Pacific Symposium on Biocomputing 12:472-483(2007)

DISCOVERING MOTIFS WITH TRANSCRIPTION FACTOR DOMAIN KNOWLEDGE*

HENRY C.M. LEUNG
FRANCIS Y.L. CHIN
BETHANY M.Y. CHAN

Department of Computer Science, University of Hong Kong, Pokfulam
Hong Kong, China

We introduce a new motif-discovery algorithm, DIMDom, which exploits two additional kinds of information not commonly exploited: (a) the characteristic pattern of binding site classes, where class is determined based on biological information about transcription factor domains and (b) posterior probabilities of these classes. We compared the performance of DIMDom with MEME on all the transcription factors of Drosophila with at least one known binding site in the TRANSFAC database and found that DOMDom outperformed MEME with 2.5 times the number of successes and 1.5 times in the accuracy in finding binding sties and motifs.

1. Introduction

One important problem in bioinformatics is understanding how genes cooperate to perform functions. Related to this is the subproblem of *discovering motifs*. The context behind the motif discovering problem is the following. *Gene expression* is the process whereby a gene is decoded to form an mRNA sequence which is then used to produce the corresponding protein sequence. In order to start the gene expression process, a molecule called a *transcription factor* will bind to a short substring, called a *binding site*, in the promoter region of the gene. A transcription factor can bind to several binding sites in the promoter regions of different genes to make these genes co-express, and such binding sites should have common patterns. The *motif discovering problem* is to discover the common patterns, or motifs, from a set of promoter regions without knowing the positions of the binding sites. However, many motifs in real biological data cannot be discovered by existing algorithms because the existing models [3, 8, 12, 13, 20] that represent motifs might not be able to capture the different pattern variations of the binding sites.

PSSM (Position Specific Scoring Matrix) [2, 4, 6, 7, 10, 11, 14] is the most common motif representation. It uses a $4 \times l$ matrix of real numbers to represent a length-l motif. The j-th column of 4 numbers gives us the probability,

* The research was supported in parts by the RGC grant HKU 7120/06E.

respectively, that symbol 'A', 'C', 'G' or 'T' occupies at the *j*-th position of the motif. The goal is to discover the optimal motif matrix which maximizes the likelihood of the input sequences being generated according to the matrix.

Existing algorithms assume the prior probability of each matrix being chosen to generate the input sequences is the same. However, this assumption is not correct in real biological data. Transcription factors mainly bind to the binding sites by substructures called *active binding domains* (in short, *domain*), e.g. zinc finger [23], leucine zipper [16] and homeodomain [19]. Although the binding sites of transcription factors with the same domain do not necessarily have the same patterns, they should share some common characteristics [18]. For example, binding sites of zinc finger usually contain the nucleotide 'G' regularly and binding sites of homeodomain usually contain the "TAAT" substring. If we know which domains of the transcription factors contact the binding sites, we can improve the accuracy of existing motif discovering algorithms by adding constraints on the motifs [5, 15, 21]. For some motif classes, it might be possible to find the motif by considering only substrings in the DNA sequences with certain characteristics as candidates for binding sites. However, we usually do not know which transcription factors or, more specifically, which domains of the transcription factors contact the binding sites. The approach of searching for substrings with characteristics of each possible motif class is not only time-consuming, but may even fail to find the hidden motif because of the following two weaknesses of this approach. Firstly, the number of wrongly predicted binding sites might be large, e.g. many substrings in the input sequences with pattern [CG] . . [CG] . . [CG] are not binding sites of a motif in Class I (to be introduced in Section 2). Secondly, some binding sites of a motif in a particular class may not have the corresponding characteristics exactly, e.g. a binding site of motif in Class IV may contain the pattern TGA.*TGA instead of TGA.*TCA. A natural question is: can we improve the performance of motif discovering problem by knowing only the characteristics of each possible motif class?

