A probabilistic method to detect regulatory modules

Saurabh Sinha, Erik van Nimwegen and Eric D. Siggia*

Center for Studies in Physics and Biology, The Rockefeller University, New York, NY 10021, USA

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: The discovery of cis-regulatory modules in metazoan genomes is crucial for understanding the connection between genes and organism diversity.

Results: We develop a computational method that uses Hidden Markov Models and an Expectation Maximization algorithm to detect such modules, given the weight matrices of a set of transcription factors known to work together. Two novel features of our probabilistic model are: (i) correlations between binding sites, known to be required for module activity, are exploited, and (ii) phylogenetic comparisons among sequences from multiple species are made to highlight a regulatory module. The novel features are shown to improve detection of modules, in experiments on synthetic as well as biological data.

Availability: The source code for the programs is available upon request from the authors.

Contact: {saurabh, erik, siggia}@lonnrot.rockefeller.edu

Keywords: hidden Markov model, cis-regulatory modules, motif correlations, phylogenetic comparison

INTRODUCTION

Many genes in multicellular organisms exhibit complex patterns of expression in space and time. These patterns are mediated by a combinatorial code of transcription factors that bind to cis-regulatory regions of the genome. These regulatory regions (100–1000 bp in length), often termed modules, can be moved from their native context and still recapitulate a portion of the native expression pattern, thus acting as autonomous units (Davidson, 2001; Carroll et al., 2001). Their importance for understanding evolution has grown with the realization that evolutionary novelty (particularly over short times) issues more from changes in regulation than from creation of new protein coding regions, a paradigm that is perhaps best evident in the context of development. This paper addresses the important problem of discovering such regulatory modules computationally, since it is vastly quicker to check a computationally predicted module for functionality than to dissect all the non-coding sequence around a gene for regulatory potential. The approach taken here is guided by the principles deduced from studies of the fruit fly, which appear broadly applicable.

Modules in fly typically have multiple binding sites for each transcription factor, with 3–6 different factors contributing (Davidson, 2001; Carroll et al., 2001). Several groups (Berman et al., 2002; Halfon et al., 2002; Rajewsky et al., 2002) have shown, for early development in the fly, that modules can be identified by searching for clusters of binding sites (‘motifs’) for groups of transcription factors known to work together, without regard to order or spacing. Each of these methods scans the genomic sequence with the input set of motifs, and a scheme to detect if the local sequence shows a clustering of these motifs. While Berman et al. (2002) and Halfon et al. (2002) use specific rules on counts of motifs, Rajewsky et al. (2002) have proposed the use of Hidden Markov Models (HMMs) and Expectation Maximization (EM) on the parameters controlling the number of each motif. The HMM (also used in this context in Frith et al. (2001)) supplies a statistically sound measure of the likelihood that the data is derived from a model (e.g. giving some weight to multiple weak occurrences that individually fall below a cutoff). The EM step greatly enhances the discrimination by concentrating weight on the relevant factors, leaving no parameters to be adjusted by hand. This work extends the HMM model to utilize two important sources of information in detecting regulatory modules—(i) correlation between binding sites in a module, and (ii) comparison of multiple species using a statistical model for motif evolution.

The first extension is motivated by cases were modules predicted by the above methods appear suitable, yet where something in the arrangement of the sites (or a wider context) renders the module non-functional. Also, many cases are known where factors must interact with a cofactor to be functional, and spatial correlations between their binding sites are observed. For example, the repression of zen in ventral regions of the Drosophila blastoderm has been attributed to closely spaced binding
sites for the transcription factors dl and dri in the zen promoter (Mannervik et al., 1999). Unlike previous methods, the ‘history-conscious’ HMM developed in this paper has explicit parameters to capture such correlations, and gives greater weight to a module where one motif consistently follows another, than another module where the same motifs are arranged at random. Our work takes a first step toward deducing the ‘grammar’ rules that define a functional module, with the goal of using them for genome-wide module detection. Similar ideas have been used in earlier work on motif detection (e.g. Grundy et al., 1997; GuhaThakurta and Stormo, 2001; Sinha, 2002).

Interspecies comparisons are recognized as an important resource for discovery of regulatory modules. The genomes of human, mouse, and rat are close enough to highlight these modules, and we will soon have multiple species of yeast and two fly genomes to examine. But how to consistently score binding site matches in both conserved and unconserved regions from multiple species at variable evolutionary distances (e.g. mouse and rat are much closer than either to human) is still an open question (Loots et al., 2002). The method proposed here looks at homologous sequence windows spanning multiple species, and in scoring a window treats its conserved and unconserved portions in different ways. Binding site matches in conserved blocks are evaluated using a suitable phylogenetic model, and matches in all unconserved but homologous regions also contribute to the score.

The program developed here is called Stubb. The target application is to run it genome-wide, possibly using multiple species data, and extract the top few predicted modules to be tested for function. We perform preliminary tests to demonstrate the advantage of the new methods over previous approaches, using both synthetic and biological data.

**ALGORITHM**

The fundamental operation in the Stubb system is to score one (or several, related by descent) DNA sequence(s) S with a set W of motifs. The particular score used is a log likelihood ratio, and reflects how likely it is that S was generated by a probabilistic process that uses the motifs of W, as compared to being generated by a random background process. The score will be high for sequences that have a cluster of motifs from W. The motifs in W are described by their ‘Position Weight Matrices’ (PWM’s) and can be of varying lengths.

To scan a genomic sequence for cis-regulatory modules formed by a motif set W, the algorithm proceeds from one end to the other (e.g. 5’ to 3’), looking at successive sequence ‘windows’ of a fixed (parameterized) length L, which define S. Each window is scored with W, and the series of scores is output. The next two sections describe the probabilistic model used in computing the window score, with and without motif correlations. The following section extends the probabilistic framework to include sequence and phylogenetic information from multiple species.

