FINDING WEAK MOTIFS IN DNA SEQUENCES

S.-H. SZE[1], M.S. GELFAND[2], P.A. PEVZNER[1]
[1]*Department of Computer Science and Engineering,*
University of California at San Diego, La Jolla, CA 92093-0114.
[2]*Integrated Genomics — Moscow, P.O. Box 348, 117333, Moscow, Russia.*

Recognition of regulatory sites in unaligned DNA sequences is an old and well-studied problem in computational molecular biology. Recently, large-scale expression studies and comparative genomics brought this problem into a spotlight by generating a large number of samples with unknown regulatory signals. Here we develop algorithms for recognition of signals in corrupted samples (where only a fraction of sequences contain sites) with biased nucleotide composition. We further benchmark these and other algorithms on several bacterial and archaeal sites in a setting specifically designed to imitate the situations arising in comparative genomics studies.

1 Introduction

Large-scale expression analysis and comparative genomics recently generated numerous samples of potentially co-regulated genes whose upstream regions are likely to contain regulatory sites. These samples are often corrupted (with only a fraction of sequences in the sample containing a site) and the corresponding signals may be relatively weak. In difference from previous "gene-by-gene" research efforts, the possibilities of experimental localization of site positions (i.e., via reduction in the length of sequences in the samples by footprinting experiments) in postgenomic era are limited. As a result, computer predictions are often the only realistic way to find regulatory signals in these regions.

The first attempts to find regulatory sites appeared in the early eighties (for reviews, see Gelfand[1], Frech *et al.*[2], or Brazma *et al.*[3]). Current approaches can be roughly subdivided into pattern-driven techniques[4,5,6,7] and profile-based optimization algorithms (greedy search[8], simulated annealing[9], Gibbs sampler[10], and expectation-maximization[11]). Most pattern finding algorithms were developed and tested in a situation when all or most sequences in the analyzed sample contain regulatory sites (mostly single site). This is no longer a valid assumption. Comparative genomics produce samples of genes that are *likely* to be co-regulated, but there is no guarantee that some (or maybe even a majority) of the genes are expressed constitutively or regulated by the same mechanism. Similarly, expression studies often result in the identification of co-expressed genes in response to certain environmental stimuli, but usually do not resolve regulatory cascades and other complex interactions[12].

Several papers reported benchmarking results of signal recognition algorithms. Fickett and Hatzigeorgiou[13] evaluated algorithms for finding eukaryotic promoters. Roulet *et al.*[14] compared predicted affinities of an eukaryotic transcription factor to synthetic oligonucleotides from SELEX data. Frech *et al.*[2] benchmarked programs for finding signals on several prokaryotic and eukaryotic samples. Pevzner and Sze[6] compared several pattern finding algorithms on simulated sequences with implanted signals. Although these results allow one to assess the current state of affairs in controlled situations, they do not provide insight to the behavior of existing programs in real life situations.

Here we are primarily interested in benchmarking with corrupted samples where the signal is present only in a fraction of sequences. This is almost always the case in biological samples. This study models a common situation when it is unclear how to set up the size of the upstream regions and the search parameters (i.e. the length and stringency of the motif). A failure to choose the right parameters may lead to missing the signals. We investigate the effect of using different lengths of upstream sequences and study how the corruption of the sample sequences influences the quality of recognition. We are also interested in how the addition of sequences with a different signal (possibly of different length) to a sample affects recognition. We investigate "how weak" a weak signal should be to become undetectable.

The algorithms WINNOWER and SP-STAR from Pevzner and Sze[6] have been modified to take into account specifics of real biological samples. We compare these programs with a few of the best currently available approaches, including CONSENSUS[15], GibbsDNA[10], and MEME[16]. The choice of the programs is somehow subjective and is limited to those that are the most popular among biologists. We will identify the shortcomings of these approaches and gain insight into what problems future approaches should address.

2 Test Samples

The algorithms were tested on three samples from the *E.coli* genome. Each sample consists of sequences in a $[-1500, 500]$ window with respect to the translational start site annotated with known sites from the Robison *et al.*[17] compilation. The sequences were extracted from GenBank using GenomeExplorer[18]. In our experiments, subsamples within a smaller window of these samples are considered, so that the actual lengths of the sequences used will be smaller.

