Finding Motifs Using Random Projections

Jeremy Buhler and Martin Tompa
Department of Computer Science and Engineering
Box 352350
University of Washington
Seattle, WA 98195-2350 USA
{jbuhler,tompa}@cs.washington.edu

ABSTRACT

Pevzner and Sze [23] considered a precise version of the motif discovery problem and simultaneously issued an algorithmic challenge: find a motif M of length 15, where each planted instance differs from M in 4 positions. Whereas previous algorithms all failed to solve this (15,4)-motif problem, Pevzner and Sze introduced algorithms that succeeded. However, their algorithms failed to solve the considerably more difficult (14,4)-, (16,5)-, and (18,6)-motif problems.

We introduce a novel motif discovery algorithm based on the use of *random projections* of the input's substrings. Experiments on simulated data demonstrate that this algorithm performs better than existing algorithms and, in particular, typically solves the difficult (14,4)-, (16,5)-, and (18,6)-motif problems quite efficiently. A probabilistic estimate shows that the small values of d for which the algorithm fails to recover the planted (l, d)-motif are in all likelihood inherently impossible to solve. We also present experimental results on realistic biological data by identifying ribosome binding sites in prokaryotes as well as a number of known transcriptional regulatory motifs in eukaryotes.

1. CHALLENGING MOTIF PROBLEMS

Pevzner and Sze [23] considered a very precise version of the motif discovery problem of computational biology, which had also been considered by Sagot [26]. Based on this formulation, they issued an algorithmic challenge:

Planted (l, d)-Motif Problem: Suppose there is a fixed but unknown nucleotide sequence M (the *motif*) of length l. The problem is to determine M, given t nucleotide sequences each of length n, and each containing a planted variant of M. More precisely, each such planted variant is a substring that is M with exactly d point substitutions.

One instantiation that they labeled "The Challenge Problem" was parameterized as finding a planted (15,4)-motif in $t = 20$ sequences each of length $n = 600$. These values of n, t, and l are typical for such motif discovery problems as finding transcription factor binding sites in a collection of coregulated gene promoter regions in yeast.

RECOMB 2001, Montreal, Canada

A number of algorithms to find motifs have been proposed previously, for example, Bailey and Elkan [2], Hertz and Stormo [13], Lawrence *et al.* [16], Lawrence and Reilly [17], and Rocke and Tompa [25]. Since these algorithms employ some form of local search such as Gibbs sampling, expectation maximization, or the greedy method, each may end in a local optimum rather than finding the best motif. Indeed, Pevzner and Sze [23] discovered that CONSENSUS [13], Gibbs sampling [16], and MEME [2] all performed poorly on the particular planted (15,4)-motif challenge problem stated above, usually ending at local optima representing randomly occurring patterns rather than the superior planted motif.

In addition to the algorithms mentioned above, there are also a number of motif-finding algorithms, based on exhaustive enumeration of the possible motifs M, that are guaranteed to find the optimal motif. (See, for example, Blanchette *et al.* [4], Brāzma *et al.* [6], Galas *et al.* [10], Sagot [26], Sinha and Tompa [27], Staden [28], Tompa [29], and van Helden *et al.* [30].) However, these enumerative algorithms run in time exponential in the motif length l and become impractical for the sizes involved in the challenge problem. Other motif discovery algorithms have been proposed by Fraenkel *et al.* [9] and Rigoutsos and Floratos [24].

Pevzner and Sze [23] introduced two novel algorithms, WINNOWER and SP-STAR, both of which succeeded in solving the planted (15,4)-motif challenge problem. In summary, WINNOWER constructs a graph whose vertices correspond to the l-mers present in the t input sequences, with an edge connecting two vertices if and only if the corresponding l-mers differ in at most $2d$ positions and do not both come from the same input sequence. WINNOWER then looks for a clique of size t in this graph. Their second algorithm, SP-STAR, starts in turn from each individual l-mer x in the input, chooses the closest match to x from every other input sequence, and uses a sum-of-pairs score and iterative refinement to converge on a good motif.

The planted (15,4)-motif problem is not the most challenging for $t = 20$ and $n = 600$. For example, the planted (14,4)-, (16,5)-, and (18,6)-motif problems are all considerably more difficult. Both WINNOWER and SP-STAR fail to find these motifs. (See Table 1.)

We introduce an algorithm called PROJECTION that is based on the use of *random projections* of the input's l-mers, an idea very differ-

ent from all the motif discovery algorithms listed above. The key idea in this algorithm is to choose k of the l positions at random, then use the k selected positions of each l-mer x as a hash function $h(x)$. (This idea is derived from "locality-sensitive hashing," employed in the context of computational geometry by Indyk and Motwani [14], in databases by Gionis *et al.* [11], and in computational biology by Buhler [7]. A different randomized projection algorithm was used by Linial *et al.* [19] to cluster proteins.) When a sufficient number of l-mers hash to the same bucket, they are likely to be enriched for the planted motif M.