**Hidden Markov models**

The probabilistic process that is assumed to generate the sequence S is described by a Hidden Markov Model (HMM). At each step, the process chooses either a motif w_i at random from the set W, or the background motif w_b. This choice is dictated by the transition probabilities p_i of the motifs, which are model parameters. Once the process has chosen a motif w, it samples a sequence from the PWM of w, appends it at the end of the sequence S created so far, and proceeds to the next step. The process stops when the length of S reaches L. The sequence of motifs chosen in the successive steps of the process is called a ‘parse’ of the sequence. The model parameters θ, which include the transition probabilities {p_i} and the motif set W, associate a well-defined probability Pr(S, T|θ) with each parse T of the sequence S. The probability that S was generated by an HMM with parameters θ is then given by

Pr(S|θ) = \sum_T Pr(S, T|θ)

Let Pr(S|θ_b) be the probability of generating S by using only w_b. For a given θ, we define

F(S, θ) = \log \frac{Pr(S|θ)}{Pr(S|θ_b)}

This is the function used by Stubb to score a sequence S. However, θ is not a parameter to the algorithm, it is chosen so as to maximize F(S, θ).

In this simple version of the HMM, there is a single transition probability p_i associated with each motif w_i, including the background motif w_b, with the constraint that \sum_i p_i = 1. The background motif w_b is a special kind of PWM—it is of length 1, but the sampling probability of a base depends upon the previous k bases in the sequence. (k is a fixed non-negative integer and is called the Markov order of the background motif.) The HMM described so far is exactly the probabilistic model used by Rajewsky et al. (2002), and we shall henceforth refer to it as HMM0.

An essential component of the score computation mentioned above is the subsequence probability Pr(s|w). This is the probability of generating the subsequence s of length l (length of w), when sampling from w. A simple way to compute this probability is as follows: let s = s_1s_2...s_l, and let w_{ij} be the probability of sampling base i at the j^{th} position of the PWM w. Then, Pr(s|w) = \prod_{i=1}^{l} w_{s_i,j}.

This assumes that when a motif w is sampled and planted,
its orientation must match the direction in which the sequence is generated, called the ‘forward’ direction. If we also wish to allow occurrences of \( w \) in the reverse direction (for binding sites on the complementary strand of DNA), the subsequence probability will be of the form \( b_w \Pr(s|w) + (1-b_w) \Pr(s^R|w) \), where \( s^R \) is the reverse complement of \( s \), and \( b_w \) captures the strand bias of the motif \( w \). For instance, the case \( b_w = 1 \) represents the prior belief that \( w \) can occur only in the forward direction, while \( b_w = 0.5 \) means that the motif has no strand preference. Henceforth, the term \( \Pr(s|w) \) will represent the overall subsequence probability of \( s \) given \( w \), computed using appropriate strand biases.

As mentioned earlier, for each sequence window \( S \), the algorithm finds the \( \theta \) that maximizes \( F(S, \theta) \). This training of parameters is done by the Baum-Welch algorithm (Baum, 1970), which uses Expectation-Maximization (EM) to iteratively converge to a locally optimum \( \theta \). (See the Appendix for details.) The update procedure is guaranteed to improve the value of \( F(S, \theta) \) in each iteration, until convergence. The value of \( \log \Pr(S|\theta) \), for a given \( \theta \), is computed using dynamic programming (the Forward-Backward algorithm, Durbin et al., 1999). In most biological examples examined to date, the global maximum is found, as evidenced by the smooth change in score with incremental change in window position.

It should be noted that an HMM, as described here, does not use thresholds to determine motif occurrences in the input sequence. Weak motif occurrences also contribute to the score, albeit less significantly than strong occurrences. We also note that the motif set \( W \) is typically of size 1–20, and that overfitting may become a problem with larger motif sets, given the typical sequence window size of less than 1000 bp.

### Motif correlations

The HMM0 assumes a statistical model for the data in which the motifs and background bases are placed independently. In reality motifs may be correlated both in order (e.g. \( w_j \) follows \( w_i \)) and in space. For instance, a module with few occurrences of \( w_j \) may be functional only if these occurrences are in the vicinity of \( w_i \). We therefore add to \( \theta \) a correlated transition probability \( p_{ij} \), with the interpretation that when the model chooses a motif to place, if the previous non-background motif placed is \( w_i \) then motif \( w_j \) is chosen with probability \( p_{ij} \). This ‘history-conscious HMM’ (hcHMM), only the previous non-background motif is ‘remembered’ (and in this way our model differs from the canonical first order HMM).

### Detecting correlations.

Firstly, the algorithm has to decide if the parameter \( p_{ij} \) for a specific pair \( \{w_i, w_j\} \) should be included in \( \theta \). Including \( p_{ij} \) for all pairs of motifs makes the number of parameters large, and may cause overfitting of the model for typical input dimensions. Hence \( p_{ij} \) is added to \( \theta \) only if there is evidence for a correlation in occurrences of \( w_i \) and \( w_j \), detected as follows.

Consider a random sequence \( X \) of length \( L \) generated by an HMM0 with parameters \( \theta \). Let \( A_{ij}(X) \) be the average number of times \( w_j \) follows \( w_i \) (with nothing except background in between) in a parse of \( X \), the average being taken over all parses of \( X \) weighted with their respective probabilities. Let \( E_{ij} \) and \( \sigma_{ij} \) be the expectation and standard deviation of the random variable \( A_{ij}(X) \), over all \( L \)-length sequences \( X \). Then we can define a test statistic

\[
Z_{ij} = \frac{A_{ij}(S) - E_{ij}}{\sigma_{ij}}
\]

(1)

to measure how different the observed number of (ordered) paired occurrences \( A_{ij} \) is from what is expected by chance in a random sequence. Computation of \( E_{ij} \) and \( \sigma_{ij} \) is described in the Appendix. The computations have \( O(L) \) time complexity. The parameter \( p_{ij} \) is added to the model only if \( Z_{ij} \) is above a threshold and \( E_{ij} \) is also above a threshold.