The first (ARG) sample contains 9 sequences (genes regulated by the arginine repressor ArgR). Each sequence contains a two-part site with the length of each part being 18 nt and separated by 3 nt in all but one sequence where the separation is 2 nt. One of the sequences also has an extra one-part site of length 18 nt. The second (PUR) sample contains 19 sequences with sites of length 16 nt (genes regulated by the purine repressor PurR). Among them, 17 sequences contain 1 site and 2 sequences contain 2 sites. The third CRP sample contains 33 sequences with CRP repressor binding sites of length 22 nt. Among them, 17 sequences contain 1 site, 10 sequences contain 2 sites, 1 sequence contains 3 sites, 1 sequence contains 4 sites, and 4 sequences contain no sites. The sites in the CRP sample are mostly weak. Most sites are found within 200 nt upstream of the translational start site, although a few sites are found up to 400 nt upstream or downstream of the start site.

Define the *majority string* for a collection of strings $\mathcal{W} = \{W_1, \ldots, W_t\}$ as the string W' whose ith letter is the most frequent ith letter in $\mathcal{W}$. We estimate the mutation rate of the signal in each sample by finding the majority string from the set of annotated patterns and computing the average number of substitutions to convert the majority string to each of the annotated patterns. We use the notation of a (l, d)-sample to denote a sample with signals of length l and probability of mutation $p = d/l$ (see Pevzner and Sze[6], the VM mode). The ARG sample contains two-part sites. When only the two-part sites separated by 3 nt are considered with each two-part site treated as one site, the sample corresponds roughly to a (39,11)-sample (i.e., 11 mutations per 39 positions or 28% mismatches on average) and only 8 out of 9 sequences contain a site. When each part is considered a site by itself, the sample is roughly a (18,5.6)-sample (31% mismatches on average) and most sequences contain two sites. The PUR sample corresponds roughly to a (16,3.4)-sample (21% mismatches on average) and most sequences contain one site. The CRP sample corresponds roughly to a (22,9.1)-sample (41% mismatches on average) and most sequences contain one or two sites.

We also study two samples with unknown regulatory sites. The IRON-FACTOR sample contains 12 sequences each of length 250 nt. These sequences are the upstream regions of operons from various gamma-proteobacteria likely to be involved in iron utilization and regulated by homologous repressors other than FUR (E. Panina, personal communication). The PYRO-PURINES sample contains 13 sequences each of length 300 nt (upstream regions of genes involved in the purine metabolism in *Pyrococus horikoshii*). Recently, Gelfand *et al.*[19] made (still unconfirmed) prediction of regulatory sites in this sample.

3 Results

Following Pevzner and Sze[6], we use the performance coefficient $|K \cap P|/|K \cup P|$ to evaluate the performance of signal finding algorithms, where K is the set of known signal positions in a sample and P is the set of predicted positions.

3.1 Samples with a Single Site per Sequence

Most motif finding programs have a performance tradeoff when using a more general model versus a more restricted model. To compare the performance of various approaches fairly, we first assume that the signal length is known as the annotated length and at most one site appears in a sequence. Although the second assumption does not hold for our samples, we assume it here for simplicity and expect that the programs to only be able to pick up the strongest site in a sequence with more than one site. In particular, for the ARG sample, we treat the two-part signal as one signal of length 39 nt and change the annotation to remove the extra one-part site and the exceptional two-part site with separation distance 2 nt. The 3 nt in the separation portion of the two-part signal is considered to be annotated. For the PUR sample, we remove the weaker site from the annotation in each of the two sequences where there are two annotated sites. Since a lot of the sequences in the CRP sample have more than one site, we postpone its test to the later sections when more complicated models are considered. Since there is no convenient way to test GibbsDNA or WINNOWER under the current model (both programs can return more than one site per sequence), we postpone their tests.

We investigate the effect of using different lengths of upstream sequences from 200 nt to 1500 nt. Since all the sites are found upstream of the translational start site, we fix the right end of the sequences to be the position just before the start site and vary the left end. All the programs performed similarly (data not shown). In most cases, the performance was 0.89 on the ARG sample and 0.95 on the PUR sample, independent of the length chosen.

We are interested in how the addition of random sequences to each of the ARG and PUR samples influences the signal recognition. Since most sites can be found within 200 nt upstream of the start site, we fix the upstream sequence length under investigation to be 200 nt. A sample of 666 random fragments of length 200 nt is also given. These sequences contain intergenic regions between convergently transcribed genes which are not expected to contain binding sites for any regulator. An increasing number of these random sequences are added to each sample. Table 1 compares the performance of the various algorithms. We allow each program to return suboptimal solutions in addition to the optimal one and the top-ranked non-overlapping suboptimal solutions are considered. MEME was the best in returning a strong signal as the optimal solution, but sometimes with performance tradeoff since CONSENSUS performed very well in returning an excellent quality result among the top few non-overlapping suboptimal solutions. SP-STAR sometimes performed better than CONSENSUS or MEME but gave inferior results in general.