Experiments demonstrate that PROJECTION performs better than all the algorithms above on planted motif problems. In our experiments on randomly generated input sequences, PROJECTION typically solved the difficult planted (14,4)-, (16,5)-, and (18,6)-motif problems. Even on these difficult problems the longest successful PROJECTION run took approximately one hour, while many runs required only minutes. PROJECTION solves Pevzner and Sze's planted (15,4)-motif challenge problem in under two minutes. Another advantage over WINNOWER and SP-STAR is that PROJECTION is a probabilistic algorithm. This means that if the user is willing to run more iterations on a given input, the probability of discovering the planted motif (as well as other existing motifs) increases.

A probabilistic analysis, given in Section 3.2, shows (again for Pevzner and Sze's parameters $t = 20$ and $n = 600$) that those small values of d for which PROJECTION fails to recover the planted (l, d)-motif are in all likelihood inherently impossible to solve. Specifically, problem instances with these parameters are likely to contain spurious motifs that are as strong as the planted motif. For example, 20 random sequences each of length 600 (with no planted motif) are expected to contain at least one (9,2)-motif by chance, whereas the expected number of (10,2)-motifs that they contain is approximately 10^{-7}. Similar statements can be made for (11,3)- vs. (12,3)-motifs, (13,4)- vs. (14,4)-motifs, (15,5)- vs. (16,5)-motifs, and (17,6)- vs. (18,6)-motifs. Thus, there is a rather sharp line between those planted motif problems that PROJECTION solves, and those that inherently cannot be solved.

After describing the algorithm in Section 2, we present experimental results in Section 3. In Section 3.1 we first present the natural experiments for the planted motif problem as described above. Namely, simulated data is used in which (1) the motif M is chosen randomly; (2) the t independent planted instances are each produced by randomly selecting (without replacement) d positions in M, randomly changing each of those positions to some other nucleotide, and randomly selecting the position of this planted instance in its input sequence; and (3) the remaining $n - l$ residues of each input sequence are chosen randomly. In this simulated data, all random choices are made uniformly and independently. This corresponds to the "FM model" used in most of the experiments reported by Pevzner and Sze [23].

In Sections 3.3 and 3.4 we complement these simulated results, addressing realistic biological data first by identifying transcription factor binding sites in the promoter regions of eukaryotic genes, then by tackling the ribosome binding site problem in prokaryotes. In these experiments, the background nucleotide distribution differs substantially from the independent random model. Our examples of promoter regions contain just a few motif instances in an equal number of sequences, while the ribosome binding site examples consist of thousands of nucleotide sequences, only a fraction of which contain the motif. We validate these experimental results by comparing them to published sites from the literature and, for the ribosome binding sites, to their complementary 16S rRNA sequences.

2. THE PROJECTION ALGORITHM

Like many probabilistic algorithms, PROJECTION performs a number of independent trials of a basic iterant. In each such trial, it chooses a random projection h and hashes each l-mer x in the input sequences to its bucket $h(x)$. Any hash bucket with sufficiently many entries is explored as a source of the planted motif, using a series of refinement steps as described in Section 2.2.

2.1 Random Projections

As outlined in Section 1, the hash function $h(x)$ is constructed by choosing k of the l positions at random, without replacement, for a value of k to be determined later. If x is an l-mer, then $h(x)$ is simply the k-mer that results from concatenating the selected k residues of x. Viewing x as a point in an l-dimensional Hamming space, $h(x)$ is the projection of x onto a k-dimensional subspace.

If M is the (unknown) planted motif, we will call the bucket with hash value $h(M)$ the *planted bucket*. The fundamental intuition underlying PROJECTION is that, if $k < l - d$, there is a good chance that a number of the t planted instances of M will hash together into the planted bucket. In particular, those planted instances for which the d mutated positions are disjoint from the k hash positions will hash to the planted bucket. At the same time, if k is not too small, it is unlikely that many random l-mers from the input will hash to the planted bucket, because they must agree with M in all k chosen positions. Thus, there is a good chance that the l-mers in the planted bucket will be highly enriched for the planted motif, enabling the algorithm to recover it from the bucket and the input sequences.

Of course, the algorithm does not know which the planted bucket is, and so attempts to recover the motif from every bucket that contains at least s elements, for a parameter s to be chosen later. On occasion, PROJECTION actually succeeds in recovering the correct motif by refining some bucket other than the planted bucket, which is an added bonus.

The PROJECTION algorithm requires the choice of three key parameters, namely the projection size k, the bucket threshold s, and the number of independent trials to run. We first attempt to minimize contamination of the planted bucket by random background sequences. Since we are hashing $t(n - l + 1)$ l-mers into 4^k buckets, if $4^k > t(n - l + 1)$ the average bucket will contain less than one random l-mer. For the simulated challenge problems of Section 3.1 and the promoter examples of Section 3.3, we can choose k large enough to satisfy this low-noise condition without violating the constraint that $k < l - d$. These examples contain only a small number (4-20) of motif instances overall, so we cannot expect too many instances to hash to the same bucket in a reasonable number of trials. We therefore set a low bucket size threshold s (3-4), keeping in mind that even for such small thresholds, the planted bucket is expected to contain more motif instances than random sequences.