### Training a history-conscious HMM.

The second technical challenge is to train the parameters of the hcHMM. We begin by describing all the transition probabilities \( \Pr(i \rightarrow j) \), which is the probability of choosing a \( w_j \) following a \( w_i \) with nothing except background in between. The parameter \( \theta \) includes all possible \( p_i \) and some or all \( p_{ij} \). Let \( \text{Corr}(i, j) = \text{true} \) if and only if correlation was detected for \( \{w_i, w_j\} \). If \( \text{Corr}(i, j) = \text{true} \), then \( \Pr(i \rightarrow j) = p_{ij} \). For all \( i, j \) such that \( \text{Corr}(i, j) = \text{false} \),

\[
\Pr(i \rightarrow j) = p_j \left( 1 - \sum_{k} I(\text{Corr}(i, k) = \text{true}) p_{ik} \right) / \sum_{k} I(\text{Corr}(i, k) = \text{false}) p_{jk}
\]

Here, the parameter \( p_j \) is normalized appropriately to ensure that \( \sum_j \Pr(i \rightarrow j) = 1 \). Given that \( w_i \neq w_b \) is the previous non-background motif planted, a motif \( w_j \in W \cup \{w_b\} \) is planted with probability \( \Pr(i \rightarrow j) \). The spacing between motifs is thus controlled by the exponential decay of powers of \( \Pr(i \rightarrow b) \). If no non-background motif has been planted so far, \( w_j \in W \cup \{w_b\} \) is planted with probability \( \Pr(b \rightarrow j) \). Note that if \( \text{Corr}(i, j) = \text{false} \) for all \( j \), then \( \Pr(i \rightarrow j) = p_j \), i.e. the \( p_j \)'s have the same semantics as in HMM0.

To our knowledge, the Baum-Welch algorithm for training HMM parameters does not have a simple extension to the history-conscious HMM described here. We derive an update criterion for \( \theta \) that, following the EM theory, is guaranteed to improve the objective function \( F(S, \theta) \) in each iteration, until convergence. The calculations are
Probabilistic method to detect regulatory modules

outlined in the Appendix. In the extreme cases when \( \text{Corr}(i, j) \) is true for all \( i, j \) or for none, the update formulae reduce to the standard Baum-Welch updates.

The overall algorithm to score \( S \) using pairwise motif correlations is summarized below.

Algorithm ComputeScore
Input: Sequence \( S \), motif set \( W \cup \{w_{h}\} \), real numbers \( \tau_{c}, \tau_{e} \); Output: Score of \( S \).
1. Set \( \text{Corr}(i,j) = \text{false} \) for all pairs \( i, j \). Set \( \theta \) to include all \( p_i \), but no \( p_j \).
2. Train \( \theta \) so as to maximize \( F(S, \theta) \).
3. For each pair \( (i,j) \) such that \( w_i \in W \) and \( w_j \in W \) do
   4. Use the trained \( \theta \) to compute \( Z_{ij} \) using Formula 1.
   5. If \( Z_{ij} > \tau_c \) and \( E_{ij} > \tau_e \), set \( \text{Corr}(i,j) = \text{true} \).
   6. End For
7. Set \( \theta \) to include all \( p_i \) and all \( p_j \) for which \( \text{Corr}(i,j) = \text{true} \).
   8. If \( \text{Corr}(i,j) = \text{false} \) for all \( i, j \), output the maximum \( F(S, \theta) \) computed in Step 2, else
   9. Train \( \theta \) so as to maximize \( F(S, \theta) \), output this maximum as the score of \( S \).

Note that correlations with \( w_h \) are not detected, hence \( p_{ih} \) or \( p_{hi} \) is never trained, for any \( i \). Each iteration in the training of the hHMM (Step 9) has \( O(L|W|^2) \) time complexity, instead of the \( O(L|W|) \) complexity in HMM0. However, Step 9 is executed only if a correlation was detected, hence the genome-wide running time only changes marginally. Algorithm ComputeScore deploys the motif correlation detection (Steps 3–6) on the input sequence \( S \). However, our implementation allows this detection procedure to be run separately, on a potentially different sequence that serves as training data.

The input genomic sequence is parsed into a series of overlapping windows of length \( L \) each, whose starting positions differ by a parameterized shift-size \( \delta \), and each window \( S \) is scored by the above algorithm.

Multiple species and phylogenetic information
In this section, we extend \( \text{Stubb} \) to utilize phylogenetic comparisons between sequences from multiple species. Many of the species used for such comparisons are sufficiently closely related (e.g. mouse and rat, or various budding yeasts—Cliften et al., 2001) that the neutral point mutations would not have had time to randomize non-functional sequence. Thus some degree of sequence correlation is expected by chance, and must be taken into account. Secondly, regulatory sequence does not just change one base at a time (e.g. for fly, see Bergman and Kreitman, 2001). For instance, when homologous regulatory regions are compared between fly species, there are obvious conserved blocks in the 30–100 bp range, much larger than a protein binding site, yet much smaller than a typical module. Between them sits unaligned sequence in a comparable size range. Experimentally known binding sites for early development occur in both these regions (E. Emberly, personal communication). A binding site found in an aligned block clearly carries some additional significance. On the other hand, if clusters of binding sites occur outside of the blocks, in any species, their functionality should also be explored. Hence \( \text{Stubb} \) considers both aligned blocks and the unaligned sequences between them in scoring a window.

Sequences from multiple species are scored in two steps. The first step finds prominent (ungapped) conserved blocks of sequence and constructs the best syntetic parse of the entire sequence into such blocks. For this purpose, we use DiAlign (Morgenstern et al., 1998) when dealing with more than two species, and Lagan (Brudno et al., 2003) for two species. In the second step, the blocks are used to define homologous windows between the species. A homologous window may contain one or more consecutive aligned blocks, and unaligned sequences sitting between two adjacent blocks in the window are also included. Each block in a homologous window is then scored as a unit, taking into account the neutral point mutation rate. The score from these blocks, along with independent contributions from the unaligned sequences in the window (computed as for single species), comprises the total score of this window. Small blocks that are missed, or non-syntenous blocks (rare in the flies we compare) are not a serious problem. Matrices resident there will still be scored but as independent events.