Table 1: Comparison of the performance of the various algorithms by adding random sequences to the ARG and PUR samples with upstream sequences of length 200 nt under the restricted model where all the programs return at most one site per sequence. Annotations of the samples have been changed to suit the restricted model. For CONSENSUS, the stopping condition is that each sequence has contributed exactly one word to the saved matrices and we consider all the top matrices from each cycle. MEME is run in zoops mode, not allowed to shorten motifs, and is instructed to find three different motifs. SP-STAR is run with local improvements on the top 10% initial signals. The known signal length is 39 nt for the ARG sample and 16 nt for the PUR sample, which is used as an input parameter to all the programs. The top three non-overlapping suboptimal solutions among these results are considered, where each one does not overlap with any of the higher-scored ones, and the one with the highest performance among these non-overlapping solutions is reported along with its suboptimal position in parentheses. Note that sometimes less than three suboptimal solutions are returned from a program.

		number of random sequences added								
sample	program	0	20	40	60	80	100	120	140	160
ARG	CONSENSUS	0.81(1)	1(1)	1(1)	1(1)	1(1)	1(1)	1(2)	1(2)	0.29(3)
	MEME	0.89(1)	0.90(1)	0.90(1)	0.89(1)	0.73(1)	0.34(1)	0.72(2)	0.89(1)	1(1)
	SP-STAR	1(1)	0.81(1)	1(1)	1(1)	0.53(1)	0.53(1)	0.53(2)	0.42(2)	0.53(2)
PUR	CONSENSUS	0.94(1)	1(1)	1(1)	0.94(1)	0.88(1)	0.88(1)	0.88(1)	0.58(1)	0.58(2)
	MEME	0.94(1)	1(1)	0.94(1)	0.94(1)	0.60(1)	0.60(1)	0.60(1)	0.52(1)	0.63(1)
	SP-STAR	0.94(1)	1(1)	1(1)	1(1)	1(1)	1(1)	0.53(1)	0.53(1)	0.53(2)

3.2 Samples with Multiple Sites per Sequence

All the programs in this study can predict multiple sites per sequence. For CONSENSUS, MEME and SP-STAR, the total number of sites in a prediction is restricted to mt, where m is an input parameter to be determined, and t is the number of sequences in a sample. For GibbsDNA, mt is used as the expected number of sites supplied as a parameter to the program. For WINNOWER, all solutions with the total number of sites greater than mt are discarded. Of course, the "mt-restriction" has different implications for different programs, but they represent the closest possible models that these programs offer so that the performances are approximately comparable. We want to set m appropriately so as to obtain the best sensitivity for each program, which means that we have to set m to be as small as possible but should still allow the programs to include most or all sites in a prediction. For the ARG sample, when the signal is considered to be a single (two-part) signal (we do not change the annotation, so there are definitely misses of sites), we can set m to 1. When the signal is considered to be one-part, there are 19 sites in 9 sequences. We can set m to be either 2 or 3. We choose to use $m = 2$ in our experiments since some of the sites will be excluded when smaller window subsamples are considered, which allows a maximum of 18 sites to be predicted with better sensitivity. For the PUR sample, there are 21 sites in 19 sequences, we set m to 1 for similar reasons. For the CRP sample, there are 45 sites in 33 sequences (with two of them overlapping), so we set m to 2. All programs are instructed to return predictions with non-overlapping sites.

The first experiment investigates the effect of the length of upstream sequences. Since almost all the sites are found upstream of the start site, we fix the right end to be the position just before the start site and vary the left end. Table 2 compares the performance of the various algorithms. While CONSENSUS and MEME had good performance in general, GibbsDNA and SP-STAR started to break in some cases when very long upstream sequences are used. WINNOWER only had good performance when short upstream sequences are used (partly due to the fact that we use clique size $k = 2$ instead of $k = 3$ to save computational resources).

