Vol. 19 Suppl. 2 2003, pages ii16–ii25
DOI: 10.1093/bioinformatics/btg1054

Searching for statistically significant regulatory modules

Timothy L. Bailey[1,*] and William Stafford Noble[2,†]

[1]Institute for Molecular Bioscience, University of Queensland, Brisbane, Australia and [2]Department of Genome Sciences, University of Washington, 1705 NE Pacific Street, Seattle, WA, 98195, USA

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Motivation: The regulatory machinery controlling gene expression is complex, frequently requiring multiple, simultaneous DNA-protein interactions. The rate at which a gene is transcribed may depend upon the presence or absence of a collection of transcription factors bound to the DNA near the gene. Locating transcription factor binding sites in genomic DNA is difficult because the individual sites are small and tend to occur frequently by chance. True binding sites may be identified by their tendency to occur in clusters, sometimes known as regulatory modules.

Results: We describe an algorithm for detecting occurrences of regulatory modules in genomic DNA. The algorithm, called MCAST, takes as input a DNA database and a collection of binding site motifs that are known to operate in concert. MCAST uses a motif-based hidden Markov model with several novel features. The model incorporates motif-specific p-values, thereby allowing scores from motifs of different widths and specificities to be compared directly. The p-value scoring also allows MCAST to only accept motif occurrences with significance below a user-specified threshold, while still assigning better scores to motif occurrences with lower p-values. MCAST can search long DNA sequences, modeling length distributions between motifs within a regulatory module, but ignoring length distributions between modules. The algorithm produces a list of predicted regulatory modules, ranked by E-value. We validate the algorithm using simulated data as well as real data sets from fruitfly and human.

Availability: http://meme.sdsc.edu/MCAST/paper

Contact: tlb@maths.uq.edu.au

INTRODUCTION

One surprise from the recent analysis of the mouse and human genomes is the relatively large portion of the mouse genome that is evolutionarily conserved but does not code for proteins (Mouse Genome Sequencing Consortium, 2002). Perhaps the most important function of this non-coding DNA is to regulate the rate at which individual genes are transcribed. We refer to the sequence elements that modulate transcription as regulatory modules (Krivan and Wasserman, 2001; Wasserman and Fickett, 1998), though they have alternatively been referred to as promoter modules (Klingenhoff *et al.*, 1999), *cis*-element clusters (Frith *et al.*, 2001), and *cis*-regulatory modules (Berman *et al.*, 2002). A regulatory module typically lies upstream of the gene that it regulates, though examples of downstream, intronic and distant regulators do occur. Each module contains a dispersed collection of short sequences (approximately 6–20 bases), each of which specifically binds to a particular transcription factor protein. The complex interaction among the genomic DNA, the various transcription factors, and the RNA polymerase yields an overall rate of transcription for that particular gene (Ptashne and Gann, 2002).

Two primary computational challenges are associated with regulatory modules. The first involves identifying transcription factor binding site motifs. A given transcription factor can typically bind to a variety of similar, short sites. Hence, given a collection of regulatory modules, it is non-trivial to locate within them the binding sites, even if the given sequences are known *a priori* to interact with a particular transcription factor. The problem becomes harder when the locations of the regulatory modules are uncertain (so that the input sequences must be quite long), or when multiple transcription factors operate on the same set of sequences.

In this work, we address a second, related task (see Fig. 1). We assume that the first problem is solved—that the binding sites have been located within a collection of regulatory modules. We are provided as input a handful of binding site models (motifs) of transcription factors that are believed to operate together. We are also given a large sequence database. The task is to locate in that database occurrences of regulatory modules containing the given binding sites. Hence, the output is a list of predicted site-

*To whom correspondence should be addressed.
†Formerly William Noble Grundy, see www.gs.washington.edu/noble/name-change.html

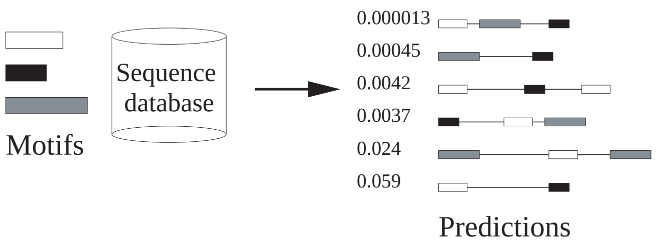

Fig. 1. The regulatory module search problem. The input consists of a collection of motif models and a sequence database. The output is a list of predicted regulatory modules, with annotated binding sites and an associated confidence value for each module.

annotated regulatory modules, along with a measure of confidence in each prediction. We will refer to this as the 'database search task'. This computational problem is a simplified version of the problem solved in the cell by the transcription machinery. In the cell, the sequence database consists of the entire genome, and the binding site models consist of the entire complement of transcription factor binding sites.

Our approach to the database search task is based on a novel scoring function for the alignment of the motif models in the query with a sequence, and an algorithm for finding the alignment that optimizes the scoring function. The scoring function is an extension of the MAST (Bailey and Gribskov, 1998) scoring function that combines the scores of multiple, non-overlapping matches to the motifs in the query. The alignment algorithm is similar to the repeated match algorithm for aligning two sequences (Durbin *et al.*, 1998, p. 24–25). We implemented the algorithm by introducing a new sequence model and search algorithm into Meta-MEME(Grundy *et al.*, 1997), which is part of the MEME family of tools (Bailey and Elkan, 1994). The search algorithm includes a new module for estimating the E-values of match scores. This search algorithm, called MCAST (Motif Cluster Alignment Search Tool), can scan extremely long sequences in limited memory. As a result of several optimizations, MCAST is efficient and can search a ten megabase sequence database with a typical five-motif query in less than a minute on a 800MHz PC, scaling linearly to larger databases or larger queries.