Parsing windows. We first detail how \( \text{Stubb} \) creates homologous windows when there are only two species \( A \) and \( B \) with regulatory sequences \( S_A \) and \( S_B \) for a common gene. It takes \( S_A \) as a reference, and marks off successive \( L \)-length windows spaced by \( \delta \ll L \). Suppose a window \( X \) from \( S_A \) contains a set of non-overlapping subsequences \( \{x_1, x_2, \ldots, x_k\} \) aligned with similar subsequences \( \{y_1, y_2, \ldots, y_k\} \) of \( S_B \). The corresponding homologous window then has the following components: (i) aligned blocks \( (x_i, y_i) \), for \( i = 1 \ldots k \), (ii) all subsequences of \( X \) outside the aligned regions, and (iii) the unaligned sequences of \( S_B \) between \( y_i \) and \( y_{i+1} \) (for \( i = 1 \ldots k - 1 \)). Thus, if there is only one aligned region \( x_1 \) within \( X \), the contribution from \( S_B \) consists only of \( y_1 \), and if there is no block in \( X \) then \( S_B \) does not contribute to the score.

Scoring an aligned block. Aligned blocks in a homologous window are scored as a unit, as described next. All the sequences in an aligned block derive from a common ancestor, and our weight matrices are assumed to apply to the common ancestor and all descendants, a reasonable assumption given the propensity for modules to re-
tain function when moved between species. For simplicity, we assume the species are related by a star topology. To apply the HMM we need to generalize the expression for \( Pr(s|w) \) to a set \( \sigma \) of subsequences, each of which occupies the homologous position in the aligned block. Our evolutionary model assumes that all bases evolve independently, at equal rates, and that the probability of fixation of a mutation \( \alpha \rightarrow \beta \) at position \( i \) is proportional to the weight matrix entry of \( \beta \) at that position.

Under these assumptions, we have

\[
Pr(\sigma|w) = \prod_{i=1}^{l} \left[ \sum_{\beta \in \Sigma} w_{\beta i} \prod_{s \in \sigma} (w_{s i} \mu_{s} + (1 - \mu_{s}) \delta_{s i}) \right] \tag{2}
\]

where \( l \) is the length of \( w \), \( \Sigma = \{A, C, G, T\} \), \( w_{\beta i} \) is the probability of emitting \( \beta \) at the \( i^{th} \) position of \( w \), \( \delta_{xy} = 1 \) if \( x = y \) and 0 otherwise, and \( \mu_{s} = 1 - e^{-\lambda t_{s}} \) is a function of the neutral mutation rate \( \lambda \) and the evolution time \( t_{s} \) between the ancestor and the species \( s \). For each position \( i \), one ‘creates’ a base \( \beta \) in the ancestor with frequency \( w_{\beta i} \), and each such base is either passed on to species \( s \) unchanged (probability \( 1 - \mu_{s} \)) or mutated with probability \( \mu_{s} \) and a new base selected with a frequency defined by \( w \). If \( \mu_{s} \) is small (as for very closely related species), then finding different bases in homologous positions strongly suppresses \( Pr(\sigma|w) \), even if their frequency in \( w \) is the same. For \( \mu_{s} \approx 1 \), the sum over the ancestor base (\( \sum_{\beta \in \Sigma} w_{\beta i} \)) is replaced by 1, and the sequences in \( \sigma \) are scored as independent, with the important caveat that the relative alignment of the sequences is fixed. Our model is translationally invariant in time since if there are only two species the sum over the ancestor base can be done explicitly and (2) reduces to

\[
Pr(\sigma|w) = \prod_{i=1}^{l} \left( w_{s_{1 i}} (w_{s_{2 i}} \mu + (1 - \mu) \delta_{s_{2 i}}) \right), \tag{2}
\]

where \( s_{1}, s_{2} \) are the bases in the two species at position \( i \), and \( \mu \) is composed from the total time of evolution between them.

**Implementation**

The \textit{Stubb} system is implemented in C++, and can scan the entire fly genome with a set of ~15 weight matrices in a day on a work station. Its scores are used to rank windows as putative modules, with the expectation that there will be one to several hundred per genome. Typically, the windows in the neighborhood of a high-scoring window are also high-scoring. Hence, when reporting a high-ranking window, \textit{Stubb} suppresses all overlapping windows that score less than it.

To compute the strand bias \( b_{w} \) of a motif \( w \), \textit{Stubb} counts the ‘occurrences’ of \( w \) (using a weak threshold) in \( S \) in both directions, in a pre-processing step, and uses strand bias in proportion to these counts. To derive the background motif \( w_{b} \) to be used in scoring a window (of length \( L \)), \textit{Stubb} first constructs a ‘context’ window \( C \) of length \( rL \) (\( r \) is a configurable parameter) from the current window and its flanking regions. For an order \( k \) background model, the frequencies of all \((k+1)\)-mers in \( C \) are used to derive \( w_{b} \).

The first step in scoring sequences from two species involves computation of the best syntenic parse of conserved blocks. For this purpose, we run the Lagan alignment tool of Brudno et al. (2003). Given two long sequences, this tool computes ‘anchors’ of local similarity between the sequences, puts together the best syntenic series of such anchors, and uses dynamic programming to align the regions between anchors. \textit{Stubb} takes two sequences aligned in this manner, and extracts all untagged, aligned blocks of a certain minimum size and percent-identity to serve as the blocks of common descent.

**EXPERIMENTS**

We first tested the effect of using \textit{hcHMM} on synthetic sequences in which two motifs were planted with varying degrees of correlation. An \textit{hcHMM} was used to create the random sequences, using parameters \( p_{1} = p_{2} = 0.01, p_{12} = (1 + c)p_{2} \), where \( c \geq 0 \) parameterizes the correlation. (All other parameters are defined by normalization.) For long (e.g. 10 kb) data sets, \textit{Stubb} (\textit{hcHMM}, with parameter \( p_{12} \)) recovered the input value of \( c \) to within the fluctuations expected for ~100 samples of motif 1 in the data. For shorter sequences, indicative of the window sizes we use for scanning the fly genome, fluctuations were larger. For purposes of detecting modules, even small differences between the HMM0 and \textit{hcHMM} scores are meaningful since the same data is being compared. For instance, with \( L = 500 \) the score difference normalized by the HMM0 score was 0.038 for \( c = 3 \) and 0.096 for \( c = 20 \). In the absence of correlation (\( c = 0 \)), this was only 0.014, indicating that the extra parameter in the \textit{hcHMM} had little effect on the score in this case.