Narlikar et al. [17] trained 3847 binding sites in the TRANSFAC database and defined three motif classifiers using 1387 features. Each motif classifier can represent the common features for binding sites in the corresponding motif class precisely. However, the definition of the motif classifiers highly depends on a large set of training binding sites and may not capture the real common features of binding sites in the motif class. Xing and Karp [24] used a similar method by training 271 motif matrices in the TRANSFAC database which represents about 2000 binding sites

In this paper, we model the common features of different motif classes by much less parameters than the above methods (Section 2). Our algorithm DIMDom (Section 3), which stands for DIscovering Motifs with DOMain

knowledge, discovers motifs by an EM approach: the *expectation* step finds over-represented patterns in the DNA sequence, while the *maximization* step, based on the motif matrix with the maximum log likelihood, guesses the class of the binding site patterns according to posterior probabilities and then modifies the motif matrix according to the class guessed. Besides getting more accurate motifs, the binding sites with domain knowledge can converge to the real solution (motif) more quickly as shown in the experiments (Section 4) on real biological data when compared with the popular algorithm MEME.

2. Our Model

The input sequences can be broken up into length-l (overlapping) substrings $X = \{X_1, X_2, \ldots, X_w\}$ and each substring in X either belongs to a background (non-motif) substring with a prior probability λ_b or belongs to an instance of the hidden motif M with a prior probability $1 - \lambda_b$. In particular, $Z = (Z_1, Z_2, \ldots, Z_w)$ is the missing data that determines whether X_i is generated according to the background probability B ($Z_i = 1$) or the hidden matrix M ($Z_i = 0$). The likelihood of some particular B, M, λ_b being the hidden parameters of the finite mixture model [2] is defined as

$$\begin{aligned} L(B, M, \lambda_b \mid X, Z) &= P(X, Z \mid B, M, \lambda_b) \\ &= \prod_{i=1}^{w} \left(\left[\lambda_b \prod_{j=1}^{l} b(X_i[j]) \right]^{Z_i} \left[(1-\lambda_b) \prod_{j=1}^{l} M(X_i[j], j) \right]^{1-Z_i} \right) \end{aligned} \quad (1)$$

The goal of many existing algorithms [2, 4, 10] is to discover the B, M, λ_b with the maximum likelihood (or log likelihood).

Transcription factors are protein sequences with different three dimensional structures. They have different substructures, or domains, for recognizing and binding to specific binding sites. The binding affinity of a transcription factor depends on whether the binding sites have certain DNA patterns match with the domains of the transcription factor. For example, basic helix-loop-helix proteins usually bind to strings with the pattern "CA . . TG" [1]. Other examples can be found in [16, 19, 23, 25].

Narlikar and Hartemink [18] analyzed 3847 published binding sites. They found that these binding sites can be classified into six groups with different occurrence counts. These counts represent the prior probabilities as shown in Table 1. For example, the probability $P_m(2)$ that the hidden matrix is in Class II (Cys_4) is approximately 734/3847. Based on this observation, we introduce the Bayesian Mixture Model to describe these uneven probabilities.

Table 1. The six classes of binding sites patterns.

Class name	Characteristics	Count
I. Cys2His2 (zinc-coordinating)	G . . G | G . . G . . G | [CG] . . [CG] . . [CG]	776
II. Cys_4 (zinc-coordinating)	AGGTCA | TGACCT	734
III. bHLH (basic domain)	CA . . TG	182
IV. bZip (basic domain)	TGA .* TCA	1353
V. Forkhead (helix-turn-helix)	no characteristics	281
VI. Homeodomain (helix-turn-helix)	TAAT | ATTA	621
	Total	3847

" . " means any nucleotide. " .* " means zero or more nucleotides. " [ ] " means one of the nucleotides in the bracket. " | " means or.

2.1. *Bayesian Mixture Model*

Each substring in X is assumed either generated according to a background probability $B = (b(\mathrm{A}), b(\mathrm{C}), b(\mathrm{G}), b(\mathrm{T}))$ or a hidden matrix M. However, the prior probability of each matrix being the hidden matrix is not the same. A motif class g, $g = 1, \ldots, 6$ is randomly chosen according to probability distribution $P_m = \{P_m(g)\}$ where $\sum_{g=1}^{6} P_m(g) = 1$. Once a motif class is chosen, a probability matrix is picked, with equal probability, from the chosen class as the hidden matrix. The goal of the motif discovering problem is to discover motif M and other parameters with maximum likelihood with respect to the given X and P_m.