We describe the scoring function, sequence model, search algorithm and E-value computation below. We then describe our methods for testing the algorithm on real and synthetic data, and describe two sets of experimental results: simulations demonstrating the accuracy of our E-value calculation, and a comparison of the performance of MCAST and a previously published method on four real data sets. We end with a discussion of our results and related methods.

ALGORITHM

The scoring function

Conceptually, we think of the alignment between the query motifs and a target DNA sequence as consisting of a set of 'matches', where each match corresponds to a (putative) regulatory module in the sequence. The positions in the sequence that are aligned with a motif model are putative transcription factor binding sites, and we refer to them as 'hits'. We refer to the positions within the matches that are not aligned with motifs as 'gaps'. We call the positions between matches 'inter-cluster regions'. (Note that a match contains one or more hits, and need not contain hits to all of the motifs in the query.)

The goal of the alignment scoring function is to distinguish true regulatory modules from randomly occurring matches. In light of what is known about regulatory modules, any such function must take into account the closeness of the agreement between the motif models and the corresponding hits, the number of hits and the proximity of the hits to each other. Thus, the design of the function must address:

- how to score the hits,
- how to score (penalize) the gaps,
- how to combine the scores of the hits and gaps within a match, and
- how to combine the scores of matches and inter-cluster regions.

Below, we describe how our scoring function addresses each of the above issues, and compare it to the scoring functions of existing methods.

Our scoring function is the sum of scores for the hits, gaps and inter-cluster regions of an alignment. We define A, the overall alignment score between a query (a set of motifs) and a sequence to be

$$A = \sum_{i=1}^{c} M_i + cR,$$

where c is the number of matches (clusters) in the alignment, M_i is the match score of the ith cluster, and R is a 'match penalty'. The match score of the ith cluster we define to be

$$M_i = \sum_{j=1}^{n} (h_j + d_j g),$$

where n is the number of hits in the match, h_j is the score of the jth hit in the match, d_j is the length of the gap between hit $j-1$ and hit j ($d_1 = 0$), and g is a 'gap penalty'. To complete the description of the scoring function, we describe below how we calculate hit scores and match and gap penalties.

We model a transcription factor binding site motif using a position-specific probability matrix (PSPM), in common with two recent approaches to the database search problem CIS-ANALYST (Berman *et al.*, 2002), and COMET (Frith *et al.*, 2002). A PSPM defines a hidden Markov model of motif sites with emission probabilities given by the entries in the matrix, and transition probabilities of 1 between adjacent columns from left to right. The probability of a length-w sequence given such a width-w motif model is the product of the probabilities of the letters, which are given by the matrix. Like the other two search methods, our score for a hit is based on the log-odds score of the subsequence in the target sequence, and the motif model. For subsequence x, the log-odds score, s, is defined as

$$s = \log_2 \frac{Pr(x|\text{motif model})}{Pr(x|\text{background model})}.$$

We use a fixed, user-specified, zero-order Markov background sequence model, as does CIS-ANALYST, and unlike COMET, which uses a background model that varies with the local sequence composition. (For a more complete discussion of the differences between MCAST and COMET, see the Discussion section.)

We define the hit score for the match between a motif model and subsequence x to be

$$P_p(s) = -\log_2 \left(\frac{P(s)}{p} \right),$$

where s is the log-odds score (defined above), p is a user-specified 'p-value threshold', and $P(s)$ is the p-value of log-odds score s—the probability that a randomly chosen sequence position would have log-odds score greater than or equal to s. (In what follows, the variable p always refers to the user-specified p-value threshold.) $P(s)$ is estimated based on the motif model and the background sequence model in the same way as for the MAST algorithm (Bailey and Gribskov, 1998) and further described in Bailey (2003). Clearly, the score $P_p(\cdot)$, which we refer to as a 'p-score', increases monotonically in s, so large values of the p-score correspond to large log-odds scores. However, taking the p-value of s has the effect of down-weighting the log-odds scores of wider motifs because the log-odds score corresponding to a given p-value increases with the width of the motif. Finally, dividing the p-value of s by threshold p means that only scores that are more significant than p will have positive p-scores. The contribution of hit scores to the match score is proportional to the product of the hit p-values. This has been shown to be a highly effective method of combining evidence from multiple motif matches to sequences (Bailey and Gribskov, 1998). Our algorithm only considers positions with positive p-scores as potential hits, so the MCAST scoring method identifies essentially the same potential hits as CIS-ANALYST. In contrast, potential hits considered by COMET depend on the local composition surrounding the hit (via the changing background model) and on a user-specified mean-gap-length parameter, a.

In our scoring function, the total score for a gap is a linear function of its length. Unlike CIS-ANALYST, which ignores gaps up to a user-specified length, gaps are penalized by our scoring function. Our linear gap cost is similar to the affine gap cost imposed by a Viterbi alignment to a hidden Markov model, but avoids the implicit assumption that gaps within matches are best modeled by a geometric length distribution.