The second experiment investigates how the addition of an increasing number of random sequences to the ARG, PUR and CRP samples with upstream sequences of length 200 nt influences the signal recognition. Table 3 compares the performance of the various algorithms, employing the same treatment to allow suboptimal solutions as in Table 1 (excluding GibbsDNA and WINNOWER since the versions we have are not designed for this type of problems). For the ARG samples looking for two-part signals, performance of CONSENSUS and SP-STAR were not bad while MEME returned a good prediction as the top result through a wider range. For the ARG sample (one-part signals) or the CRP sample, SP-STAR had the best performance in returning good solutions among the top results even when a lot of random sequences are added. For the PUR sample, CONSENSUS was the best to return the closest signal as more and more random sequences are added, but it also failed earlier.

In the third experiment we are interested in how the various algorithms perform on samples containing natural but weak sites. We remove sequences successively from the CRP sample (with upstream sequences of length 200 nt) in decreasing order of the strength of the strongest site in a sequence (stronger ones removed first) and investigate when the algorithms break. We compute site strength by the following procedure. Compute the majority string of all the sites and the sum-of-pairs (SP) similarity score of each column of aligned sites. We ignore all positions in the majority string with negative SP column scores and take this string to be the consensus pattern. For the CRP sample, the consensus pattern is found to be `a--tgtga------tcaca-tt`. The site strength is defined to be the similarity score between the site and the consensus pattern computed over retained positions. Table 4 compares the performance of the various algorithms. When not too many CRP sequences are removed, performances of all the programs were comparable except that

Table 2: Comparison of the performance of the various algorithms by using upstream sequences of different lengths from the ARG, PUR and CRP samples allowing multiple sites per sequence. For CONSENSUS, the stopping condition is that the saved matrices contain a maximum of mt words, where m is a parameter and t is the number of sequences, and the first matrix among the list of matrices from each cycle is returned. GibbsDNA is run with the expected number of sites being mt, set to disregard fragmentation and the result with the highest NetMAP score over 100 runs is returned. MEME is run in tcm mode, with the maximum number of sites restricted to mt and not allowed to shorten motifs. WINNOWER is run with clique size $k = 2$ (not tested on the PUR and CRP samples since extensive computation time and resources are needed). SP-STAR is run with local improvements on the top 10% initial signals with the maximum number of sites in a prediction being mt. The known signal length is 39 nt for the ARG sample (2-part) with $m = 1$, 18 nt for the ARG sample (1-part) with $m = 2$, 16 nt for the PUR sample with $m = 1$, and 22 nt for the CRP sample with $m = 2$. All programs return predictions with non-overlapping sites.

		length of upstream sequences													
sample	program	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500
ARG	CONSENSUS	0.81	0.79	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
(2-part)	GibbsDNA	0.81	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
	MEME	0.73	0.79	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.71	0.71	0.71	0.65
	WINNOWER	0.62	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
	SP-STAR	0.81	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.00
ARG	CONSENSUS	0.71	0.62	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.62	0.62	0.62
(1-part)	GibbsDNA	0.69	0.68	0.63	0.63	0.63	0.64	0.64	0.67	0.67	0.67	0.67	0.60	0.54	0.58
	MEME	0.48	0.62	0.62	0.80	0.71	0.76	0.80	0.80	0.62	0.62	0.85	0.56	0.85	0.56
	WINNOWER	0.37	0.40	0.36	0.40	0.31	0.36	0.40	0.40	0.40	0.40	0.40	0.14	0.14	0.14
	SP-STAR	0.54	0.83	0.83	0.83	0.55	0.55	0.55	0.55	0.52	0.48	0.48	0.48	0.48	0.48
PUR	CONSENSUS	0.94	0.95	0.95	0.90	0.81	0.85	0.85	0.85	0.85	0.85	0.85	0.76	0.76	0.76
	GibbsDNA	0.89	0.95	0.95	0.90	0.69	0.69	0.55	0.79	0.86	0.86	0.86	0.82	0.82	0.78
	MEME	0.94	0.95	0.95	0.90	0.81	0.81	0.68	0.81	0.81	0.81	0.90	0.81	0.81	0.85
	SP-STAR	0.94	0.94	0.95	0.89	0.95	0.95	0.85	0.85	0.80	0.80	0.80	0.90	0.80	0.48
CRP	CONSENSUS	0.35	0.38	0.44	0.39	0.38	0.37	0.37	0.37	0.32	0.39	0.32	0.31	0.23	0.23
	GibbsDNA	0.47	0.39	0.35	0.35	0.29	0.26	0.33	0.32	0.34	0.25	0.30	0.26	0.00	0.00
	MEME	0.38	0.36	0.45	0.44	0.44	0.44	0.44	0.43	0.43	0.45	0.43	0.41	0.40	0.38
	SP-STAR	0.43	0.32	0.33	0.38	0.33	0.35	0.37	0.25	0.24	0.24	0.24	0.24	0.32	0.32