Given the joint distribution of the substring X, the missing data Z, the hidden motif M and the motif class g, the likelihood of some particular B, λb, Pm being the hidden parameters of the Bayesian mixture model is defined as

$$\begin{aligned} L(B, \lambda_b, P_m \mid X, Z, M, g) &= P(X, Z, M, g \mid B, \lambda_b, P_m) \\ &= P(X, Z \mid M, g, B, \lambda_b, P_m) P(M, g \mid B, \lambda_b, P_m) \\ &= P(X, Z \mid B, M, \lambda_b) P(M, g \mid P_m) \\ &= L(B, M, \lambda_b \mid X, Z) P(M \mid g, P_m) P(g \mid P_m) \\ &= L(B, M, \lambda_b \mid X, Z) P(M \mid g) P_m(g) \end{aligned} \quad (2)$$

Therefore, the likelihood $L(B, \lambda_b, P_m \mid \mathrm{X}, \mathrm{Z}, M, g)$ is equal to $L(B, M, \lambda_b \mid \mathrm{X}, \mathrm{Z})$ times the term $P(M \mid g)P_m(g)$ which is the probability of class g being chosen and matrix M being picked from class g.

2.2. *Characteristics of the Motif Classes*

Each motif class can be characterized by a regular expression as shown in Table 1. A matrix for a particular motif class should contain a $4 \times l'$ sub-matrix M' where $l' \leq l$, which satisfies the restriction stated by the regular expression. Note that a probability matrix can belong to more than one motif class.

Each symbol 'A', 'C', 'G', 'T' in the regular expression means the entries $M'(\mathrm{A},j)$, $M'(\mathrm{C},j)$, $M'(\mathrm{G},j)$ or $M'(\mathrm{T},j)$ of the corresponding j-th column of the sub-matrix M' are larger than some predefined threshold β, $0.25 < \beta \leq 1$. For example, the regular expression "CA . . TG" in Class III means all matrices in

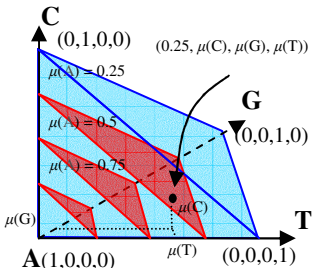

Figure 1. Graphical representation of all possible column vectors (μ(A), μ(C), μ(G), μ(T)) of a probability matrix.

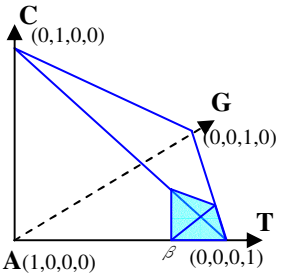

Figure 2. Graphical representation of all possible column vectors with $\mu(T) \geq \beta$.

Class III must contain a 4 × 6 sub-matrix *M'* such that *M'*(C, 1) ≥ β, *M'*(A, 2) ≥ β, *M'*(T, 5) ≥ β and *M'*(G, 6) ≥ β. Since the Class V has no characteristics, we assume all matrices belong to Class V, i.e. the regular expression is ".*".

Since the size of the sample space for each motif class is not the same, the likelihood of a particular class g given a matrix M, i.e. $P(M \mid g = k)$, $k = 1, \ldots, 6$, is not the same for different motif classes. In order to compare (without finding their exact values) the likelihood of different motif classes when given a matrix, we consider a 4 × 1 column vector $CV = (\mu(A), \mu(C), \mu(G), \mu(T))$ in a probability matrix. Since $0 \leq \mu(A), \mu(C), \mu(G), \mu(T) \leq 1$ and $\mu(A) + \mu(C) + \mu(G) + \mu(T) = 1$, the sample space of CV can be represented by the set of points in the tetrahedron shown in Figure 1 [10]. The four corners of the tetrahedron at (1,0,0,0), (0,1,0,0), (0,0,1,0) and (0,0,0,1) represent the four nucleotides A, C, G and T. Without loss of generality, let CV be the first column of a 4 × 4 matrix with the pattern "TAAT" in motif Class VI (Table 1), in which case $\mu(T) \geq \beta$.

To illustrate the idea, let us consider two classes of motif. In Class V a column vector *CV'* is randomly picked from all possible column vectors, whereas in Class VI, a column vector CV is randomly picked from all column vectors with $\mu(T) \geq \beta$. As the size of the sample space for column vectors with $\mu(T) \geq \beta$, i.e. the tetrahedron shown in Figure 2, is $(1 - \beta)^3$ of the size of the sample space for arbitrary column vectors, i.e. the whole tetrahedron, conditional probability $P(CV \mid g = 6)$ is $1/(1 - \beta)^3$ times higher than the conditional probability $P(CV' \mid g = 5)$.