We next constructed a toy example from yeast regulatory sequences, again to test \textit{hcHMM}. The transcription factors \textit{MCM1} and \textit{MAT\alpha2} are known to act cooperatively in the mating pathway (Mead et al., 2002). Six regulatory regions where \textit{MAT\alpha2} is known to occur were collected (Zhu and Zhang, 1999) and fit simultaneously with the matrices for these two factors. HMM0 gave a score of 31.3, while \textit{hcHMM} boosted this significantly to 54.5. After training of parameters, the transition probability matrix showed \( Pr(1 \rightarrow 1) = 0, Pr(1 \rightarrow 2) = 0.38, Pr(2 \rightarrow 1) = 0.005, Pr(2 \rightarrow 2) = 0.0003, Pr(1 \rightarrow b) = 0.62, Pr(2 \rightarrow b) = 0.995 \), where 1 represents \textit{MCM1}, 2 represents \textit{MAT\alpha2}, and \( b \) represents the background motif. The probabilities strongly suggest a motif structure \( \textit{MAT\alpha2}\rightarrow\textit{MCM1}\rightarrow\textit{MAT\alpha2} \) (with the first interval
(2 → 1) much longer than the second), in accord with experiment (Mead et al., 2002).

The next experiments focus on the gap gene system from the fly Drosophila melanogaster. As the input motif set \( W \), we use the set of PWMs for the transcription factors \( bcd, hb, kni, Kr, cd, dl \) and torER (Rajewsky et al., 2002). The 10 kb upstream regions of the genes \( eve, gt, kni, Kr, run, ill, \) and \( hb \), as well as the 12 kb upstream region of \( hairy \) are used as input sequences, in separate runs of the program, with \( \text{hcHMM}, L = 500, \delta = 20 \) and background Markov order 2. All top ranking windows with scores above 12.0, as well as all the modules known for these genes from the literature (as collected in Rajewsky et al. (2002)), are presented in Table 1. The last column measures the fractional overlap between the known and predicted modules. All 16 known modules are recovered by the program. Two additional modules are predicted—one with score 15.4 at coordinates 4140-4639 in \( kni \) and one with score 17.3 at 6540-7039 in \( run \). We are currently investigating if these are known \( cis \)-regulatory modules. This experiment represents the typical input to \( Stubb \), and the performance is extremely encouraging.

Table 2 shows situations where substantial correlation was observed, and used to boost the window score. Each row corresponds to a known module, followed by the highest scoring window (by \( \text{hcHMM} \)) near the module, its scores \( f_{hc} \) and \( f_0 \) by \( \text{hcHMM} \) and \( \text{HMM0} \) respectively, and the difference. The most significant correlation was discovered for \( zen \), a gene involved in dorsal-ventral patterning for which we used the PWM’s \( dl, twi, sna, \) \( brk, dri, \) and \( ntf \). The \( zen \) promoter is known to have a functional correlation between \( dl \) and a DNA binding cofactor \( dri \), which is precisely what \( Stubb \) reported. We expect that a large number of \( dl \) regulated modules will be reported from microarray experiments, and \( Stubb \) can be used to fit the entire set. Notice in Table 2 that the observed score differences are much higher than the average value of 1.4% found in the absence of correlations. (See experiment above, case \( c = 0 \).) We also observed that the scores for non-modules do not change significantly from \( \text{HMM0} \) to \( \text{hcHMM} \), and that the module for \( Kr \) predicted by \( \text{hcHMM} \) has a much better overlap with the known module than the \( \text{HMM0} \) prediction. (Data not shown.)

The next experiments are designed to test if using multiple species data improves the discrimination of modules from non-modules. Two versions of \( Stubb \) are run, one (called \( SSPECIES \)) using single species data and the other (called \( MSPECIES \)) using multiple species data. It is not meaningful to compare the absolute scores for a window from the two versions, since a homologous window typically contains more sequence data than the corresponding single species window. Hence, we design a score to measure the discrimination of a window from baseline scores. We first compute (for each version) the average baseline score \( b \) of a window of length \( L \) from sequences that do not have modules. The discrimination score of a window is \( r = (f - b)/b \), where \( f \) is the absolute score of this window. This measures the fractional increase in score from the baseline level, and can be used to compare the discrimination afforded by the two versions of \( Stubb \).

In one experiment, synthetic sequences of length \( L = 500 \) were created for two species, and two motifs were planted with \( p_i = 0.01 \). A motif planted in one species was conserved in the other species with probability 0.5, and a base in a conserved position was mutated with probability 0.1. The baseline scores for \( SSPECIES \) and \( MSPECIES \) were separately computed from random sequences that lacked these motifs. For each synthetic sequence, the discrimination scores \( r_s \) and \( r_m \) were computed for \( SSPECIES \) and \( MSPECIES \) respectively, and their difference was noted. \( r_m \) was about 2.6 units more than \( r_s \), averaged over 100 experiments, indicating that \( MSPECIES \) gives better discrimination of modules than

