k+1})]$$

where $\Delta G_s(A_k, A_{k+1})$is the free energy of binding of the dinucleotide stack(SantaLucia, Jr and Hicks, 2004).

Alignment filter

We designed the alignment filter to coarsely approximate the partition function model of DNA binding. This filter computes a score that rewards runs of consecutive identical bases in each sequence, and that penalizes loops analogously to the loop entropy functions in(SantaLucia, Jr and Hicks, 2004). The parameters that we use to reward consecutive matches and penalize loops were optimized for this application.

Table 6. The loop penalty matrix

1.050	0.120	0.010
0.120	0.800	0.003
0.010	0.003	0.003

The filter value is computed first by filling a dynamic programming matrix, and then computing the sum of all of its entries. This filter uses an AT reward parameter α, and a GC reward parameter β. We set $\alpha = 1.1$ and $\beta = 1.15$. This is analogous to assigning a slightly more stable energy to GC base pairs than AT base pairs, but this filter neglects specific dinucleotide effects.

The dynamic programming matrix is filled in as follows. If A_i is not equal to B_j, then $F_{i,j}$ is set to zero. Otherwise, if A_i is equal to B_j, then

$$F_{i,j} = \max_{i-3 \le x < i,\, j-3 \le y < j} (R * F_{x,y} * L[i-x, j-y])$$

where $R = \alpha$ if A_i and B_j are both A or T, and $R = \beta$ otherwise. The loop penalty matrix L is given in Table 6. The element in the first row and column is greater than 1 in order to reward consecutive matches.