In the ribosome binding site examples of Section 3.4, the total amount of sequence is so large that we cannot choose k to simultaneously satisfy $k < l - d$ and $4^k > t(n - l + 1)$. However, we believe that the planted motif is quite frequent in the input and so are no longer bound to limit s to a small integer. For these examples, we set $k = l - d - 1$, as large as possible, and select for buckets

in which motif instances are at least as common as background instances, that is, buckets with more signal than noise. In particular, we set s to be twice the average bucket size $t(n-l+1)/4^k$.

Finally, we come to the determination of the number m of independent trials to run. We choose m so that the probability is at least $q = 0.95$ that the planted bucket contains s or more planted motif instances in at least one of the m trials. For the chosen value of q, PROJECTION often encounters several trials in which the planted bucket is significantly enriched for the motif. This condition leads to the following derivation of m.

Let $\hat{p}(l,d,k)$ be the probability that a given planted motif instance hashes to the planted bucket, that is,

$$\hat{p}(l,d,k) = \frac{\binom{l-d}{k}}{\binom{l}{k}}.$$

Let $\hat{t}$ be an estimate of the number of input sequences containing a planted motif instance ($\hat{t} = t$ for the simulated challenge problems and promoter examples, and $\hat{t} = t/3$ for the ribosome binding site application). Then the probability that fewer than s planted instances hash to the planted bucket in a given trial is $B_{\hat{t},\hat{p}(l,d,k)}(s)$, where $B_{\hat{t},p}(s)$ is the probability that there are fewer than s successes in $\hat{t}$ independent Bernoulli trials, each trial having probability p of success. If PROJECTION is run for m trials, the probability that s or more planted instances hash to the planted bucket in at least one trial is

$$1 - \left[B_{\hat{t},\hat{p}(l,d,k)}(s)\right]^m \geq q.$$

In order to satisfy this inequality, choose

$$m = \left\lceil \frac{\log(1-q)}{\log(B_{\hat{t},\hat{p}(l,d,k)}(s))} \right\rceil. \qquad (1)$$

Using this criterion for m, we find that our choices for k and s above require at most thousands of trials, and usually many fewer, to produce a bucket containing sufficiently many instances of the planted motif.

2.2 Motif Refinement

In this section we describe how each bucket with at least s elements is explored to recover the planted motif. The idea is that, if the current bucket is the planted bucket, PROJECTION has already discovered k of the planted motif residues. These residues, plus the information in the remaining $l-k$ positions of those l-mers that hashed to the current bucket, provide a very strong signal, starting from which a few iterations of refinement should lead to the motif.

Our primary tool for refining candidate motifs is expectation maximization (EM), as formulated for the motif finding problem by Lawrence and Reilly [17]. This EM formulation derives from the following simplified probabilistic model. An instance of some length-l motif occurs exactly once per input sequence. Motif instances are generated from a $4 \times l$ weight matrix model W whose (i,j)th entry gives the probability that base i occurs in position j of an instance, independent of its other positions. The remaining $n-l$ residues in each sequence are chosen randomly and independently according to some background nucleotide distribution. Although this model only approximates motifs in real biosequences, it is both efficient and sensitive enough to recover many such motifs in practice. Bailey and Elkan [2] give more accurate motif models, but fitting their parameters from sequence data requires significant additional computation.

Let S be a set of t input sequences, and let P be the background distribution. EM-based refinement seeks a weight matrix model W^* that maximizes the likelihood ratio

$$\frac{\Pr(S \mid W^*, P)}{\Pr(S \mid P)}$$

that is, a motif model that explains the observed sequences much better than the background model alone. The position at which the motif occurs in each sequence is not fixed *a priori*, making computation of W^* difficult because $\Pr(S \mid W^*, P)$ must be summed over all possible locations of the motif instances. To address this difficulty, the core EM algorithm [8] specifies an iterative calculation that, given an initial guess W_0 at the motif model, converges linearly to a *locally* maximum-likelihood model in the neighborhood of W_0.

PROJECTION performs EM refinement on every bucket with at least s instances. We form an initial guess W_h from a bucket h as follows: set $W_h(i,j)$ to be the frequency of base i among the jth positions of all l-mers in h. This guess forms a *centroid* for h, in the sense that positions that are well conserved in h are strongly biased in W_h, while poorly conserved positions are less biased. In order to avoid zero entries in W_h, we add a Laplace correction of b_i to $W_h(i,j)$, where b_i is the background probability of residue i in the input.

