BIOINFORMATICS

Vol. 19 Suppl. 1 2003, pages i292–i301
DOI: 10.1093/bioinformatics/btg1040

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 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

*To whom correspondence should be addressed.

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 $Pr(S, T|\theta)$ with each parse T of the sequence S. The probability that S was generated by an HMM with parameters θ is then given by

$$Pr(S|\theta) = \sum_T Pr(S, T|\theta)$$

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

$$F(S, \theta) = \log \frac{Pr(S|\theta)}{Pr(S|\theta_b)}$$

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 $F(S, \theta)$.

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 $s = s_1 s_2 \ldots s_l$, and let w_{ij} be the probability of sampling base i at the j^{th} position of the PWM w. Then, $Pr(s|w) = \prod_{i=1}^{l} w_{s_i i}$. 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 θ 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 θ. (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 θ, 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 spacing. 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 θ 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}. In 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 θ. 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 θ 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 θ. 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 σ_{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

$$Z_{ij} = \frac{A_{ij}(S) - E_{ij}}{\sigma_{ij}} \tag{1}$$

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 σ_{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 $Pr(i \to j)$, which is the probability of choosing a w_j following a w_i with nothing except background in between. The parameter set θ includes all possible p_i and some or all p_{ij}. Let $\texttt{Corr}(i, j) = \texttt{true}$ if and only if correlation was detected for $\{w_i, w_j\}$. If $\texttt{Corr}(i, j) = \texttt{true}$, then $Pr(i \to j) = p_{ij}$. For all i, j such that $\texttt{Corr}(i, j) = \texttt{false}$,

$$Pr(i \to j) = p_j \left(\frac{1 - \sum_{k|\texttt{Corr}(i,k)=\texttt{true}} p_{ik}}{\sum_{k|\texttt{Corr}(i,k)=\texttt{false}} p_k} \right)$$

Here, the parameter p_j is normalized appropriately to ensure that $\sum_j Pr(i \to j) = 1$. 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 $Pr(i \to j)$. The spacing between motifs is thus controlled by the exponential decay of powers of $Pr(i \to b)$. If no non-background motif has been planted so far, $w_j \in W \cup \{w_b\}$ is planted with probability $Pr(b \to j)$. Note that if $\texttt{Corr}(i, j) = \texttt{false}$ for all j, then $Pr(i \to j) = p_j$, 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 θ that, following the EM theory, is guaranteed to improve the objective function $F(S, \theta)$ in each iteration, until convergence. The calculations are outlined in the Appendix. In the extreme cases when $\texttt{Corr}(i, j)$ 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 $W \cup \{w_b\}$, real numbers
$\tau_z$,$\tau_e$; Output: Score of $S$.
1. Set Corr(i,j) = false for all pairs $i, j$. Set $\theta$
to include all $p_i$, but no $p_{ij}$.
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 $Z_{ij} > \tau_z$ and $E_{ij} > \tau_e$, set Corr(i,j) = true.
6. End For
7. Set $\theta$ to include all $p_i$ and all $p_{ij}$ for which
Corr(i,j) = true.
8. If Corr(i,j) = 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_b are not detected, hence p_{ib} or p_{bi} is never trained, for any i. Each iteration in the training of the hcHMM (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 δ, and each window S is score by the above algorithm.

Multiple species and phylogenetic information

In this section, we extend *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 *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 syntenic 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 *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 retain 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 σ 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 β at that position.

Under these assumptions, we have

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

where l is the length of w, $\Sigma = \{A, C, G, T\}$, $w_{\beta i}$ is the probability of emitting β 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 λ and the evolution time t_s between the ancestor and the species s. For each position i, one 'creates' a base β 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 μ_s and a new base selected with a frequency defined by w. If μ_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 $\mu_s \sim 1$, the sum over the ancestor base ($\sum_{\beta\in\Sigma} w_{\beta i}$) is replaced by 1, and the sequences in σ 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 $Pr(\sigma|w) = \prod_{i=1}^{l} \left(w_{s_{i1}i}(w_{s_{i2}i}\mu + (1-\mu)\delta_{s_{i1}s_{i2}}) \right)$, where s_{i1}, s_{i2} are the bases in the two species at position i, and μ is composed from the total time of evolution between them.

Implementation

The *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, *Stubb* suppresses all overlapping windows that score less than it.

To compute the strand bias b_w of a motif w, *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), *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. *Stubb* takes two sequences aligned in this manner, and extracts all ungapped, 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 hcHMM on synthetic sequences in which two motifs were planted with varying degrees of correlation. An 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, *Stubb* (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 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 hcHMM had little effect on the score in this case.