The gap penalty, g, is calculated so that, with a random target sequence, random hits and their associated gaps will have scores that essentially cancel one another. That is, g is set so that the expected score of a random hit is approximately equal to (minus) the expected total score of a random gap. It can be shown (Bailey, 2003) that the expected value of the p-score of a random subsequence scored with a single motif is

$$\mu \approx 1/\log 2.$$

The expected score of a gap is g times the expected length of a random gap, D_g. For simplicity, we set

$$g = \frac{-1}{D_0 \log 2},$$

assuming that gaps have zero cost, so the resulting value of g will tend to over-penalize gaps slightly. To estimate D_g, let m be twice the number of motifs in the query. (As we shall see, below, the model will contain $2m$ motifs since we introduce a 'reverse complement' motif for each motif in the query to handle motif occurrences on the reverse complement DNA strand.) When g is zero, D_g can be shown to be approximately

$$D_0 \approx \sum_{i=0}^{L} i p_d(i),$$

where $p_d(i)$ is the probability of a length-i random gap,

$$p_d(i) = \frac{\hat{p}(1-\hat{p})^i}{1-(1-p)^{mL}},$$

and $\hat{p}$ is the probability of a random hit,

$$\hat{p} = 1 - (1-p)^m.$$

The above derivations assume that the motifs are independent of each other and their reverse complements. When this is not true (for example, with motifs that are perfect

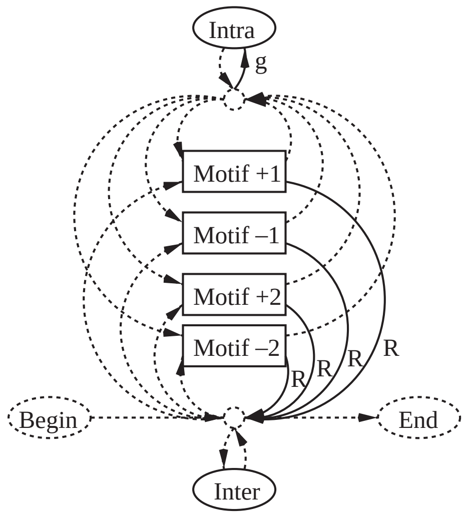

Fig. 2. The hidden Markov model. Dotted lines correspond to zero-cost transitions, and dotted circles are non-emitting states. The values g and R are transition *costs*. Motif states emit fixed-length strings, with costs computed using p-scores (see text for details). The '$+$' and '$-$' versions of each motif correspond to forward-strand and reverse-strand occurrences. The inter-module intra-module states emit single characters with no associated cost.

palindromes), the value of g estimated above will be larger than it should be. Experiments indicate that small changes in g have little effect on the accuracy of the algorithm (data not shown.)[†]

The final component of our alignment scoring function is the penalty for inter-cluster regions (the spaces between putative regulatory modules). We allow the user to specify a maximum allowed gap width, L. We then set the match penalty R to be Lg. Note that R is a *penalty*, and, like g, is negative. The inter-cluster penalty is *not* length-dependent, thereby preventing the optimal alignment from containing gaps longer than L. As a side-effect, the optimal alignment will also contain no matches with score less than R.

The model

The Meta-MEME model for regulatory modules in DNA sequences is shown in Figure 2. MCAST automatically creates this model from the motifs in the query plus the reverse complement of each DNA motif. In the model, motif states emit multiple characters at once, and are therefore similar to a series of single-character states connected by transitions with 1.0 probabilities. The score generated when a motif emits a subsequence is the p-score of the subsequence, as described above. Note that, although we speak of the model 'generating' sequences, the model is not a traditional HMM, because we allow the transition costs to be arbitrary, rather than constraining them to be the logarithms of transition probabilities.

The other states in the model are either non-emitting or free-emitting. The non-emitting states are simply a topological shorthand. We could specify an equivalent but more complex model that did not contain any such states. The inter-module and intra-module states do emit letters (one at a time) but do not accrue any score for doing so. This arrangement is equivalent to setting the background emission probability equal to the foreground probability in these states. We thereby assume that the score of a match is independent of the sequence composition of the non-motif regions in and around the match.

The linear gap cost of intra-match gaps is controlled by the single parameter g. Entering the state marked 'Intra' costs g, so the cost of an intra-match gap of length d is dg. In contrast, the inter-module state has a match cost R associated with entering it, but no cost associated with extending it. Thus this state is like the so-called 'free-insertion modules' used in profile HMMs (Krogh *et al.*, 1994). The match cost must be paid after generating each regulatory module, thereby guaranteeing that a match will occur only if it scores better than R.

The gap penalty g and the match penalty R together determine the maximum length of a gap L between adjacent motifs in a regulatory module. The relationship is simple: $L = R/g$. Using this definition, any gap longer than L will receive a score less than $-R$. The sequence could therefore be generated with less cost by entering the inter-module state. Thus, any candidate match containing a gap longer than L (or more generally, any subsequence scoring less than $-R$) will be split in two.

The MCAST algorithm

Having described the Meta-MEME model, the MCAST algorithm is simple to describe: MCAST is simply the Viterbi algorithm (Viterbi, 1967) applied to this slightly non-conventional HMM, with the added constraint of forbidding transitions into a motif with a negative p-score. The algorithm builds a trellis, in which each row corresponds to a state in the model, and each column corresponds to a position in the sequence. The transitions between adjacent columns in the trellis are determined by the topology of the network. The algorithm finds a path through the model that scores highest with respect to the given DNA sequence. The algorithm is provably optimal, maximizing the scoring function described above. The Viterbi algorithm runs in time $O(\Xi n)$, where Ξ is the number of transitions in the model and n is the length of the DNA sequence. In our model, the number of

[†] For queries with highly dependent motifs, MCAST provides an additional parameter, α, that can be used to make the value of g larger. Details are given in Bailey (2003).

transitions is proportional to the number of motifs, which we assume is a small constant. Hence, the algorithm runs in linear time in the length of the DNA sequence.

For some readers, it may be helpful to note that performing the Viterbi algorithm with this model is equivalent to performing a variant of the repeated match algorithm (Durbin *et al.*, 1998, p. 24–25) with a simpler model. In the repeated match algorithm, the repeat threshold R is specified external to the model. Hence, the free-insertion 'inter' state is removed and some additional bookkeeping is stored in the row corresponding to the 'begin' state. In this formulation, the model has a simple star topology, with the 'intra' state at the center of the star.