Because EM converges only linearly, running it to convergence for every W_h would be computationally prohibitive. Fortunately, just a few iterations of EM (five in our implementation) can significantly improve a well-chosen starting model to the point where it identifies the planted motif. Let W_h^* be the candidate motif model refined from W_h. We use W_h^* to identify the planted motif by selecting from each input sequence the l-mer x with the largest likelihood ratio $\Pr(x \mid W_h^*)/\Pr(x \mid P)$. This multiset T of l-mers represents the motif in the input that is most consistent with W_h^*. Let C_T be the consensus of T, and let $s(T)$ be the number of elements of T whose Hamming distance to C_T exceeds d. In the applications to biological examples in Sections 3.3 and 3.4, PROJECTION outputs C_T for that T that minimizes $s(T)$ over all buckets and all m trials.

To maximize the number of motif instances recovered in the simulated challenge problems in Section 3.1, we perform a further heuristic refinement of T that works well once we have already found a number of planted instances. (This further refinement process is similar to SP-STAR [23] but uses a different score function.) Compute the consensus C of the sequences in T, and define the score of T to be the number of sequences in T whose Hamming distance to C is at most d. Let T' contain the l-mer from each input sequence that is closest in Hamming distance to C. If the score of T' is greater than the score of T, replace T by T' and repeat. This refinement converges after a few iterations. If it converges with a score of t, then report the corresponding consensus as the planted motif. Otherwise, report the consensus of that T that maximizes the score over all buckets and all m trials.

The PROJECTION algorithm is primarily a technique for picking starting motif models that are in the neighborhood of the true

planted motif; it is not strongly tied to the use of a particular refinement strategy such as EM. Good starting points are crucial to many local refinement strategies – including EM, Gibbs sampling, and SP-STAR – because all these techniques may otherwise terminate at local optima different from the planted motif. Existing motif finders, including the algorithm of Bailey and Elkan [2] and SP-STAR, start refinement from every individual l-mer (i.e. every potential motif instance) in the input, in the hope that one of these instances will fall close to the consensus of the planted motif. Unfortunately, the subtle motifs of the challenge problems are formulated precisely so that the planted consensus lies far from any one of its instances! By forming centroids of multiple similar instances, PROJECTION generates starting models that are closer to the actual consensus of the planted motif, so that iterative refinement is more likely to converge to this motif rather than to some inferior local optimum.

3. EXPERIMENTAL RESULTS

3.1 Challenge Problems on Simulated Data

Table 1 compares the performance of PROJECTION with that of previous motif discovery algorithms, using simulated data. The measure used in that table is the performance coefficient of Pevzner and Sze [23], defined as follows. Let K denote the set of $t \cdot l$ residue positions in the t planted motif instances, and let P denote the corresponding set of residue positions in the t instances predicted by the algorithm. Then the *performance coefficient* is defined to be $|K \cap P|/|K \cup P|$.

All input instances consist of $t = 20$ sequences, each of length $n = 600$. Each such random instance was constructed as described at the end of Section 1. For PROJECTION, for each line in Table 1, twenty such random instances were prepared, and the table entry gives the average performance coefficient over those twenty instances. All runs used projection size $k = 7$ and bucket threshold $s = 4$. The number m of iterations shown in the table was computed by Equation (1) of Section 2.1.

The column of Table 1 labeled "Correct" is the number of instances (out of twenty) for which PROJECTION recovered a consensus sequence that was identical to the consensus of the t planted motif instances.

Note that the average performance coefficient of PROJECTION in every line of Table 1 is at least as great as that of any of the previous algorithms. Most striking is the difference in performances on the planted (14,4)-, (16,5)-, (17,5)-, (18,6)-, and (19,6)-motif problems.

As in Pevzner and Sze [23], we also tested PROJECTION on the planted (15,4)-motif problem with input sequences of length $n = 1300$ instead of 600. Again averaging over twenty random instances, the average performance coefficient was 0.88, recovering the planted consensus in all twenty instances. Of all the algorithms reported by Pevzner and Sze, only WINNOWER with parameter $k = 3$ maintains an average performance coefficient of 0.8 at this length. The others are down to 0.23 or less by the time n reaches 1000 [23, Table 1].

3.2 Limitations on Solvable *(l,d)*-Motif Problems

Looking at Table 1, it is natural to ask whether there is a more effective algorithm, in the sense that it recovers planted (9,2)-, (11,3)-, (13,4)-, (15,5)-, or (17,6)-motifs (for $t = 20$ and $n = 600$), whereas PROJECTION fails to do so. A probabilistic analysis shows that these problems are quantitatively different from the problems in Table 1. For instance, 20 random sequences each of length 600 (with no planted motif) are expected to contain more than one (9,2)-motif by chance, whereas the expected number of (10,2)-motifs that they contain is approximately 6.1×10^{-8}. Such estimates are derived as follows. Let

$$p_d = \sum_{i=0}^{d} \binom{l}{i} \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^{l-i}$$

be the probability that a given l-mer C occurs with up to d substitutions at a given position of a random sequence. Then the expected number of length l motifs that occur with up to d substitutions at least once in each of t random length n sequences is approximately

$$E(l,d) = 4^l \left(1 - (1 - p_d)^{n-l+1}\right)^t .$$

(The reason this is only an estimate is that overlapping occurrences of a given motif C do not occur independently.)