**Table 1.** Performance of \( Stubb \) (hcHMM) on gap gene upstream regions. The last column measures the fractional overlap between the known and predicted modules.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Predicted Modules</th>
<th>Score</th>
<th>Known Module</th>
<th>Overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>eve</td>
<td>2780-3279</td>
<td>27.9</td>
<td>2763-3273</td>
<td>0.98</td>
</tr>
<tr>
<td>gt</td>
<td>5100-5600</td>
<td>17.0</td>
<td>4974-5644</td>
<td>1.00</td>
</tr>
<tr>
<td>hairy</td>
<td>7360-7859</td>
<td>16.0</td>
<td>7242-8184</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>1340-1839</td>
<td>15.7</td>
<td>829-1760</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>2600-3099</td>
<td>32.7</td>
<td>2601-3147</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>5640-6139</td>
<td>12.3</td>
<td>5831-6132</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>7100-7599</td>
<td>18.6</td>
<td>6396-7551</td>
<td>1.00</td>
</tr>
<tr>
<td>kni</td>
<td>4140-4639</td>
<td>15.4</td>
<td>not known</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>6900-7399</td>
<td>23.2</td>
<td>6926-6992</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>7380-7879</td>
<td>28.7</td>
<td>7422-8998</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>5640-6139</td>
<td>18.2</td>
<td>5668-6389</td>
<td>0.94</td>
</tr>
<tr>
<td>Kr</td>
<td>60-559</td>
<td>15.2</td>
<td>37-862</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>6540-7039</td>
<td>17.3</td>
<td>not known</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>7140-7639</td>
<td>23.9</td>
<td>6997-7476</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>8420-8919</td>
<td>19.6</td>
<td>8564-8946</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>9400-9899</td>
<td>13.7</td>
<td>9418-9592</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2420-2919</td>
<td>16.6</td>
<td>2335-3357</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>9000-9499</td>
<td>14.0</td>
<td>8834-9554</td>
<td>1.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gene</th>
<th>Predicted Modules</th>
<th>Score</th>
<th>Known Module</th>
<th>Overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>giant</td>
<td>7242-8184</td>
<td>15.2</td>
<td>7360-7859</td>
<td>1.00</td>
</tr>
<tr>
<td>hairy</td>
<td>829-1760</td>
<td>15.4</td>
<td>1340-1839</td>
<td>1.00</td>
</tr>
<tr>
<td>kruppel</td>
<td>5668-6389</td>
<td>18.2</td>
<td>5640-6139</td>
<td>1.00</td>
</tr>
<tr>
<td>zen</td>
<td>2615-3016</td>
<td>13.0</td>
<td>2540-3039</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Table 2.** Advantage of hcHMM over HMM0 in detecting modules. \( f_{hc} \) \( \text{hcHMM} \) score, \( f_0 \) \( \text{HMM0} \) score, \( \Delta f = f_{hc} - f_0 \).

<table>
<thead>
<tr>
<th>Module</th>
<th>Predicted</th>
<th>( f_{hc} )</th>
<th>( f_0 )</th>
<th>( \Delta f )</th>
<th>( \Delta f/f_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>giant</td>
<td>7242-8184</td>
<td>7360-7859</td>
<td>16.0</td>
<td>14.5</td>
<td>1.5</td>
</tr>
<tr>
<td>hairy</td>
<td>829-1760</td>
<td>1340-1839</td>
<td>15.7</td>
<td>14.3</td>
<td>1.4</td>
</tr>
<tr>
<td>kruppel</td>
<td>5668-6389</td>
<td>5640-6139</td>
<td>18.2</td>
<td>16.1</td>
<td>2.1</td>
</tr>
<tr>
<td>zen</td>
<td>2615-3016</td>
<td>2540-3039</td>
<td>13.0</td>
<td>10.8</td>
<td>2.2</td>
</tr>
</tbody>
</table>
Table 3. Comparison of the discrimination of modules by SSPECIES and MSPECIES

<table>
<thead>
<tr>
<th>Known module</th>
<th>MSPECIES</th>
<th>SSPECIES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prediction</td>
<td>$r_m$</td>
</tr>
<tr>
<td>eve MHE</td>
<td>580–1079</td>
<td>13.7</td>
</tr>
<tr>
<td>run</td>
<td>2680–3179</td>
<td>12.0</td>
</tr>
<tr>
<td>ill: 1918–1397</td>
<td>840–1339</td>
<td>21.0</td>
</tr>
<tr>
<td>ill: 1248–2867</td>
<td>2140–2639</td>
<td>11.1</td>
</tr>
<tr>
<td>hairy: 1286–2127</td>
<td>1800–2299</td>
<td>8.6</td>
</tr>
<tr>
<td>hairy: 3058–3604</td>
<td>3060–3559</td>
<td>24.9</td>
</tr>
<tr>
<td>hairy: 6288–6589</td>
<td>6140–6639</td>
<td>11.4</td>
</tr>
<tr>
<td>hairy: 6396–7551</td>
<td>7460–7959</td>
<td>14.3</td>
</tr>
</tbody>
</table>

SSPECIES. The same experiment, when conducted with the planted motifs being different from those used by Stubb (meaning that the synthetic sequences were non-modules), showed that the average $r_m - r_s$ was $\sim 0.2$.

We did a similar comparison on upstream sequences of the gap genes hairy, run, ill and the MHE promoter (Halfon et al., 2002) of eve, taken from two species—D melanogaster and D. virilis. The motif set used is the same as the gap gene PWMs used above, except for the eve.MHE promoter, where we used the set of motifs from Halfon et al. (2002). $\mu$ was set to 0.5. The baseline scores were computed from the upstream sequence of dme f2, which does not have gap gene input. Table 3 compares the discrimination scores for each gene between SSPECIES and MSPECIES. All high ranking windows where either version had a discrimination score above 10.0 are tabulated. MSPECIES is found to discriminate better on most of the windows. For example, in the hairy module predicted at position 7460, the unaligned sequence from virilis has two strong occurrences of the Kr motif, which boosts the score. The ill (1918–1397) module and the hairy (6288–6589) module are also discriminated better by MSPECIES.

We are currently experimenting with sequences from D melanogaster and their putative orthologs from D. pseudoobscura. Upon alignment by Lagan, and extraction of ungapped blocks of length 10 or more and percent-identity 70 or more, about 40 – 50% of the sequences is found to be covered by such blocks. The synteny is good, as evidenced by the fact that two adjacent blocks are rarely separated by more than 1000 bp in either species. The neutral point mutation rate ($\mu$), as estimated from non-synonymous substitution rates in the coding regions, is $\sim 0.8$. However, finding regulatory modules based on the density of aligned blocks alone is not very effective, implying that running Stubb (MSPECIES) on these sequences should be an interesting exercise.