One drawback to the Viterbi algorithm is that its memory requirement is $O(mn)$, where m is the number of states in the model. For a multi-motif model and a multi-megabase sequence, this memory requirement quickly becomes unwieldy. Memory-efficient variants of the Viterbi algorithm are available, but they trade memory for speed. Our implementation performs the Viterbi algorithm in large, overlapping sliding windows. The window width and overlap sizes are set larger than the maximum length of a biologically plausible match.

Calculating E-values

We calculate the E-values of match scores by assuming that random match scores follow an exponential distribution. We show empirically that this is true, below. It has been reported that match scores using log-odds hit-scores and affine gap costs also follow an exponential distribution (Frith *et al.*, 2002).

MCAST estimates the parameters of the score distribution empirically from the actual scores generated in a database search. To separate the random scores from true matches we use expectation maximization (EM) (Dempster *et al.*, 1977). We assume a Gaussian distribution for the true match scores, and EM simultaneously estimates the parameters of both components of the mixture of scores (exponential random scores and Gaussian true match scores). The assumption of a Gaussian distribution for true match scores is made for convenience; any unimodal distribution would probably work as well.

We assume that random match scores are exponentially distributed with cumulative density function

$$F(x) = 1 - \exp\left(\frac{R - x}{\mu}\right).$$

The p-value of match score x is therefore

$$P(x) = 1 - F(x) = \exp\left(\frac{R - x}{\mu}\right). \qquad (1)$$

We convert p-values to E-values (the expected number of random match scores greater than x) by multiplying by the (estimated) number of random matches found in the search.

METHODS

We test MCAST to evaluate the accuracy of its estimated E-values and its predictive accuracy. In this section we describe the data we use (motifs and sequence datasets), and our measurement techniques. In all searches, we use as the MCAST background model the base frequencies in the DNA database being searched.

Simulated sequence databases and queries

For some tests of the accuracy of MCAST E-values, we create synthetic sequences containing simulated occurrences of regulatory modules. These are generated using a Meta-MEME model like the one in Figure 2. The model is constructed from five human regulatory motifs, selected at random from TRANSFAC version 6.0 (Wingender *et al.*, 2000). The resulting model (minus the 'Inter' state) is used in a generative fashion to produce motif occurrences interspersed with spacer regions. From a large pool of such sequences, a collection of 10 are selected that contain between 9 and 20 motifs in the first 1000 base pairs (bp). This choice of number and spacing of motifs is based on known CRMs in *Drosophila*. These randomly generated, 1000 bp sequences are then padded on either end with 4500 bp of zero-order random background sequence. The entire process, including motif selection, is repeated 100 times, yielding 100 distinct sets of 10 positive examples. Each of these positive data sets is embedded in a database of 2000 negative examples. Like the positive sequences, the negative sequences are 10000 bp long. The negative sequences are generated from the same zero-order model used to generate the non-motif portions of the positive sequences. For the tests of E-value accuracy, the queries consist of subsets of the same TRANSFAC motifs as used to generate the sequences. For each of the 100 sets of synthetic sequences, four queries containing one, two, four or all five motifs are used.

Real sequence databases and queries

Four sets of co-acting regulatory motifs were collected from previous studies. The first set contains five *Drosophila* motifs (Bcd, Cad, Hb, Kr, Kni) that control expression of the *even-skipped* gene's second stripe. For testing using these motifs, we employ a negative database of 2039 randomly selected intergenic regions from *Drosophila* with average length 1981 bp. The positive database contains nineteen *Drosophila* cis-regulatory modules (CRMs) with average length 1283 bp. These CRMs are taken from nine *Drosophila* genes known to be required for embryonic development and to be regulated by at least one of the transcription factors Bcd, Cad, Hb, Kr and Kni. For further testing of the accuracy of estimated E-values, we use a much larger database (92Mb) containing all intergenic regions from *Drosophila*. These sequence and motif databases were all taken from Berman *et al.* (2002).

The remaining three motif sets are from human. The first set (LSF) contains four motifs associated with LSF-regulated promoters: LSF, Sp1, Ets and the TATA box. These were assembled from Frith *et al.* (2001) (LSF), TRANSFAC accession no. M00196 (Sp1) (Wingender *et al.*, 2000), from Frith *et al.* (2002) (Ets), and from Bucher (1990) (TATA). The second set (muscle) contains five motifs (Mef-2, Myf, SRF, Tef and Sp1) that occur in 27 experimentally supported muscle-specific regulatory regions. The third set of motifs (muscle') is a variant of the second, in which the motif matrices are derived solely from non-muscle-specific genes. This latter set avoids some of the circularity involved in defining matrices based upon motif occurrences that appear in the test set. The muscle and muscle' motif sets were assembled by Wasserman and Fickett (1998). The positive LSF dataset contains nine human CRMs assembled by Frith *et al.* (2002). The positive muscle and muscle' datasets contain the 27 human CRMs assembled by Wasserman and Fickett (1998). For the three human data sets (LSF, muscle and muscle'), the negative dataset is a collection of 2005 randomly-chosen, 2000-bp human genomic sequences, as selected by Frith *et al.* (2002).

All of the real datasets were processed to mask tandem repeats using Tandem Repeats Finder (Benson, 1999). Low complexity regions were then masked with DUST (R. Tatusov and D. Lipman, manuscript in preparation). The parameters used with Tandem Repeats Finder are: (2 7 7 80 10 50 500 -m). With DUST we use the default parameters.

Evaluating the accuracy of E-values.

Because E-values are just p-values multiplied by the number of random matches found in the search, E-value accuracy and p-value accuracy are equivalent. For simplicity, we consider the accuracy of MCAST p-values.

No matter what the form of the match score distribution, the expected p-value of the ith largest random score in a search yielding n matches is

$$p_r(i) = \frac{i+1}{n+1}.$$

We will refer to $p_r(i)$ as the 'rank p-value' of the (ith largest) score.