Table 2 lists the relevant values of $E(l,d)$ and $E(l+1,d)$ for comparison. Note in each line of the table that the expected number of spurious (l,d)-motifs is around 1–5, whereas the expected number of spurious $(l+1,d)$-motifs is negligible. This means there is likely to be too much random noise to recover a planted (l,d)-motif, for these values of l and d.

Although these table entries are only estimates, we do know from an exhaustive enumeration of 9-mers and an exact calculation of their probabilities of occurrence in 20 random 600-mers that the expected number of spurious (9,2)-motifs is 1.621. (The probability calculation was done using an algorithm described in [29, Section 3.1].) Thus, the estimates may not be too inaccurate.

To corroborate this analysis, PROJECTION was run on twenty random instances of the planted (l,d)-motif problem, exactly as described in Section 3.1. The results are also shown in Table 2. The average performance coefficient is given in the column labeled "apc." The number of instances (out of twenty) on which PROJECTION returned the consensus of the planted motif is shown in the column labeled "Correct." The column labeled "Spurious" shows the number of instances in which PROJECTION returned a perfect (l,d)-motif (i.e. one occurring in all 20 input sequences) that was *not* the planted motif. In the remaining instances the best motif PROJECTION found occurred in only 19 of the 20 input sequences, and was again not the planted motif. These experiments give further evidence that there are many spurious motifs for these values of l and d.

3.3 Transcription Factor Binding Sites

In order to test PROJECTION on more realistic biological data, we used it to find known transcriptional regulatory elements upstream of several eukaryotic genes. We examined orthologous sequences from a variety of organisms taken from regions upstream of four types of gene: preproinsulin, dihydrofolate reductase (DHFR), metallothioneins, and c-*fos*. (See Blanchette [3] for an alternative approach to finding motifs in these sequences.) These sequences are known to contain binding sites for specific transcription factors. We also tested a collection of promoter regions[1] from the yeast *S. cerevisiae* that is known to contain a shared cell-cycle-dependent promoter [20].

[1] *Genes used*: SWI4, CLN3, CDC6, CDC46, and CDC47.

l	d	Gibbs	WINNOWER	SP-STAR	PROJECTION	Correct	m
10	2	0.20	0.78	0.56	0.82	20	72
11	2	0.68	0.90	0.84	0.91	20	16
12	3	0.03	0.75	0.33	0.81	20	259
13	3	0.60	0.92	0.92	0.92	20	62
14	4	0.02	0.02	0.20	0.77	19	647
15	4	0.19	0.92	0.73	0.93	20	172
16	5	0.02	0.03	0.04	0.70	16	1292
17	5	0.28	0.03	0.69	0.93	19	378
18	6	0.03	0.03	0.03	0.74	16	2217
19	6	0.05	0.03	0.40	0.96	20	711

Table 1: Average performance coefficients on planted (l, d)-motifs in simulated data. Each input instance consists of $t = 20$ sequences each of length $n = 600$. Average performance coefficients of Gibbs, WINNOWER ($k = 2$), and SP-STAR are from Pevzner and Sze [personal communication], who averaged the performance coefficient over eight random instances. For PROJECTION, averages were taken over twenty random instances, with projection size $k = 7$ and threshold $s = 4$.

l	d	$E(l, d)$	$E(l+1, d)$	apc	Correct	Spurious	19/20	m
9	2	1.6	6.1×10^{-8}	0.28	11	5	4	1483
11	3	4.7	3.2×10^{-7}	0.026	1	13	6	2443
13	4	5.2	4.2×10^{-7}	0.062	2	15	3	4178
15	5	2.8	2.3×10^{-7}	0.018	0	7	13	6495
17	6	0.88	7.1×10^{-8}	0.022	0	8	12	9272

Table 2: Statistics of spurious (l, d)-motifs in simulated data. The column labeled "apc" shows the average performance coefficient averaged over twenty random instances, each consisting of $t = 20$ sequences of length $n = 600$. The projection size was chosen to be $k = 7$, the bucket threshold was set to $s = 4$, and the number of iterations is shown in the column labeled "m."

Unlike the synthetic examples of Section 3.1, our promoter examples contain background DNA that varies substantially from our simple random model. The embedded motifs are better conserved than in our synthetic examples but must be inferred from a much smaller number of instances. These motifs are also less "subtle" than our synthetic examples; that is, nucleotide substitutions in each motif instance tend to occur at the same few positions, rather than independently at random as in the challenge problem. We were unable to locate published examples of biological motifs as subtle as those of Section 3.1, though we cannot say whether the dearth of such examples implies that they are biologically uncommon, or only that they are inaccessible to previous search techniques.