WINNOWER gave slightly worse results. Since this sample is a very good representative of samples with weak sites, we further investigate in detail the effect of both varying the length of the upstream sequences and the number of sequences removed. We start from the sample with upstream sequences of length 1500 nt and remove sequences in decreasing order of the strength of the strongest site as before. Samples of shorter lengths are obtained by varying the left end. The consensus pattern computed from this larger sample is `aa-tgtga------tcaca-tt`, slightly different than before. Table 5 compares the performance of CONSENSUS, MEME and SP-STAR. While CONSENSUS and MEME had a better performance when the upstream sequence length is

Table 3: Comparison of the performance of the various algorithms by adding random sequences to the ARG, PUR and CRP samples with upstream sequences of length 200 nt allowing multiple sites per sequence. Settings are the same as in Table 2. The treatment of suboptimal solutions and the notations used are the same as in Table 1. When the top three solutions all have performance less than 0.05, we put 0.00 in the entry to emphasize that the run fails completely.

		number of random sequences added								
sample	program	0	20	40	60	80	100	120	140	160
ARG	CONSENSUS	0.81(1)	0.81(1)	0.81(1)	0.81(2)	0.81(2)	0.81(2)	0.81(2)	0.81(2)	0.28(3)
(2-part)	MEME	0.73(1)	0.73(1)	0.73(1)	0.56(1)	0.73(1)	0.73(1)	0.73(1)	0.81(1)	0.41(2)
	SP-STAR	0.81(1)	0.66(1)	0.81(1)	0.66(1)	0.66(1)	0.46(1)	0.49(2)	0.46(2)	0.46(3)
ARG	CONSENSUS	0.71(1)	0.54(1)	0.64(1)	0.46(1)	0.48(3)	0.45(2)	0.00	0.00	0.00
(1-part)	MEME	0.48(1)	0.46(1)	0.47(1)	0.43(1)	0.37(1)	0.39(2)	0.00	0.00	0.00
	SP-STAR	0.54(1)	0.56(1)	0.56(1)	0.47(1)	0.43(1)	0.42(1)	0.45(3)	0.12(3)	0.44(3)
PUR	CONSENSUS	0.94(1)	0.94(1)	0.89(1)	0.83(1)	0.83(1)	0.83(1)	0.55(2)	0.55(2)	0.00
	MEME	0.94(1)	0.94(1)	0.89(1)	0.89(1)	0.57(1)	0.52(1)	0.50(1)	0.45(1)	0.50(2)
	SP-STAR	0.94(1)	0.94(1)	0.94(1)	0.94(1)	0.94(2)	0.50(2)	0.50(2)	0.50(2)	0.50(2)
CRP	CONSENSUS	0.35(1)	0.37(1)	0.36(1)	0.31(3)	0.00	0.00	0.00	0.00	0.00
	MEME	0.38(1)	0.40(1)	0.35(1)	0.33(2)	0.27(3)	0.28(3)	0.00	0.25(2)	0.00
	SP-STAR	0.43(1)	0.42(1)	0.35(1)	0.41(1)	0.36(2)	0.37(2)	0.36(3)	0.00	0.00

Table 4: Comparison of the performance of the various algorithms by removing sequences from the CRP sample with upstream sequences of length 200 nt in decreasing order of a sequence's strongest site strength. Settings are the same as in Table 2.

CRP	number of CRP sequences removed								
sample	0	3	6	9	12	15	18	21	24
CONSENSUS	0.35	0.39	0.20	0.20	0.10	0.08	0.15	0.09	0.00
GibbsDNA	0.38	0.41	0.36	0.30	0.32	0.16	0.18	0.11	0.00
MEME	0.38	0.42	0.27	0.18	0.12	0.20	0.14	0.15	0.00
WINNOWER	0.19	0.09	0.10	0.13	0.00	0.09	0.00	0.00	0.00
SP-STAR	0.52	0.42	0.34	0.27	0.22	0.23	0.25	0.14	0.00

not too long, SP-STAR was more successful in the difficult cases. The maximum performance achieved was only about 50%, mostly due to the variability of the signal: if we consider the 14 non-degenerate positions in the consensus pattern, about half of the instances are at least 4 mismatches away, which is beyond the limit of the algorithms.