The accuracy of p-values estimated by MCAST can be evaluated by comparison with the corresponding rank p-values. For a quantitative evaluation, we use the 'p-value slope error' (PSE) (Bailey and Gribskov, 2002) metric. PSE measures how closely the estimated p-values match the corresponding rank p-values. The value of PSE is one minus the slope of the regression curve to the points $\{< P(x_i), p_r(i) >\}$. PSE has the value zero if the estimated p-values are perfectly accurate. For qualitatively verifying the accuracy of estimated p-values, we plot the rank p-value versus match score for all the random matches found in a search, along with a plot of $P(x)$ (Equation (1)). The accuracy of the estimated p-values can be judged by seeing how well the match score points agree with the $P(x)$ curve.

Measuring predictive accuracy

The output of the MCAST algorithm is a ranked list of predicted matches. To evaluate the performance of the algorithm, this ranked list is compared to the set of true matches, and each predicted match is labeled as a 'positive' or 'negative'. Starting from the top of the ranked list, we label as 'positive' each predicted match that overlaps a true match. Whenever a positive is labeled, any lower-ranked predicted matches that overlap the same true match are eliminated from the list. Predicted matches that do not overlap any true match are labeled 'negative'.

Given the ranked list of positive and negative predicted matches, we measure the search algorithm's performance using a variant of the receiver operating characteristic (ROC) score (Gribskov and Robinson, 1996; Schaffer *et al.*, 2001). The ROC curve plots the true positive rate as a function of false positive rate, for varying classification thresholds in the ranked list of predicted matches. The ROC_{50} score is the area under this curve, up to the fiftieth false positive match, normalized so that the score falls between 0 and 1. An ROC_{50} of 1 corresponds to perfect performance (all positives ranked above all negatives). On this task, random performance corresponds to an ROC_{50} close to 0.

For consistency with Frith *et al.* (2002), we also measure the average number of (kilo) base-pairs per false positive at a threshold that yields approximately 60% sensitivity (FP_{60}). However, we believe that ROC_{50} is a more revealing measure of predictive accuracy than FP_{60} when comparing search methods, because ROC_{50} takes into account the complete ordering, rather than focusing on a single point along the ROC curve.

RESULTS

In this section we show that the E-values for match scores computed by MCAST are accurate and show how to use them to select the best settings of the search parameters p and g. We then show the predictive accuracy possible when MCAST is used to search for cis-regulatory modules in *Drosophila* and human sequences, and compare with results using an earlier method. Finally, we examine

Table 1. Accuracy of estimated E-values: real motifs, synthetic sequences

query			*p*		
size	0.00001	0.00005	0.0001	0.0005	0.001
1	–	0.100	0.066	0.022	0.021
2	–	0.057	0.039	0.011	0.008
4	0.068	0.034	0.026	0.006	0.005
5	0.067	0.030	0.015	0.005	0.003

The average PSE (see Methods) for searches of synthetic sequences using various numbers of real human transcription factor binding site motifs in the query is shown. Each value of PSE is the mean of 100 independent experiments. Results are shown for distinct values of p. The parameter g is 200 in all searches. A dash ('–') indicates that the minimum number of matches for E-value estimation (200) are not found in any experiment.

the speed of the MCAST algorithm for searching large genomic databases.

Accuracy of E-values

We test the accuracy of the E-value reported by MCAST in two ways. First, we search synthetic sequences (some of which contain synthetic cis-regulatory modules) with sets of real transcription factor motifs. Second, we search real sequences with shuffled versions of motifs.

In experiments using actual TRANSFAC motifs and synthetic sequences containing occurrences of those motifs, MCAST accurately estimates the E-values of random matches. (See Methods section: Simulated sequence databases and queries.) The average PSE under these conditions is shown in Table 1. Similar results are obtained with different settings of g between 35 and 1000 (data not shown). By way of comparison, PSE for Smith-Waterman alignment scores of protein sequences using the best empirical estimation methods is around 0.03 (Bailey and Gribskov, 2002). These experiments show that both the form of the random match score distribution and our EM algorithm to estimate the distribution from the empirical scores work well when the random matches come from zero-order Markov sequence.

To test the accuracy of the E-values estimated by MCAST when searching real DNA, we shuffle the motifs in each of the four real motif sets (*Drosophila*, LSF, muscle, muscle'). Shuffling is accomplished by randomly permuting the entries in each column (motif position) in the motif. We use the shuffled *Drosophila* motifs to search the large (92Mb) *Drosophila* sequence database. We search our human genomic database using shuffled versions of each of the other three motif sets. Because the motifs are shuffled, we expect all match scores to be essentially random. Typical results are shown in Figure 3. The accuracy of MCAST p-values depends somewhat on p, the p-value threshold. For settings of p below 0.001, the rank p-values (y values of points) are virtually

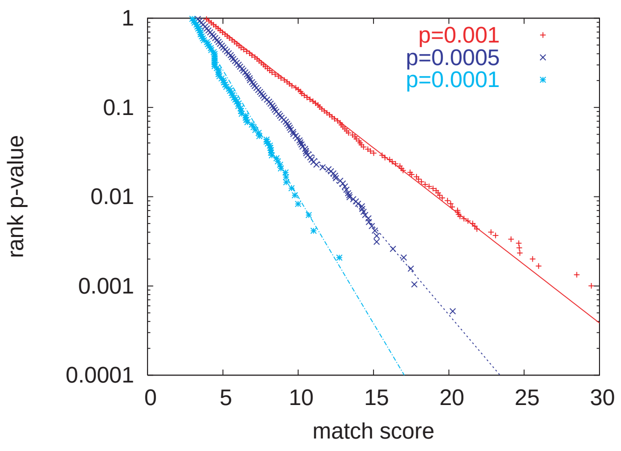

Fig. 3. E-value accuracy: real sequences, shuffled motifs. The plot shows three MCAST searches using the four shuffled LSF motifs and the human intergenic database. The MCAST search parameters are $L = 200$ and the indicated values of p. Each set of points corresponds to one MCAST search. Each curve is the exponential score distribution estimated by MCAST (using EM) for that search.

identical to the estimated p-values (the y values of the estimated exponential distribution curves). The p-values continue to be accurate even for settings of p as low as 0.00005 (data not shown for clarity). Some high-scoring outliers are seen for larger settings of p, so the accuracy of MCAST E-values may suffer with settings of $p \geq 0.001$ in some searches.