We next constructed a toy example from yeast regulatory sequences, again to test hcHMM. The transcription factors MCM1 and MATα2 are known to act cooperatively in the mating pathway (Mead *et al.*, 2002). Six regulatory regions where MATα2 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 hcHMM boosted this significantly to 54.5. After training of parameters, the transition probability matrix showed $Pr(1\rightarrow1) = 0$, $Pr(1\rightarrow2) = 0.38$, $Pr(2\rightarrow1) = 0.005$, $Pr(2\rightarrow2) = 0.0003$, $Pr(1\rightarrow b) = 0.62$, $Pr(2\rightarrow b) = 0.995$, where 1 represents MCM1, 2 represents MATα2, and b represents the background motif. The probabilities strongly suggest a motif structure MATα2→ MCM1→ MATα2 (with the first interval

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

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

($2 \rightarrow 1$) 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*, *tll*, *cd*, *dl* and torRE (Rajewsky *et al.*, 2002). The 10 kb upstream regions of the genes *eve*, *gt*, *kni*, *Kr*, *run*, *tll*, and *hb*, as well as the 12 kb upstream region of *hairy* are used as input sequences, in separate runs of the program, with 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 hcHMM) near the module, its scores f_{hc} and f_0 by hcHMM and 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 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 HMM0 to hcHMM, and that the module for *Kr* predicted by hcHMM has a much better overlap with the known module than the HMM0 prediction. (Data not shown.)

Table 2. Advantage of hcHMM over HMM0 in detecting modules. f_{hc}: hcHMM score, f_0: HMM0 score, $\Delta_f = f_{hc} - f_0$

Module	Predicted	f_{hc}	f_0	Δ_f	Δ_f/f_0
giant: 7242–8184	7360–7859	16.0	14.5	1.5	0.10
hairy: 829–1760	1340–1839	15.7	14.4	1.3	0.09
kruppel: 5668–6389	5640–6139	18.2	16.1	2.1	0.13
zen: 2615–3016	2540–3039	13.0	10.8	2.2	0.20

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 3. Comparison of the discrimination of modules by SSPECIES and MSPECIES

Known module	MSPECIES		SSPECIES
	Prediction	r_m	r_s
eve MHE: 592–882	580–1079	**13.7**	12.7
run	2680–3179	**12.0**	11.6
tll: 918–1397	840–1339	**21.0**	16.5
tll: 2485–2867	2140–2639	**11.1**	10.8
hairy: 1286–2217	1800–2299	8.6	**10.9**
hairy: 3058–3604	3060–3559	24.9	**25.9**
hairy: 6288–6589	6140–6639	**11.4**	9.0
hairy: 6396–7551	7460–7959	**14.3**	12.6

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 ~ 0.2.

We did a similar comparison on upstream sequences of the gap genes *hairy*, *run*, *tll* 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). μ was set to 0.5. The baseline scores were computed from the upstream sequence of *dmef2*, 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 *tll* (918–1397) 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 (μ), 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 Nicholaus 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

Baum,L. (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41**, 164–171.

Bergman,C. and Kreitman,M. (2001) Analysis of conserved noncoding DNA in Drosophila reveals similar constraints in intergenic and intronic sequences. *Genome Res.*, **111**, 335–345.

Berman,B., Nibu,Y., Pfeiffer,B., Tomancak,P., Celniker,S., Levine,M., Rubin,G. and Eisen,M. (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.

Brudno,M., Do,C., Cooper,G., Kim,M., Davydov,E., Green,E., Sidow,A. and Batzoglou,S. (2003) LAGAN and Multi-LAGAN: Efficient Tools for Large-Scale Multiple Alignment of Genomic DNA. *Genome Res.*, In press.

Carroll,S., Grenier,J. and Weatherbee,S. (2001) *From DNA to Diversity*. Blackwell Science, London.

Cliften,P., Hillier,L., Fulton,L., Graves,T., Miner,T., Gish,W., Waterston,R. and M.J., (2001) Surveying Saccharomyces genomes to identify functional elements by comparative DNA sequence analysis. *Genome Res.*, **11**, 1175–1186.

Davidson,E. (2001) *Genomic Regulatory Systems*. Academic Press, San Diego.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1999) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

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

Grundy,W.N., Bailey,T.L., Elkan,C.P. and Baker,M.E. (1997) Meta-meme: Motif-based hidden markov models of protein families. *Computer Applications in the Biosciences*, **13**, 397–406.