Similarly, we may compare the conditional probability of a particular matrix *M'* being picked given that it is from Class V (all probability matrices) and the conditional probability of another matrix M being picked given that it is from one of the remaining classes. For example, assume $l = 4$ and $\beta = 0.8$. The conditional probability $P(M \mid g = 6)$ that a particular 4 × 4 matrix M in Class VI is picked from all length-4 matrices in Class VI is $1/(2(1 - 0.8)^{3\times4}) = 1.2 \times 10^8$ times larger than the conditional probability $P(M' \mid g = 5)$ that another matrix *M'*

is picked from all length-4 matrices in Class V. Note that, if *M'* does not belong to Class VI, $P(M' \mid g = 6) = 0$.

When the motif length l is not exactly 4, care should be taken not to double count those matrices with more than one sub-matrix satisfying the requirement (by using the Inclusion and Exclusion Principle).

3. DIMDom Algorithm

DIMDOM, which stands for DIscovering Motifs with DOMain knowledge, uses the expectation maximization (EM) approach to discover the motif matrix from the input sequences. In the expectation step (E-step), based on the current estimates of parameters M, B, λ_b and g, DIMDom algorithm calculates the expected log likelihood $\log L(B, \lambda_b, P_m \mid X, Z, M, g)$, over the conditional probability distribution of the missing data Z from the input sequences X. In the maximization step (M-step), DIMDom algorithm calculates a new set of parameters M, B, λ_b and g based on the new estimated Z for maximizing the log likelihood. These two steps will be iterated in order to obtain a probability matrix with larger log likelihood. In order to discover the probability matrix with maximum log likelihood (instead of local maxima), DIMDom algorithm repeats the EM steps with different *seed* matrices.

3.1. *Expectation step*

Given a fixed probability matrix $M^{(0)}$, the background probability $B^{(0)}$, prior probability $\lambda_b{}^{(0)}$ and the motif class $g^{(0)}$, the expected log likelihood is

$$
\begin{aligned}
&\mathop{\mathrm{E}}_{(Z|X, M^{(0)}, B^{(0)}, \lambda_b{}^{(0)}, g^{(0)})} \left(\log L(B, \lambda_b, P_m \mid X, Z, M, g)\right) \\
&= \sum_{i=1}^{w} \left\{ Z_i^{(0)} \left[\log(\lambda_b{}^{(0)}) + \sum_{j=1}^{l} \log\left(b^{(0)}(X_i[j])\right) \right] \right. \\
&= \qquad \left. + (1 - Z_i^{(0)}) \left[\log(1 - \lambda_b{}^{(0)}) + \sum_{j=1}^{l} \log\left(M^{(0)}(X_i[j], j)\right) \right] \right\} \qquad \text{from (1) and (2)} \\
&\quad + \log\left((p(M^{(0)} \mid g^{(0)}) P_m(g) \right) \qquad (3) \\
&= \sum_{i=1}^{w} \left\{ Z_i^{(0)} \sum_{j=1}^{l} \log\left(b(x_i[j])\right) \right\} \\
&\quad + \sum_{i=1}^{w} \left\{ (1 - Z_i^{(0)}) \sum_{j=1}^{l} \log\left(M(x_i[j], j)\right) \right\} + \log\left((p(M \mid g) P_m(g) \right) \\
&\quad + \sum_{i=1}^{w} \left\{ Z_i^{(0)} \log(\lambda_b) + (1 - Z_i^{(0)}) \log(1 - \lambda_b) \right\} \qquad (4)
\end{aligned}
$$

where $Z_i^{(0)} = \mathrm{E}(Z_i \mid \mathrm{X}, M^{(0)}, B^{(0)}, \lambda_b{}^{(0)})$ which can be calculated as follows

$$Z_i^{(0)} = \frac{\lambda_b{}^{(0)} \prod_{j=1}^{l} b^{(0)}(X_i[j])}{\lambda_b{}^{(0)} \prod_{j=1}^{l} b^{(0)}(X_i[j]) + (1-\lambda_b{}^{(0)}) \prod_{j=1}^{l} M^{(0)}(X_i[j], j)} \quad (5)$$

Therefore, we can calculate the expected log likelihood and the expected $Z^{(0)}$ from X, $M^{(0)}$, $B^{(0)}$,$\lambda_b{}^{(0)}$ and $g^{(0)}$ by Equations (4) and (5).