Predictive accuracy

We measure the predictive accuracy of MCAST using the *Drosophila* motif set from Berman *et al.* (2002) and the three human motif sets from Frith *et al.* (2002), as described in the Methods section. For comparison, we also measure the accuracy of COMET on the same data. For each search method, we try several parameter settings and report the optimum value of each accuracy measure. This optimization is done separately for each method and each accuracy measure. Five distinct settings of p and ten settings of L are used with MCAST (fifty parameter combinations). Nine distinct settings of expected gap length, a, and six window sizes, w, are used with COMET (fifty-four parameter combinations). All other parameters are left at their defaults.[‡] The best FP$_{60}$ is not always obtained at the same parameter settings as the best ROC$_{50}$.

[‡] Values of MCAST and COMET parameters used in experiments:

$$\text{MCAST}\begin{cases} p \in \{0.00001, 0.00005, 0.0001, 0.0005, 0.001\} \text{ and} \\ L \in \{25, 35, 50, 100, 150, 200, 300, 400, 500, 700\}. \end{cases}$$
$$\text{COMET}\begin{cases} w \in \{75, 150, 300, 600, 1200, 2400\} \text{ and} \\ a \in \{5, 10, 15, 25, 35, 50, 100, 125, 150\}. \end{cases}$$

Table 2. Predictive accuracy for two search methods

motif set	FP$_{60}$		ROC$_{50}$	
	MCAST	COMET	MCAST	COMET
Drosophila	>**4041**	1010	**0.68**	0.61
LSF	**167**	85	**0.44**	0.35
muscle	30	**69**	0.38	**0.46**
muscle’	**14**	6	0.16	**0.25**

Performance is given in terms of FP$_{60}$: number of thousands of base-pairs per false positive at a threshold yielding approximately 60% sensitivity, and, ROC$_{50}$: receiver operating characteristic integrated up to the fiftieth false positive. Performance figures in **bold** are the better of the two. For FP$_{60}$, the sensitivity levels (true positives/total positives) are: 11/19 (*Drosophila*), 5/9 (LSF) and 16/27 (muscle, muscle’).

The predictive accuracy of the two search methods is shown in Table 2. The relative performance of the methods depends on the accuracy metric used. In terms of FP$_{60}$, MCAST performs better in three cases and worse in one. In terms of ROC$_{50}$ each method performs better on two motif sets, and worse on two.

The most striking result in Table 2 is with the *Drosophila* motifs. MCAST makes at least four times fewer false positive predictions than COMET at a sensitivity of 60%. With that motif set as well as with the LSF motif set, MCAST achieves higher ROC$_{50}$ values than COMET. On the other hand, COMET has better ROC$_{50}$ than MCAST with the muscle and muscle’ motif sets, despite having poorer FP$_{60}$ on one of them. This discrepancy highlights the difference in the two accuracy measures. Whereas FP$_{60}$ puts emphasis on the selectivity at a given sensitivity (60%), ROC$_{50}$ takes both selectivity and sensitivity into account, ignoring performance beyond the fiftieth false positive.

The predictive accuracies of MCAST and COMET vary greatly as a function of their parameters. Figure 4 is typical of how ROC$_{50}$ varies with the settings of the two search parameters. Each point in the figure shows the accuracy of a MCAST or COMET search with the *Drosophila* data. Curves labeled $p = 0.001$ etc. show ROC$_{50}$ for MCAST searches with the given p and values of L shown on the x-axis. The two curves labeled $w = 75$ and $w = 1200$, respectively, show ROC$_{50}$ for COMET searches on the *Drosophila* data using the given window sizes and various values of the expected gap length parameter, a, shown on the x-axis.

The discrepancy in predictive accuracy between MCAST and COMET is much larger (in MCAST’s favor) when only the default value of COMET’s window size parameter ($w = 75$) is used. In that case, with the best setting for the other COMET parameter (expected gap length, a), COMET’s FP$_{60}$ value on the *Drosophila* dataset is 59, sixty-eight times worse than MCAST’s. As can be seen in Figure 4, COMET’s ROC$_{50}$ is also much worse with the

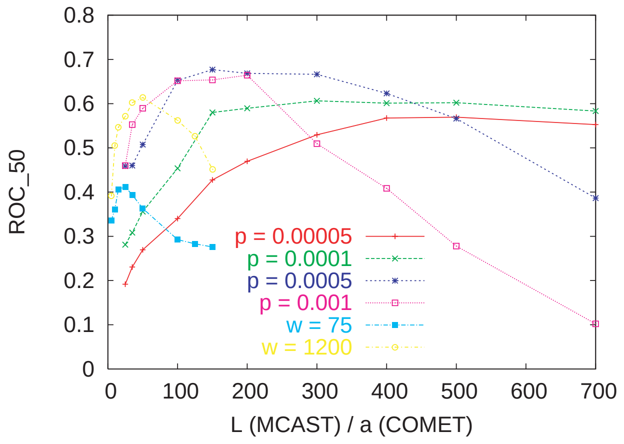

Fig. 4. ROC$_{50}$ as a function of the search parameters. The *Drosophila* dataset is searched using the *Drosophila* motifs. The first four curves are for MCAST searches and show how ROC$_{50}$ varies as a function of the maximum gap length (L) with the p-value threshold (p) held constant at different values. The last two curves show ROC$_{50}$ as a function of COMET’s expected gap length parameter (a) using the default window size ($w = 75$) and the experimentally-determined optimal size ($w = 1200$).

default window size. This points out the importance of trying different window sizes when using COMET.