In all experiments we set $l = 20$ and $d = 2$, which worked well despite the fact that the actual motifs varied considerably in length. We chose a projection size $k = 7$ to obtain less than one expected background l-mer per bucket and set a size threshold $s = 3$, reflecting the fact that the number of motif occurrences in each experiment is quite small (4-5). The numbers of iterations m, chosen according to the procedure of Section 2.1, were also small, so that the experiments required only a few seconds each.

Table 3 gives the best (20,2)-motif consensus found by PROJECTION, along with a published motif that closely matches a substring of each consensus. Analysis of the preproinsulin promoter region yielded a motif known from the TRANSFAC database [31], while the other four experiments produced motifs corresponding to experimentally verified transcription factor binding sites. Multiple refinements often produced two or more distinct motifs with the same (maximum) score; in these cases, the different motifs were usually slightly shifted windows covering the same underlying site. For these experiments, we report only the most $5'$-shifted consensus string.

Although the motifs we found are not particularly subtle, and indeed have previously been found by standard iterative methods such as MEME [3], the results of these experiments are noteworthy for two reasons. First, we achieved good performance even with a fairly primitive refinement strategy that did not include, e.g., motif length corrections, information-based scoring, or iteration of EM to convergence. We expect that random projections could yield even better performance if adjoined to a more sophisticated motif finder that implements these additional techniques. Second, PROJECTION performed fewer total refinements in every case than the standard heuristic that takes every individual l-mer in the input as a starting point for refinement. In our experiments, PROJECTION refined 30% to 80% fewer starting points than this heuristic would have examined. This cost savings is possible because our algorithm preferentially selects starting points that are likely to lead to a successful refinement.

The PROJECTION algorithm returned only a single high-scoring locus in our experiments, even though the input sequences sometimes contained more than one known promoter site. For example, analysis of the preproinsulin locus yielded a site known from TRANSFAC but missed the better known CT-II promoter element [5]. Enhancements to EM refinement, such as probabilistic erasing [2], can ameliorate this method's tendency to prefer only one out of several possible high-scoring motifs.

Simply using less stringent selection criteria for motifs can also identify multiple sites in one experiment, albeit with a significant amount of noise. When we reanalyzed the preproinsulin locus to find high-scoring (14,2)- (rather than (20,2)-) motifs, the result contained multiple hits on the CT-II element and on two TRANSFAC sites, including the one from the previous experiment. However, these three sites together accounted for only 20 of 39 unique high-scoring motifs reported; the other reported motifs did not match

Sequence	Input size (bases)	t	m	Best (20,2) motif from PROJECTION	Published motif	Ref.
preproinsulin	7689	4	15	GGAAATTGCAG*CCTCAGCCC*	CCTCAGCCCC	(A)
DHFR	800	4	15	CTGCA*ATTTCGCGCCA*AACT	ATTTCnnGCCA	(B)
metallothionein	6823	4	15	CCCTC*TGCGCCCGG*ACCGGT	TGCRCYCGG	(C)
c-*fos*	3695	5	8	*CCATATTAG*GACATCTGCGT	CCATATTAGAGACTCT	(D)
yeast ECB	5000	5	8	GTA*TTTCCCGTTTAGGAAA*A	TTtCCcnntnaGGAAA	(E)

Table 3: Performance of PROJECTION on eukaryotic promoter sequences. All motifs were found using $l = 20$, $d = 2$, $k = 7$, and $s = 3$. Where multiple shifted versions of the motif were found, only the most $5'$ result is shown. Italicized portions of the motifs indicate matches to known sequence features. References: (A) TRANSFAC signal [31]; (B) non-TATA transcription start signal [21]; (C) MREa promoter [1]; (D) c-*fos* serum response element [22]; (E) yeast early cell cycle box [20].

these sites, and most could not be explained as shifted windows over one or a few additional underlying loci. Applying PROJECTION with a liberal threshold for accepting motifs can thus reveal more information about interesting sites in the input sequences. However, the results must be further sifted, either by additional refinement or by filtering with a more biologically accurate motif model, to differentiate real motifs from background noise.

3.4 Ribosome Binding Sites

To test PROJECTION's robustness on a very different sort of biological example, we applied it to the ribosome binding site problem for various prokaryotes. Such an instance includes the short DNA sequence ($n = 20$) just upstream of the translation start site of each gene of the organism. The problem is to identify the short site ($l \approx 6$) at which the 16S rRNA of the ribosome binds to the transcribed mRNAs of the organism's genes. It is known that this binding site is complementary to a short subsequence very near the $3'$ end of the 16S rRNA (Kozak [15]), which provides a check for the plausibility of the planted motif that PROJECTION reports.