3.2.*Maximization step*

Based on Equation (4), we can calculate the parameters $M^{(1)}$, $B^{(1)}$,$\lambda_b{}^{(1)}$ and $g^{(1)}$ to maximize the expected log likelihood. $\lambda_b{}^{(1)}$ is involved in the last term in Equation (4) only and the expected log likelihood will be maximized when $\lambda_b{}^{(1)} = \sum_{i=1}^{w} (Z_i^{(0)} / w)$. $B^{(1)}$ is involved in the first term in Equation (4) which will be maximized when

$$b(\alpha)^{(1)} = \frac{\sum_{i=1}^{w} \sum_{j=1}^{l} Z_i^{(0)} \mathrm{I}(X_i[j] = \alpha)}{\sum_{\alpha'=\mathrm{A,C,G,T}} \sum_{i=1}^{w} \sum_{j=1}^{l} Z_i^{(0)} \mathrm{I}(X_i[j] = \alpha')}$$

where α can be A, C, G or T, and I(s) = 1 if and only if the proposition s is true and I(s) = 0 otherwise.

$M^{(1)}$ and $g^{(1)}$ are involved in the second term in Equation (4). In order to find the probability matrix $M^{(1)}$ and the motif class $g^{(1)}$, we assign $M^{(1)}$ and $g^{(1)}$ to be the probability matrix for each motif class that maximizes the expected log likelihood. Consider $g^{(1)}$ = 5, Equation (4) will be maximized (by considering a Lagrange Multiplier of each column vector of *M'*) when

$$M'(\alpha, j) = \frac{\sum_{i=1}^{w} (1 - Z_i^{(0)}) \mathrm{I}(X_i[j] = \alpha)}{\sum_{\alpha'=\mathrm{A,C,G,T}} \sum_{i=1}^{w} (1 - Z_i^{(0)}) \mathrm{I}(X_i[j] = \alpha')} \quad (6)$$

When $g^{(1)}$ = 1, 2, 3, 4 or 6, the matrix *M'* calculated in Equation (6) will maximize the log likelihood if *M'* belongs to the corresponding class. However, when *M'* does not belong to the corresponding class, we have to test all the *boundary matrices* (by considering a Lagrange Multiplier of each column vector of *M'* for the boundary e.g. $M'(\mathrm{A},j) = \beta$) in each class, which are closest to *M'*.

For example, when we are considering $g^{(1)}$ = 6 (Class VI) and the matrix *M'* does not contain any 4 × 4 sub-matrix satisfying either TAAT or ATTA, we consider the 2(l – 4 + 1) boundary matrices of *M'* in Class VI as follows. For

each starting position $j = 1, \ldots, l - 4 + 1$, consider the 4×4 sub-matrix M_{sub} of M' formed by columns j to $j + 4 - 1$ of M'. If M_{sub} does not satisfy ATTA because some entries in M_{sub} are less than β, we set these entries to β and decrease the values of the rest entries proportionally. When $\beta = 0.8$, we will modify the following sub-matrix M_{sub}

$$\begin{pmatrix} 0.03 & 0.8 & 0.3 & 0.1 \\ 0.03 & 0.1 & 0.4 & 0.05 \\ 0.04 & 0.1 & 0.1 & 0.05 \\ 0.9 & 0 & 0.2 & 0.8 \end{pmatrix} \text{ to } \begin{pmatrix} 0.03 & 0.8 & 0.8 & 0.1 \\ 0.03 & 0.1 & 0.4\times 0.2/0.7 & 0.05 \\ 0.04 & 0.1 & 0.1\times 0.2/0.7 & 0.05 \\ 0.9 & 0 & 0.2\times 0.2/0.7 & 0.8 \end{pmatrix}$$

to form a boundary matrix of M'. We can prove that either matrix M' or one of its boundary matrices in each motif class can maximize the expected log likelihood when $Z_i^{(0)}$ is fixed. Thus, we can set $M^{(1)}$ to be the matrix with the largest expected log likelihood.

We can repeat the E-step and M-step for a fixed number (10 is used in our experiments) of times to find the motif matrix with maximum expected log likelihood locally.