On the *Drosophila* data, COMET achieves its best value of FP$_{60}$ (Table 2) and ROC$_{50}$ (Fig. 4) when its window size parameter, w, is set to 1200. The average length of the *Drosophila* dataset sequences is about 2000bp. This leads us to speculate that COMET’s technique of re-estimating the background distribution within a sliding window is responsible for its reduced accuracy (vis-a-vis MCAST) on the *Drosophila* dataset.

Speed

We have optimized the design of MCAST for searching large sequence databases. For databases containing more than one hundred thousand base pairs, search time scales nearly linearly in the size of the database and the total number of columns in the motifs making up the query. For example, on an 800Mhz Pentium PC running Linux, MCAST can search about 10 million base-pairs per motif column per second. Thus, a search of 10 million base pairs with five motif query (containing a total of 49 motif columns) takes about 49 seconds. This is approximately 100 times faster than the same search using COMET using its default parameters.

DISCUSSION

We first introduced the idea of using motif-based hidden Markov models to model biological sequences as the Meta-MEME algorithm (Grundy *et al.*, 1997). In the current work, we explore extensions to Meta-MEME specifically tailored to the problem of searching for regulatory modules. These extensions include a new model, a novel scoring function, a new search algorithm optimized for finding multiple clusters of motifs in long sequences, and a method for determining alignment E-values. We have shown that the new search algorithm, MCAST, is efficient and accurate at recognizing occurrences of regulatory modules in genomic DNA. In simulation, the algorithm produces accurate E-values. With real datasets, MCAST often performs substantially better on the task of locating regulatory modules than the COMET algorithm, which also uses motif-based HMMs.

There are three major differences between the MCAST and COMET algorithms. MCAST introduces the concept of p-scores, and MCAST alignments thus contain only statistically significant hits to the motifs in the query. MCAST uses a fixed background model in contrast to COMET's sliding-window background. This results in MCAST being about one hundred times faster than COMET when using its default (width 75) window. Unfortunately, it also results in MCAST being more prone to false positives (with very low E-values) in vertebrate sequences due to isochores. A more subtle distinction between the two algorithms is that MCAST uses the repeated match algorithm (Durbin *et al.*, 1998, p. 24–25) rather than the Waterman-Eggert algorithm (Waterman and Eggert, 1987). In the future, we may add a Waterman-Eggert search algorithm to the Meta-MEME toolkit. In conjunction with log-odds scoring (the default for Meta-MEME) and a module for computing Waterman-Eggert p-values (awaiting publication of a detailed description of the method sketched in Frith *et al.* (2002), this would yeild essentially the COMET algorithm (minus the sliding-window background model).

Other related work

Three classes of algorithms for recognizing regulatory modules have been proposed. Algorithms in the first class use a sliding window approach, scoring each subsequence that appears in the window with respect to a given collection of motifs. Examples include PromoterScan (Prestridge, 1995), FunSiteP (Kondrakhin *et al.*, 1995), ModelInspector (Frech *et al.*, 1997), CIS-ANALYST (Berman *et al.*, 2002), FLY ENHANCER (Markstein *et al.*, 2002), and several other methods (Levy and Hannenhalli, 2002). Typically, a window-based algorithm requires that the user specify three parameters: the width of the sliding window, a threshold for determining when a weak match to a given motif is considered genuine, and the minimum number of motif occurrences required to appear in a given subsequence. The sliding window approach has intuitive appeal, and has recently yielded good results in analyses of motif clusters in *Drosophila* (Berman *et al.*, 2002; Markstein *et al.*, 2002).

Our work falls into the second class of algorithms, which use hidden Markov models (HMMs) to address several of these difficulties (Crowley *et al.*, 1997; Frith *et al.*, 2001, 2002). HMMs automatically disallow overlapping motifs, and the HMM framework makes it easy to specify parameters that trade off the quality versus proximity of motif occurrences. An additional benefit of HMMs is their ability to learn parameters directly from a set of observations, freeing the user from having to set the parameters *a priori*. Unfortunately, the relative paucity of knowledge and data concerning the exact locations of many binding sites renders such a learning approach infeasible at this time. Consequently, previous models, as well as our own, require that the user specify the trade-off parameters, rather than allowing the parameters to be learned from data (Frith *et al.*, 2001, 2002).

Although the HMM approach has several benefits, it also brings with it two significant disadvantages. Both are related to the Markov property. First, within the HMM framework, it is very difficult to impose distal constraints. Thus, for example, a sliding window approach has some intuitive appeal, because the regulatory machinery must usually be physically proximal to the start of transcription. However, enforcing the constraint that a predicted regulatory module have a total length of at most n bases is (nearly) impossible in a Markov framework. The second drawback is related to the first: Markov processes in general have a difficult time accurately modeling variable-length sequences. A simple state with a self-transition will generate sequences whose lengths are geometrically distributed. Multi-state models can yield more complex length distributions, but the geometric distribution recurs frequently. This latter problem motivates one of the ways in which our model violates the probabilistic HMM framework: we avoid a geometric distribution of inter-module sequence lengths by removing the length distribution in this portion of the model.