CONCLUSIONS AND FUTURE WORK

An HMM based method for module detection is developed here, capable of exploiting motif correlations and multiple species data. It is not yet possible to properly test Stubb on two Drosophila species since most of the comparison sequence available (with the relevant PWMs known) is limited to regions dense in modules. The complete sequence of D. pseudoobscura is due by the first half of 2003, and then we can properly test how two genomes fit in a parallel fashion improves the discrimination of modules from background sequence. Also, a genome-wide run of hcHMM should reveal interesting correlations and modules.

ACKNOWLEDGEMENTS

This material is based upon support provided by the Keck Foundation. The authors wish to thank Eldon Emberly and Nicholas Rajewsky for providing sequence data for testing Stubb on. The referees’ comments and useful suggestions are also acknowledged. A special note of thanks goes to Michael Brudno for letting us use the Lagan software.

REFERENCES


APPENDIX

Training parameters in a simple HMM (HMM0)

Given a sequence \( S \) and a set of Position Weight Matrices (PWMs) \( W \), the objective function to be maximized is

\[
F(S, \theta) = \log \left( \frac{Pr(S|\theta)}{Pr(S|\theta_0)} \right),
\]

where \( Pr(S|\theta) \) is the probability of HMM0 generating the sequence \( S \) using the parameters \( \theta \), and \( \theta_0 \) represents the parameter values that only allow the background motif to be used by the HMM0. \( \theta \) includes the transition probabilities \( p_i \) for each \( w \in W \). We henceforth denote \( W \cup \{w_b\} \) as \( W' \). Since \( Pr(S|\theta_0) \) depends only on \( W' \), which is constant, we shall outline how to maximize \( \log Pr(S|\theta) \), following the description in Durbin et al. (1999). A parse of the sequence \( S \) in terms of \( W' \) is denoted by \( T \), as described in Section ALGORITHM. We thus have

\[
\log Pr(S|\theta) = \log \sum_T Pr(S, T|\theta)
\]

The maximization is iterative, with the \( t \)th iteration computing a model \( \theta^{t+1} \) that improves the objective function from the current model \( \theta^t \). Let us define a function \( Q(\theta|\theta^t) \) as

\[
Q(\theta|\theta^t) = \sum_T Pr(T|S, \theta^t) \log Pr(S, T|\theta)
\]

It is easily shown that \( \log Pr(S|\theta) - \log Pr(S|\theta^t) \geq Q(\theta|\theta^t) - Q(\theta^t|\theta^t) \). Thus, if we maximize \( Q(\theta|\theta^t) \) over all \( \theta \), we shall always improve upon \( \log Pr(S|\theta^t) \), or remain there if the local maximum has been reached.

Let \( A_i(T, S) \) be the number of times motif \( w_i \in W' \) occurs in the parse \( T \) of \( S \). Also let \( E \) denote the probability of generating the sequence \( S \) given the parse \( T \). (This is simply the product of the appropriate subsequence probabilities \( Pr(s|w) \), and is independent of \( \theta \).) Then we have

\[
Pr(S, T|\theta) = E \prod_i p_i A_i(T, S)
\]

which gives us

\[
\begin{align*}
Q(\theta|\theta^t) &= \sum_T Pr(T|S, \theta^t) \\
&= \left( \log E + \sum_i A_i(T, S) \log p_i \right) \\
&= (\log E) \sum_T Pr(T|S, \theta^t) + \sum_i \log p_i \sum_i A_i(T, S) Pr(T|S, \theta^t)
\end{align*}
\]

Dropping the first term in Equation (4) since it does not depend on \( \theta \), we now need to maximize

\[
\sum \log p_i \sum_T A_i(T, S) Pr(T|S, \theta^t)
\]

Note that \( A_i(S) = \sum_T A_i(T, S) Pr(T|S, \theta^t) \) is simply the average number of occurrences of \( w_i \) in \( S \) over all parses \( T \). Thus the term to maximize is \( \sum_j A_j(S) \log p_i \), and this is maximized when

\[
p_i = \frac{A_i(S)}{\sum_j A_j(S)} \forall i.
\]

These update criteria are used iteratively to improve \( F(S|\theta) \) till the local maximum is reached, as indicated by a very small change in its value. \( A_i(S) \) can be computed in \( O(L) \) time by using the Backward-Forward algorithm for HMMs, where \( L \) is the length of \( S \).
\[ Pr(S, T|\theta) = \prod_{i|C_i \neq \emptyset} \prod_{j \in C_i} P_{ij} \prod_{j \in C_i'} \left( \frac{p_j (1 - \sum_{k \in C_i} P_{ik})}{\sum_{k \in C_i'} P_{ik}} \right)^{A_{ij}(T, S)} \times \prod_{i|C_i = \emptyset} \prod_{j} P_{ij}^{A_{ij}(T, S)} \times E. \]

Then, using the notation \( \overline{A}_{ij} \) to represent
\[ \sum_{T} A_{ij}(T, S) Pr(T|S, \theta) \]
we can rewrite Equation (4) as
\[ Q(\theta|\theta') = \sum_{i|C_i \neq \emptyset} \left( \sum_{j \in C_i} \overline{A}_{ij} \log p_{ij} + \sum_{j \in C_i'} \overline{A}_{ij} \log \left( 1 - \sum_{k \in C_i} p_{ik} \right) \right) + \sum_{i|C_i \neq \emptyset} \left( \sum_{j \in C_i'} \overline{A}_{ij} \log p_j - \sum_{j \in C_i} \overline{A}_{ij} \log p_k \right) + \sum_{i|C_i = \emptyset} \sum_{j} \overline{A}_{ij} \log p_{ij} + \text{constant} = A + B + \text{constant} \]
where
\[ A = \sum_{i|C_i \neq \emptyset} \left( \sum_{j \in C_i} \overline{A}_{ij} \log p_{ij} + \sum_{j \in C_i'} \overline{A}_{ij} \log \left( 1 - \sum_{k \in C_i} p_{ik} \right) \right) \]
and (after simplification, and using \( \sum p_j = 1 \))
\[ B = \sum_{i} \sum_{j \in C_i'} \overline{A}_{ij} \log \frac{p_j}{\sum_{k \in C_i'} p_k}. \]