GuhaThakurta,D. and Stormo,G.D. (2001) Identifying target sites for cooperatively binding factors. In *RECOMB01: Proceedings of the Fifth Annual International Conference on Computational Molecular Biology*. Montreal, Canada.

Halfon,M., Grad,Y., Church,G. and Michelson,A. (2002) Computation-based discovery of related transcriptional regulatory modules and motifs using an experimentally validated combinatorial model. *Genome Res.*, **12**, 1019–1028.

Loots,G., Ovcharenko,I., Pachter,L., Dubchak,I. and Rubin,E. (2002) rVISTA for comparative sequence-based discovery of functional transcription factor binding sites. *Genome Res.*, **12**, 832–839.

Mannervik,M., Nibu,Y., Zhang,H. and Levine,M. (1999) Transcriptional coregulators in development. *Science*, **284**, 606–609.

Mead,J., Bruning,A. and Gill,M. (2002) Interactions of the Mcm1 MADS box protein with cofactors that regulate mating in yeast. *Mol. Cell. Biol.*, **22**, 4607–4621.

Morgenstern,B., Frech,K., Dress,A. and Werner,T. (1998) Dialign: Finding local similarities by multiple sequence alignment. *Bioinformatics*, **14**, 290–294.

Rajewsky,N., Vergassola,M., Gaul,U. and Siggia,E. (2002) Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo. *BMC Bioinformatics*, **3**.

Sinha,S. (2002) Discriminative motifs. In *RECOMB02: Proceedings of the Sixth Annual International Conference on Computational Molecular Biology*. Washington, D.C..

Zhu,J. and Zhang,M.Q. (1999) SCPD: a promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics*, **15**, 563–577. http://cgsigma.cshl.org/jian/.

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 $F(S, \theta) = \log(Pr(S|\theta)/Pr(S|\theta_b))$, where $Pr(S|\theta)$ is the probability of HMM0 generating the sequence S using the parameters θ, and θ_b represents the parameter values that only allow the background motif to be used by the HMM0. θ includes the transition probabilities p_i for each $w \in W \cup \{w_b\}$. We henceforth denote $W \cup \{w_b\}$ as W'. Since $Pr(S|\theta_b)$ 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 θ^{t+1} that improves the objective function from the current model θ^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 θ, 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 θ.) Then we have

$$Pr(S, T|\theta) = E \times \prod_i p_i^{A_i(T,S)} \tag{3}$$

which gives us

$$\begin{aligned} Q(\theta|\theta^t) &= \sum_T Pr(T|S, \theta^t) \\ &\quad \times \left(\log E + \sum_i A_i(T, S) \log p_i \right) \\ &= (\log E) \sum_T Pr(T|S, \theta^t) \\ &\quad + \sum_i \log p_i \sum_T A_i(T, S) Pr(T|S, \theta^t) \end{aligned} \tag{4}$$

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

$$\sum_i \log p_i \sum_T A_i(T, S) Pr(T|S, \theta^t)$$

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 $\sum_i A_i(S) \log p_i$, and this is maximized when

$$p_i = \frac{A_i(S)}{\sum_j A_j(S)} \quad \forall i. \tag{5}$$

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.

Training parameters in a history conscious HMM (hcHMM)

The model parameters θ now include all p_i's and some (or none, or all) p_{ij}'s. Let $C_i = \{j | p_{ij}$ is a parameter in $\theta\}$. The motifs defined by this index set are those that are correlated with motif w_i. Let C_i' denote the complement of set C_i. Let $A_{ij}(T, S)$ be the number of times w_j follows w_i (with nothing except w_b in between) in parse T of S. Then, following the transition probability definitions given in Section ALGORITHM, Equation (3) now becomes

$$Pr(S,T|\theta) =$$

$$\prod_{i|C_i\neq\phi}\prod_{j\in C_i} p_{ij}^{A_{ij}(T,S)} \prod_{j\in C_i'} \left(\frac{p_j(1-\sum_{k\in C_i} p_{ik})}{\sum_{k\in C_i'} p_k}\right)^{A_{ij}(T,S)}$$

$$\times \prod_{i|C_i=\phi}\prod_j p_j^{A_{ij}(T,S)} \times E.$$

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

$$Q(\theta|\theta^t) =$$

$$\sum_{i|C_i\neq\phi}\left(\sum_{j\in C_i}\overline{A}_{ij}\log p_{ij} + \sum_{j\in C_i'}\overline{A}_{ij}\log\left(1-\sum_{k\in C_i} p_{ik}\right)\right)$$