Both the sliding window and the HMM approaches to the regulatory module search problem are generative: both rely upon a model (implicit or explicit) of a regulatory module. The third class of algorithms uses a discriminative technique. These methods model the difference between the regulatory module and non-regulatory sequence. Logistic regression analysis (LRA) (Wasserman and Fickett, 1998; Krivan and Wasserman, 2001) is a discriminative technique based upon a sliding window. The Fisher kernel SVM method (Pavlidis *et al.*, 2001) uses a discriminative algorithm based upon a hidden Markov model. In the presence of a small amount of data, discriminative techniques typically achieve better performance than similar, generative techniques. However, the discriminative approach necessarily requires as input a collection of sequences that are known not to contain regulatory modules, and an accurate such collection is at this time difficult to obtain.

Future work

Searching for regulatory modules remains a difficult, unsolved problem. As pointed out by Frith *et al.* (2002), many known, biologically active regulatory modules are not statistically significant when modeled as unordered clusters of motifs. One potential solution is to seek ways to focus the search onto a smaller subset of the genome, thus increasing the significance of any clusters found. A second solution might be to post-process the clusters to remove clusters that don't satisfy some set of constraints (e.g., that don't contain some specific subset of motifs). Yet another approach is to attempt to introduce ordering and spacing constraints into the model. This may become feasible as more data on regulatory modules become available.

REFERENCES

Bailey,T.L. (2003) *Details of MCAST statistics*, Technical Report IMB TR0001, Institute for Molecular Bioscience, University of Queensland, http://www.maths.uq.edu.au/∼tlb.

Bailey,T.L. and Elkan,C.P. (1994) Fitting a mixture model by expectation-maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, pp. 28–36.

Bailey,T.L. and Gribskov,M. (1998) Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, **14**, 48–54.

Bailey,T.L. and Gribskov,M. (2002) Estimating and evalutating the statistics of gapped local-alignment scores. *J. Comp. Biol.*, **9**, 573–593.

Benson,G. Tandem repeats finder: a program to analyze dna sequences. *Nucleic Acids Res.*, **27**, 573–580.

Berman,B.P., Nibu,Y., Pfeifer,B.D., Tomancak,P., Celniker,S.E., Levine,M., Rubin,G.M. and Eisen,M.B. (2002) Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the *Drosophila* genome. *Proc. Natl Acad. Sci USA*, **99**, 757–762.

Bucher,P. (1990) Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol. Biol.*, **212**, 563–578.

Crowley,E.M., Roeder,K. and Bina,M. (1997) A statistical model for locating regulatory regions in genomic DNA. *J. Mol. Biol.*, **268**, 8–14.

Dempster,A.P., Laird,N.M. and Rubin,D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.*, **39(1)**, 1–38.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis*. Cambridge UP.

Frech,K., Danescu-Mayer,J. and Werner,T. (1997) A novel method to develop highly specific models for regulatory units detects a new LTR in GenBank which contains a functional promoter. *J. Mol. Biol.*, **270**, 674–687.

Frith,M.C., Hansen,U. and Weng,Z. (2001) Detection of *cis*-element clusters in higher eukaryotic DNA. *Bioinformatics*, **17(10)**, 878–889.

Frith,M.C., Spouge,J.L., Hansen,U. and Weng,Z. (2002) Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Res.*, **30(14)**, 3214–3224.

Gribskov,M. and Robinson,N.L. (1996) The use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comp. & Chem.*, **20**, 25–33.

Grundy,W.N., Bailey,T.L., Elkan,C.P. and Baker,M.E. (1997) Meta-MEME: Motif-based hidden Markov models of protein families. *Comp. Appl. Biosci.*, **13(4)**, 397–406.

Klingenhoff,A., Frech,K., Quandt,K. and Werner,T. (1999) Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity. *Bioinformatics*, **15**, 180–186.

Kondrakhin,Y.V., Kel,A.E., Kolchanov,N.A., Romashchenko,A.G. and Milanesi,L. (1995) Eukaryotic promoter recognition by binding sites for transcription factors. *Comp. Appl. Biosci.*, **11**, 477–488.

Krivan,W. and Wasserman,W. (2001) A predictive model for regulatory sequences directing liver-specific transcription. *Genome Res.*, **11**, 1559–1566.

Krogh,A., Brown,M., Mian,I., Sjolander,K. and Haussler,D. (1994) Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.

Levy,S. and Hannenhalli,S. (2002) Identification of transcription factor binding sites in the human genome sequence. *Mamm. Genome*, **13**, 510–514.

Markstein,M., Markstein,P., Markstein,V. and Levine,M.S. (2002) Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the *Drosophila* embryo. *Proc. Natl Acad. Sci USA*, **99(2)**, 763–768.

Mouse Genome Sequencing Consortium, (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**, 520–562.

Pavlidis,P., Furey,T.S., Liberto,M., Haussler,D. and Grundy,W.N. (2001) Promoter region-based classification of genes. In *Proceedings of the Pacific Symposium on Biocomputing*. pp. 151–163.

Prestridge,D. (1995) Predicting Pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, **249**, 923–932.

Ptashne,M. and Gann,A. (2002) *Genes and Signals*. Cold Spring Harbor Laboratory Press.

Schaffer,A., Aravind,L., Madden,T.L., Shavirin,S., Spouge,J.L., Wolf,Y.I., Koonin,E.V. and Altschul,S.F. (2001) Improving the accuracy of psi-blast protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.

Viterbi,A.J. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Info. Theory*, **IT–13**, 260–269.

Wasserman,W.W. and Fickett,J.W. (1998) Identification of regulatory regions which confer muscle-specific gene expression. *J. Mol. Biol.*, **278**, 167–181.

Waterman,M. and Eggert,M. (1987) A new algorithm for best subsequence alignments with application to tRNA-tRNA comparisons. *J. Mol. Biol.*, **197**, 723–725.

Wingender,E., Chen,X., Hehl,R., Karas,H., Liebich,I., Matys,V., Meinhardt,T., Pruss,M., Reuter,I. and Schacherer,F. (2000) TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Res.*, **28**, 316–319.