In the fourth experiment we are interested in how the addition of sequences from another sample influences the signal recognition. Similar to before, only upstream sequences of length 200 nt are considered. An increasing number of sequences from the CRP samples sorted in decreasing order of a sequence's strongest site strength are added to each of the ARG and PUR samples (stronger ones added first). Table 6 compares the performance of the

Table 5: Comparison of the performance of the various algorithms by varying both the lengths of the upstream sequences and the number of sequences that are removed from the CRP sample. Settings are the same as in Table 2.

CRP seqs. removed	CRP sample														
		length of upstream sequences													
	program	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500
0	CONSENSUS	0.35	0.38	0.44	0.39	0.38	0.37	0.34	0.37	0.32	0.39	0.32	0.31	0.23	0.23
	MEME	0.38	0.40	0.47	0.44	0.44	0.42	0.43	0.43	0.43	0.45	0.43	0.41	0.40	0.38
	SP-STAR	0.41	0.31	0.35	0.34	0.33	0.37	0.30	0.30	0.21	0.21	0.20	0.22	0.25	0.19
3	CONSENSUS	0.39	0.40	0.45	0.43	0.42	0.40	0.38	0.38	0.32	0.29	0.28	0.30	0.00	0.00
	MEME	0.38	0.44	0.45	0.44	0.43	0.44	0.44	0.45	0.44	0.45	0.43	0.31	0.40	0.37
	SP-STAR	0.33	0.30	0.39	0.35	0.34	0.30	0.30	0.31	0.23	0.27	0.26	0.23	0.20	0.22
6	CONSENSUS	0.40	0.43	0.39	0.40	0.37	0.37	0.36	0.36	0.34	0.34	0.00	0.37	0.25	0.00
	MEME	0.52	0.44	0.46	0.42	0.43	0.47	0.46	0.43	0.39	0.38	0.39	0.39	0.37	0.00
	SP-STAR	0.33	0.34	0.33	0.30	0.25	0.25	0.30	0.29	0.29	0.31	0.29	0.29	0.22	0.29
9	CONSENSUS	0.41	0.44	0.41	0.41	0.39	0.39	0.39	0.39	0.37	0.37	0.37	0.39	0.41	0.27
	MEME	0.44	0.47	0.46	0.41	0.41	0.46	0.46	0.42	0.44	0.43	0.40	0.40	0.36	0.00
	SP-STAR	0.35	0.38	0.33	0.30	0.28	0.35	0.32	0.31	0.31	0.31	0.31	0.31	0.31	0.31
12	CONSENSUS	0.28	0.41	0.38	0.37	0.35	0.35	0.36	0.27	0.21	0.21	0.21	0.22	0.22	0.00
	MEME	0.38	0.43	0.38	0.36	0.38	0.38	0.35	0.37	0.34	0.31	0.26	0.35	0.31	0.00
	SP-STAR	0.33	0.41	0.40	0.32	0.34	0.31	0.40	0.27	0.27	0.27	0.27	0.27	0.27	0.27
15	CONSENSUS	0.24	0.16	0.35	0.34	0.34	0.34	0.31	0.32	0.02	0.02	0.01	0.18	0.00	0.00
	MEME	0.39	0.34	0.43	0.27	0.39	0.36	0.33	0.37	0.29	0.01	0.27	0.26	0.00	0.26
	SP-STAR	0.29	0.40	0.31	0.29	0.29	0.29	0.33	0.33	0.24	0.24	0.24	0.24	0.24	0.24
18	CONSENSUS	0.12	0.08	0.09	0.09	0.09	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
	MEME	0.05	0.27	0.26	0.25	0.14	0.16	0.14	0.03	0.00	0.00	0.00	0.00	0.00	0.00
	SP-STAR	0.29	0.32	0.29	0.28	0.28	0.28	0.28	0.28	0.29	0.22	0.26	0.03	0.00	0.00
21	CONSENSUS	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01
	MEME	0.10	0.00	0.05	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.02	0.00	0.00
	SP-STAR	0.14	0.35	0.23	0.32	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	CONSENSUS	0.00	0.16	0.04	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	MEME	0.11	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SP-STAR	0.11	0.29	0.29	0.29	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

various algorithms. Overall, SP-STAR was least affected by the addition of CRP sequences. GibbsDNA and WINNOWER did not have good performance when a lot of CRP sequences are added. For the ARG sample (one-part signals), CONSENSUS and MEME were not very stable when a moderate amount of CRP sequences are added. In fact, excellent solutions were returned as the second (non-overlapping) suboptimal solution in all these cases.