$$+\sum_{i|C_i\neq\phi}\left(\sum_{j\in C_i'}\overline{A}_{ij}\log p_j - \sum_{j\in C_i'}\overline{A}_{ij}\left(\log\sum_{k\in C_i'} p_k\right)\right)$$

$$+\sum_{i|C_i=\phi}\sum_j\overline{A}_{ij}\log p_j + \texttt{constant}$$

$$= A + B + \texttt{constant}$$

where

$$A =$$

$$\sum_{i|C_i\neq\phi}\left(\sum_{j\in C_i}\overline{A}_{ij}\log p_{ij} + \sum_{j\in C_i'}\overline{A}_{ij}\log\left(1-\sum_{k\in C_i} p_{ik}\right)\right)$$

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 $Q(\theta|\theta^t)$, we need to solve for:

$$\frac{\partial A}{\partial p_{ij}} = 0 \quad \forall i, j|j\in C_i \tag{6}$$

$$\frac{\partial B}{\partial p_i} = 0 \quad \forall i \tag{7}$$

If $C_i' = \phi$ (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 $C_i' \neq \phi$, (6) translates to the following system of equations in p_{ij}

$$\frac{\overline{A}_{ij}}{p_{ij}} = \frac{\sum_{k\in C_i'}\overline{A}_{ik}}{1-\sum_{k\in C_i} p_{ik}}. \tag{8}$$

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 $\mathbf{u}$. We thus need to solve for $\nabla B(\mathbf{u}) = \mathbf{0}$, where $\nabla B(\mathbf{u})$ is the gradient vector of $B(\mathbf{u})$. We use the Newton iterative method to solve this system of equations. Each step of Newton's method uses the update relation:

$$\Delta\mathbf{u} = -(\nabla^2 B(\mathbf{u}))^{-1}\nabla B(\mathbf{u}) \tag{9}$$

where $\Delta\mathbf{u}$ is the change in $\mathbf{u}$ in the current iteration and $\nabla^2 B(\mathbf{u})$ is the Hessian matrix of $B(\mathbf{u})$, denoted by H. The expressions for $\nabla B(\mathbf{u})$ and for H are straight-forward, and are omitted here. H is a (real) symmetric matrix. However, it is singular, and hence not invertible. This is because $B(\mathbf{u})$ is scale invariant ($B(\mathbf{u}) = B(\mathbf{u}+\epsilon\mathbf{1})\ \ \forall\epsilon$), which implies $(\nabla B(\mathbf{u}))^T\mathbf{1} = 0$, 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 $\mathbf{v_i}$, is an eigenvector of H, and D is a diagonal matrix containing the eigenvalues of H, denoted by λ_i. We can thus write $H = \sum_i \lambda_i\mathbf{v_i}\mathbf{v_i}^T = \sum_{i|\lambda_i\neq 0}\lambda_i\mathbf{v_i}\mathbf{v_i}^T$, and using the fact that $V^{-1} = V^T$, we get

$$H^{-1} = \sum_{i|\lambda_i\neq 0}\frac{\mathbf{v_i}\mathbf{v_i}^T}{\lambda_i} \tag{10}$$

which is the expression used for $(\nabla^2 B(\mathbf{u}))^{-1}$ in Equation (9). It can be shown that all the eigenvalues are negative and the function $B(\mathbf{u})$ 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 θ.

Expectation and variance of $A_{ij}(X)$

Recall that X is a random sequence of length L generated by an HMM0 with parameters θ. By definition, $A_{ij}(X) = \sum_T A_{ij}(T,X)Pr(T|X,\theta)$. Its expectation is then given by

$$\begin{aligned}E_{ij} &= \sum_X A_{ij}(X)Pr(X|\theta)\\ &= \sum_X\sum_T A_{ij}(T,X)Pr(T|X,\theta)Pr(X|\theta)\\ &= \sum_X\sum_T A_{ij}(T,X)Pr(T,X|\theta).\end{aligned}$$

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 HMM0. $\alpha(x) = 0,\ \forall x < 0, \alpha(0) = 1$, 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,

$$\begin{aligned} E(A_{ij}^2) &= E\left(\left(\sum_k A_{ijk}\right)^2\right) \\ &= E\left(\sum_k A_{ijk}^2\right) + 2\sum_{k_1,k_2>k_1} E(A_{ijk_1}A_{ijk_2}) \\ &= E\left(\sum_k A_{ijk}\right) \\ &+2\sum_{k_1,k_2>k_1} Pr(A_{ijk_1} = 1 \wedge A_{ijk_2} = 1). \qquad (11) \end{aligned}$$

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.