These instances are quite different from the ones discussed in Sections 3.1 and 3.3: they contain thousands of input sequences, the background nucleotide distribution may be more strongly biased, and the binding site occurs in only a fraction of the input sequences. The results of these experiments are shown in Table 4. For all experiments we chose $l = 6$, $d = 1$, input sequence length $n = 20$, and projection size $k = 4$. The table shows the number t of input sequences, the bucket threshold s, and the number m of iterations, calculated according to the formulas of Section 2.1. The motif predicted by PROJECTION is shown in the column labeled "Motif." Each experiment finished in under one minute.

There are many pieces of evidence that corroborate the ribosome binding site motifs predicted by PROJECTION. The first is the good complementarity of those motifs to the $3'$ end of the 16S rRNA sequences (with the possible exception of *H. influenzae*), as shown in Table 4. Another is the well known fact that, in many bacteria, the binding site for the 16S rRNA during translation initiation is the Shine-Dalgarno sequence AAGGAGG or a large substring of it (Kozak [15], Lewin [18]). The recovered motifs for the four bacteria in Table 4 agree quite well with this fact. In archaea such as *M. jannaschii*, the $3'$ end of the 16S rRNA is missing a few terminal nucleotides compared to the bacterial rRNA sequences, and the 16S rRNA binding site is instead AGGTGAT or a large substring of it (Woese, personal communication). Hayes and Borodovsky [12] discovered the motif GGTGA in *M. jannaschii* using a Gibbs sampler, and Tompa [29] discovered similar binding sites in four different archaeal genomes, including *M. jannaschii*.

Tompa used a very different enumerative statistical algorithm to solve the ribosome binding site problem, ranking motifs by their z-scores. All the motifs found by PROJECTION are in good agreement with the highest scoring motifs that his algorithm reported. For example, the last column of Table 4 shows the 5-mers whose number of occurrences, allowing *no* substitutions, has the highest z-score. Note the strong overlap between each of these entries and the corresponding PROJECTION prediction.

We note that randomization is not strictly necessary to find good starting points for refinement in the ribosome binding site problem. There are only fifteen different projections of a 6-mer into four dimensions, so one could efficiently test all possible projections rather than picking them at random. Indeed, because the embedded motifs are so short, this particular problem has been addressed enumeratively without resorting to iterative search techniques at all [29]. The significance of these results is rather to show that PROJECTION is capable of handling motif-finding problems that are quite different both from the typical applications of Section 3.3 and from the formal motif model for which it was designed.

4. CONCLUSIONS AND OPEN PROBLEMS

We have presented PROJECTION, a new algorithm for finding motifs based on random projections. Experimental results have shown that PROJECTION is much more effective at recovering planted (l, d)-motifs in simulated data than existing algorithms. It has also proven effective in applications to real biological data. As for practicality, the greatest running time on any synthetic instance of Section 3.1 was approximately an hour, and most runs, both on synthetic and on biological examples, required only seconds to minutes.

We intend to improve our implementation of PROJECTION to accommodate more features of real biological motif-finding problems. Basic improvements include predicting the length of the motif, finding multiple motifs in the same input automatically, and handling features such as spacers (sequences of N's) in the motif (Sinha and Tompa [27]). A more challenging research problem is to extend PROJECTION to handle motifs whose instances contain insertions and deletions, which destroy the notion of fixed sequence positions used to define projections. Most importantly, we continue to seek biological examples of motifs that are more subtle than those described in Sections 3.3 and 3.4. Such motifs could better illustrate the particular strengths of the PROJECTION algorithm.

Organism	t	s	m	Motif	Occurs	16S rRNA	Best z-score
M. jannaschii	1679	196	14	AGGTGA	606	GGAGGTGATCC	GGTGA
H. influenzae	1716	202	17	AGGAAA	639	TAAGGAGGTGA	AAGGA
T. maritima	1846	216	13	GGAGGT	1198	GAAAGGAGGTG	AGGTG
B. subtilis	4099	480	35	AGGAGG	2742	TAGAAAGGAGG	AGGAG
E. coli	4287	502	35	AAGGAG	1306	TAAGGAGGTGA	AGGAG

Table 4: Planted (6,1)-motifs found by PROJECTION **as candidate 16S rRNA binding sites of various prokaryotes. For all experiments, the projection size was $k = 4$. The column labeled "Occurs" shows the number of input sequences containing the motif with up to one substitution. The column labeled "16S rRNA" shows the reverse complement of the $3'$ end of the organism's 16S rRNA; the true binding site should be similar to a substring of this sequence. The column labeled "Best z-score" shows the 5-mer with the greatest z-score, using the algorithm of Tompa [29].**

5. ACKNOWLEDGEMENTS

We thank Pavel Pevzner and Sing-Hoi Sze for providing the average performance coefficients given in Table 1 for their algorithms and the Gibbs sampler. Thanks also to Mathieu Blanchette for collecting the orthologous promoter sequences used in Section 3.3. This material is based upon work supported in part by a Fannie and John Hertz Foundation Fellowship and in part by the National Science Foundation under grant DBI-9974498.