3.3 Samples with Unknown Signals

Table 7(a) shows the results of running SP-STAR on the IRON-FACTOR sample. The consensus shows that the best signal found is highly palindromic

Table 6: Comparison of the performance of the various algorithms by adding CRP sequences in decreasing order of a sequence's strongest site strength to the ARG and PUR samples with upstream sequences of length 200 nt. Settings are the same as in Table 2.

		number of CRP sequences added								
sample	program	0	3	6	9	12	15	18	21	24
ARG (2-part)	CONSENSUS	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
	GibbsDNA	0.81	0.81	0.81	0.81	0.81	0.81	0.73	0.40	0.38
	MEME	0.73	0.81	0.51	0.81	0.66	0.66	0.73	0.81	0.81
	WINNOWER	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
	SP-STAR	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
ARG (1-part)	CONSENSUS	0.71	0.67	0.61	0.61	0.50	0.46	0.00	0.00	0.55
	GibbsDNA	0.69	0.63	0.61	0.53	0.47	0.28	0.13	0.25	0.09
	MEME	0.48	0.48	0.50	0.51	0.53	0.39	0.00	0.51	0.51
	WINNOWER	0.37	0.37	0.37	0.37	0.37	0.00	0.19	0.19	0.19
	SP-STAR	0.54	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61
PUR	CONSENSUS	0.94	0.94	0.94	0.89	0.89	0.89	0.85	0.85	0.85
	GibbsDNA	0.89	0.89	0.89	0.85	0.85	0.85	0.74	0.68	0.71
	MEME	0.94	0.94	0.89	0.85	0.85	0.81	0.74	0.74	0.74
	WINNOWER	0.94	0.81	0.89	0.84	0.84	0.84	0.71	0.71	0.71
	SP-STAR	0.94	1	1	0.89	0.89	0.89	0.85	0.85	0.85

which reinforces our belief that it is very likely to be a biological signal. Table 7(b) shows the results of running SP-STAR on the PYRO-PURINES sample. Gelfand *et al.*[19] made a prediction on this sample and we found that our results agree very well with their prediction. We have also run CONSENSUS (using signal lengths found in Table 7 as input parameter) and MEME on the two samples and found that these programs give similar results. If the predictions are assumed to be correct, the IRON-FACTOR sample corresponds to a (29,8.4)-sample (29% mismatches on average), while the PYRO-PURINES sample corresponds to a (22,5.6)-sample (25% mismatches on average).

4 Discussion

We have tested and compared the performance of five programs CONSENSUS, GibbsDNA, MEME, WINNOWER and SP-STAR on several biological samples. All programs perform well on non-corrupted samples when all sequences contain relevant binding sites. This condition is very difficult to satisfy in practice. Indeed, many methods used for sample generation, including clustering of genes with similar expression profiles, analysis of reconstructed metabolic maps, and locating orthologous genes from known regulons in a related species, are very likely to create sequences not belonging to the analyzed regulon. Thus, an important part of the analysis presented here is benchmark-

Table 7: Results on the (a) IRON-FACTOR and (b) PYRO-PURINES samples. Shown is the best solution given by SP-STAR while looking for signals up to 40 nt in length with the maximum total number of sites in a prediction restricted to $2t$, where t is the number of sequences in the sample. The last string shown is the majority string, showing only the positions with positive SP column score.

(a)			(b)		
name	pos	pattern	name	pos	pattern
b_alcA	38	gagaatagaagtcataattattctcattaa	*PH0239*	189	cttttgccagatatatgtctaaaaaa
b_alcR	166	ataaaagcgaatgaattgcattatcattaa	*PH0239*	231	atttttacataaacatggtgaaatta
s_foxA	189	ctaaagggtaataattcttatttacaataa	*PH0240*	190	atttcaccatgtttatgtaaaaatca
v_OM	159	atatatgcgaatcgttatcatttgtatttt	*PH0240*	232	ttttagacatatatctggcaaaagat
v_reg	189	aaaaatacaaatgataacgattcgcatata	*PH0318*	187	atttaaacatatttatgttaaaaagg
y_ybtA	103	attaatgtgaataataaccattatcaataa	*PH0318*	229	attttaacatttatacgtcaattagg
y_ybtP	150	gttattgataatggttattattcacattaa	*PH0320*	150	cgattagcacatatatgtagaaatat
		ataaatg--aat-atta--att--cattaa	*PH0323*	186	ttgttaacacgtttatgtaaacaaaa
			PH0323	229	attttgacttaaatatggtgatataa
			PH0438	186	ctattaacatagccctgtcaaaaggg
			PH0852	177	agatttctacaaatatgtcaaaaaca
			PH0852	220	attttaccgtgaaaatggtgatataa
			PH1955	166	tgattgacatttctttgtcaaaataa
			PH1955	208	atttttacatttttctggcaaataag
					atttt-acatatatatgtcaaaa--a

ing of the programs on corrupted samples. This was modeled in three ways: adding sequences with no sites, removing the strongest sites from a sample, and adding sequences with sites of a different origin to a sample.