Note that term A only has the parameters \( p_{ij} \), while B only has the parameters \( p_i \). To maximize \( Q(\theta|\theta') \), we need to solve for:
\[ \frac{\partial A}{\partial p_{ij}} = 0 \quad \forall i, j \quad (6) \]
\[ \frac{\partial B}{\partial p_i} = 0 \quad \forall i \quad (7) \]
If \( C_i' = \emptyset \) (\( p_{ij} \) is a parameter, for all \( j \)), or \( \sum_{k \in C_i'} \overline{A}_{ik} = 0 \), (6) gives the update criteria
\[ p_{ij} = \frac{\overline{A}_{ij}}{\sum_k \overline{A}_{ik}} \]
which is the straight-forward generalization of Equation 5. If \( C_i' \neq \emptyset \), (6) translates to the following system of equations in \( p_{ij} \)
\[ \frac{\overline{A}_{ij}}{p_{ij}} = \sum_{k \in C_i'} \frac{\overline{A}_{ik}}{p_{ik}}. \quad (8) \]
If \( \overline{A}_{ij} = 0 \), we set \( p_{ij} = 0 \), and drop the corresponding equation from the above system. Solving this system of equations in \( p_{ij} \) gives us the update criteria for \( p_{ij} \).

To solve (7), we first transform to the log variables \( u_i = \log p_i, \forall i \), and denote the vector of variables \( u_i \) by \( u \). We thus need to solve for \( \nabla^2 B(u) = 0 \), where \( \nabla B(u) \) is the gradient of \( B(u) \). We use the Newton iterative method to solve this system of equations. Each step of Newton’s method uses the update relation:
\[ \Delta u = - \left( \nabla^2 B(u) \right)^{-1} \nabla B(u) \quad (9) \]
where \( \Delta u \) is the change in \( u \) in the current iteration and \( \nabla^2 B(u) \) is the Hessian of \( B(u) \), denoted by \( H \). The expressions for \( \nabla B(u) \) and for \( H \) are straightforward, and are omitted here. \( \nabla \) is a (real) symmetric matrix. However, it is singular, and hence not invertible. This is because \( B(u) \) is scale invariant, \( \nabla B(u) = B(u + \epsilon) \forall \epsilon \), which implies \( \nabla B(u)^T 1 = 0 \), which in turn makes \( H \) singular.
To solve Equation (9), we compute \( H^{-1} \) as follows. Let \( H = VDV^T \), where the \( i^{th} \) column of \( V \), denoted by \( v_i \), is an eigenvector of \( H \), and \( D \) is a diagonal matrix containing the eigenvalues of \( H \), denoted by \( \lambda_i \). We can thus write \( H = \sum_i \lambda_i v_i v_i^T = \sum_{i|\lambda_i \neq 0} \lambda_i v_i v_i^T \), and using the fact that \( V^{-1} = V^T \), we get
\[ H^{-1} = \sum_{i|\lambda_i \neq 0} \frac{v_i v_i^T}{\lambda_i} \quad (10) \]
which is the expression used for \( \left( \nabla^2 B(u) \right)^{-1} \) in Equation 9. It can be shown that all the eigenvalues are negative and the function \( B(u) \) is convex, hence Newton’s method must converge. Once Newton’s method converges, the log variables \( u_i \) are transformed back to \( p_i = e^{u_i} \), and scaled appropriately to ensure \( \sum_i p_i = 1 \). This is the solution of (7) used in every update of \( \theta \).

**Expectation and variance of \( A_{ij}(X) \)**
Recall that \( X \) is a random sequence of length \( L \) generated by an HMM0 with parameters \( \theta \). By definition, \( A_{ij}(X) = \sum_T A_{ij}(T, X) Pr(T|X, \theta) \). Its expectation is then given by
\[ E_{ij} = \sum_X A_{ij}(X) Pr(X|\theta) = \sum_X \sum_T A_{ij}(T, X) Pr(T|X, \theta) Pr(X|\theta) = \sum_X \sum_T A_{ij}(T, X) Pr(T, X|\theta). \]
This can be shown to be (for the case when \( w_i \neq w_b \) and \( w_j \neq w_b \))

\[
E_{ij} = \frac{p_i p_j}{(1 - p_b)} \sum_{k=1}^{L-l_i} \alpha(k-1)(1 - p_b)^{L-l_i-k+1}
\]

where \( l_i \) is the length of \( w_i \) and \( \alpha(k) \) is the probability that all motifs placed before position \( k \) end before this position, in a random sequence generated by the HMM \( \theta \). \( \alpha(x) = 0 \), \( \forall x < 0 \), \( \alpha(0) = 1 \), and for all \( k > 0 \), we have \( \alpha(k) = \sum_{i | w_i \in W} \alpha(k-l_i) p_i \). The entire calculation takes \( O(L) \) time for each \( i, j \).

We compute the variance \( \sigma^2(A_{ij}(X)) \) by approximating it as the variance of \( A_{ij}(T, X) \) over \( T, X \). We performed experiments to confirm that this approximation is sufficiently accurate for our purposes. The expectation of \( A_{ij}(T, X) \) is \( E_{ij} \) computed above. Hence we only need to compute the second moment of this random variable, henceforth abbreviated as \( A_{ij} \). By definition, \( A_{ij} = \sum_k A_{ijk} \), where \( A_{ijk} \) is an indicator variable equal to 1 if \( w_j \) occurs at position \( k \) following a \( w_i \) (possibly with background in between), and 0 otherwise. Then,

\[
E(A_{ij}^2) = E \left( \left( \sum_k A_{ijk} \right)^2 \right) \]

\[
= E \left( \sum_k A_{ijk}^2 \right) + 2 \sum_{k_1, k_2 > k_1} E(A_{ijk_1} A_{ijk_2})
\]

\[
= E \left( \sum_k A_{ijk} \right) + 2 \sum_{k_1, k_2 > k_1} P(A_{ijk_1} = 1 \land A_{ijk_2} = 1).
\]  \( (11) \)

The first term in (11) is \( E_{ij} \) and the second term can be approximately calculated in \( O(L) \) time, giving an overall \( O(L) \) complexity for the variance computation. The details of this computation are not described here.