6. REFERENCES

[1] R. D. Andersen, S. J. Taplitz, S. Wong, G. Bristol, B. Larkin, and H. R. Herschman. Metal-dependent binding of a factor in vivo to the metal-responsive elements of the metallothionein 1 gene promoter. *Molecular and Cellular Biology*, 7:3574–81, 1987.

[2] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21(1-2):51–80, Oct. 1995.

[3] M. Blanchette. Algorithms for phylogenetic footprinting. In *RECOMB01: Proceedings of the Fifth Annual International Conference on Computational Molecular Biology*, Montreal, Canada, Apr. 2001.

[4] M. Blanchette, B. Schwikowski, and M. Tompa. An exact algorithm to identify motifs in orthologous sequences from multiple species. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 37–45, San Diego, CA, Aug. 2000. AAAI Press.

[5] D. S. W. Boam, A. R. Clark, and K. Docherty. Positive and negative regulation of the human insulin gene by multiple trans-acting factors. *Journal of Biological Chemistry*, 265:8285–96, 1990.

[6] A. Brāzma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements *in silico* on a genomic scale. *Genome Research*, 15:1202–1215, 1998.

[7] J. Buhler. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, 2001. To appear.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[9] Y. M. Fraenkel, Y. Mandel, D. Friedberg, and H. Margalit. Identification of common motifs in unaligned DNA sequences: application to *Escherichia coli* Lrp regulon. *Computer Applications in the Biosciences*, 11(4):379–387, 1995.

[10] D. J. Galas, M. Eggert, and M. S. Waterman. Rigorous pattern-recognition methods for DNA sequences: Analysis of promoter sequences from Escherichia coli. *Journal of Molecular Biology*, 186(1):117–128, Nov. 1985.

[11] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Databases*, Edinburgh, Scotland, 1999.

[12] W. S. Hayes and M. Borodovsky. Deriving ribosomal binding site (RBS) statistical models from unannotated DNA sequences and the use of the RBS model for N-terminal prediction. In *Pacific Symposium on Biocomputing*, pages 279–290, 1998.

[13] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7/8):563–577, July/August 1999.

[14] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 604–613, Dallas, TX, May 1998.

[15] M. Kozak. Comparison of initiation of protein synthesis in procaryotes, eucaryotes, and organelles. *Microbiological Reviews*, 47(1):1–45, 1983.

[16] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 8 October 1993.

[17] C. E. Lawrence and A. A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function, and Genetics*, 7:41–51, 1990.

[18] B. Lewin. *Genes VI*. Oxford University Press, 1997.

[19] M. Linial, N. Linial, N. Tishby, and G. Yona. Global self-organization of all known protein sequences reveals inherent biological signatures. *Journal of Molecular Biology*, 268:539–556, 1997.

[20] C. J. McInerny, J. F. Partridge, G. E. Mikesell, D. P. Creemer, and L. L. Breeden. A novel Mcm1-dependent element in the SWI4, CLN3, CDC6, and CDC47 promoters activates M/G_1-specific transcription. *Genes & Development*, 11:1277–1288, 1997.

[21] A. L. Means and P. G. Farnham. Transcription initiation from the dihydrofolate reductase promoter is positioned by *hip1* binding at the initiation site. *Molecular and Cellular Biology*, 10:653–61, 1990.

[22] S. Natsan and M. Gilman. Yy1 facilitates the association of serum response factor with the c-fos serum response element. *Molecular and Cellular Biology*, 15:5975–82, 1995.

[23] P. Pevzner and S.-H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 269–278, San Diego, CA, Aug. 2000. AAAI Press.

[24] I. Rigoutsos and A. Floratos. Motif discovery without alignment or enumeration. In *RECOMB98: Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 221–227, New York, NY, Mar. 1998.

[25] E. Rocke and M. Tompa. An algorithm for finding novel gapped motifs in DNA sequences. In *RECOMB98: Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 228–233, New York, NY, Mar. 1998.

[26] M.-F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. In C. L. Lucchesi and A. V. Moura, editors, *Latin '98: Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 111–127. Springer, 1998.

[27] S. Sinha and M. Tompa. A statistical method for finding transcription factor binding sites. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 344–354, San Diego, CA, Aug. 2000. AAAI Press.

[28] R. Staden. Methods for discovering novel motifs in nucleic acid sequences. *Computer Applications in the Biosciences*, 5(4):293–298, 1989.

[29] M. Tompa. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 262–271, Heidelberg, Germany, Aug. 1999. AAAI Press.

[30] J. van Helden, B. André, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281(5):827–842, Sept. 4 1998.

[31] E. Wingender, P. Dietze, H. Karas, and R. Knüppel. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Research*, 24(1):238–241, 1996. `http://transfac.gbf-braunschweig.de/TRANSFAC/`.