In the experiment on addition of random sequences with no sites, MEME outperformed CONSENSUS and SP-STAR on both analyzed samples when at most one site are allowed per sequence. When multiple sites are allowed, SP-STAR performed slightly better than the other programs in the most difficult cases. In the experiment on removal of strong sites, the leaders were MEME and SP-STAR, with GibbsDNA demonstrating comparable or even slightly superior results when not too many sites are removed. In the experiment on addition of sequences from a different sample, GibbsDNA and WINNOWER clearly trailed, with CONSENSUS and SP-STAR being the leaders.

It does not seem possible to recommend a single program for use in all situations. However, this study allows us to make a few practical suggestions. The first one is simple: use all available programs. It seems that the programs are not affected much by varying fragment lengths. As the sites may occur at varying distances from the start site, it is safer to err to the side of using longer fragments. Also, it looks like that asking for at most one site per sequence improves the performance. In this case, additional sites can be found by standard search methods using consensus or positional weight matrix representation.

Acknowledgments

We are grateful to A. Mironov for many helpful discussions and to E. Panina who provided the IRON-FACTOR sample. This work was partially supported by the Russian Fund of Basic Research (99-04-48247 and 00-15-99362), INTAS (99-1476) and the Howard Hughes Medical Institute (55000309).

References

1. M.S. Gelfand. *J. Comp. Biol.*, 2:87–115, 1995.
2. K. Frech, K. Quandt, T. Werner. *Comp. Appl. Biosci.*, 13:89–97, 1997.
3. A. Brazma, I. Jonassen, I. Eidhammer, D. Gilbert. *J. Comp. Biol.*, 5:279–305, 1998.
4. I. Rigoutsos, A. Floratos. *Bioinformatics*, 14:55–67, 1998.
5. J. van Helden, B. Andre, J. Collado-Vides. *J. Mol. Biol.*, 281:827–42, 1998.
6. P.A. Pevzner, S.-H. Sze. *Proc. of the 8th Int. Conf. on Intelligent Systems for Mol. Biol. (ISMB'2000)*, 269–78, 2000.
7. J. Buhler, M. Tompa. *Proc. of the 5th Annual Int. Conf. on Comp. Mol. Biol. (RECOMB'2001)*, 69–76, 2001.
8. G.D. Stormo, G.W. Hartzell. *Proc. Natl. Acad. Sci.*, 86:1183–7, 1989.
9. A.V. Lukashin, J. Engelbrecht, S. Brunak. *Nucleic Acids Res.*, 20:2511–6, 1992.
10. C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, J.C. Wootton. *Science*, 262:208–14, 1993.
11. T.L. Bailey, C.P. Elkan. *Proc. of the 2nd Int. Conf. on Intelligent Systems for Mol. Biol. (ISMB'1994)*, 28–36, 1994.
12. A.B. Khodursky, B.J. Peter, N.R. Cozzarelli, D. Botstein, P.O. Brown, C. Yanofsky. *Proc. Natl. Acad. Sci.*, 97:12170–5, 2000.
13. J.W. Fickett, A.G. Hatzigeorgiou. *Genome Res.*, 7:861–78, 1997.
14. E. Roulet, I. Fisch, T. Junier, P. Bucher, N. Mermod. *In Silico Biol.*, 1:21–8, 1998.
15. G.Z. Hertz, G.D. Stormo. *Bioinformatics*, 15:563–77, 1999.
16. T.L. Bailey, C.P. Elkan. *Machine Learning*, 21:51–80, 1995.
17. K. Robison, A.M. McGuire, G.M. Church. *J. Mol. Biol.*, 284:241–54, 1998.
18. A.A. Mironov, N.P. Vinokurova, M.S. Gelfand. *Mol. Biol.*, 34:253–62, 2000.
19. M.S. Gelfand, E.V. Koonin, A.A. Mironov. *Nucleic Acids Res.*, 28:695–705, 2000.