3.3. *Seed Matrices*

In order to initiate the EM-step, we should have a set of seed matrices $M^{(0)}$, background probability $B^{(0)}$, prior probability $\lambda_b{}^{(0)}$ and motif class $g^{(0)}$. Similar to Bailey and Elkan [2], when the motif length l is short, we convert each length-l DNA sequence S into a seed matrix $M^{(0)}$ by setting

$$M^{(0)}(\alpha, j) = \begin{cases} 0.7 & \alpha = S[j] \\ 0.1 & \alpha \neq S[j] \end{cases}$$

However, when the motif length l is long, as the number of seeds increases exponentially with l, it is impossible to try all seeds. Fortunately, real biological motifs usually contain a conserved region in the center (column vector with one or two entries having high probabilities) or conserved regions at two ends. Instead of considering all 4^l seeds, we consider all length-l' seeds where $l' < l$ and extend these length-l' seeds to length-l by adding column vectors with all entries equal to 0.25 at both ends to represent motifs with a conserved region in the center. Similarly, we construct a seed with all entries equal to 0.25 at the center to represent motifs with conserved regions at both ends.

Apart from $M^{(0)}$, we set the background probability $B^{(0)}$ to be the occurrence probability of each nucleotide in the input sequence $B^{(0)}(\alpha) = (\sum_{i=1}^{w}\sum_{j=1}^{l} \mathrm{I}(X_i[j] = \alpha))/(wl)$. We also set the prior probability $1 - \lambda_b{}^{(0)}$ of a substring being an instance of the motif to be the number of input sequences over w (we assume each input sequence contains one instance of the motif) and set the

motif class $g^{(0)} = 5$, which means that there is no restriction on the motif matrix $M^{(0)}$.

Table 2. Experimental results on real biological data for transcription factors of Drosophila for output with 1 and 30(in brackets) predicted motif(s) per data set.

Factor Name	*l*	*g*	Predicted g	DIMDom (class V only)	DIMDom	MEME
Ac	8	III	**III (III)**	0 (0.6667)	**0.6667 (0.6667)**	0 (0.5)
adf-1	11	V	II (II)	0.2 (0.1667)	**0.2 (0.33)**	0.1111 (0.1111)
AP-1	9	-	**-** (IV)	0 (0.25)	0 **(1)**	0 (0.5)
AS-CT3	6	III	**III (III)**	0.5 (0.5)	**0.5 (0.5)**	0.3333 (0.3333)
Bcd	8	VI	**VI (VI)**	0 (0.3333)	**0.2308 (0.3529)**	0.0227 (0.2)
Bfactor	4	**-**	- (VI)	0 (0)	0 (0)	0 **(0.2222)**
CF1	9	II	- (II)	0 (0.3333)	0 **(1)**	0 (0.5)
Ci	9	**-**	II (I)	0.1667 (0.2)	0.1429 (0.2143)	**0.25 (0.5)**
D_MEF2	10	**-**	- (III)	0 (0.3333)	0 **(0.3333)**	0 (0)
D1	11	**-**	IV (IV)	0 (0.1818)	0 **(0.2857)**	**0.0476** (0.0870)
DREF	14	**-**	- (VI)	0 (0.1429)	0 **(0.3333)**	0 (0.1429)
Dri	10	**-**	IV (IV)	0 (0.25)	**0.5 (0.5)**	0 **(0.5)**
DTF-1	6	**-**	I (I)	0.5	**0.1667** (0.1667)	0.125 **(0.5)**
E74A	17	V	IV (IV)	0.3077 (0.375)	**0.3333 (0.6667)**	0.1818 (0.4)
EcR	7	II	III (IV)	0 (0.5)	**0.3333 (0.5)**	0 (0.3333)
Elf-1	8	**-**	I (I)	0 (0.2222)	0 **(0.6667)**	**0.1** (0.4444)
En	7	VI	- (I)	0 (0.25)	0 **(0.25)**	0 (0.1)
Exd	20	VI	IV (II)	0.3333 (0.3333)	**0.3333 (0.6667)**	0.2 (0.4)
Ftz	12	VI	**VI (VI)**	0 (0.2813)	0.1429 **(0.25)**	**0.1471** (0.1875)
FTZ-F1	7	II	**II (II)**	0 (0)	**0.5 (0.5)**	0 (0)
GAGA	11	I	**I (I)**	0.0476 (0.2941)	**0.1579** (0.1579)	0 **(0.1818)**
GCM	13	**-**	III (IV)	0.0588 (0.2307)	**0.3333 (0.3333)**	0 (0.25)
H	10	III	- **(III)**	0 (0.3333)	0 **(1)**	0 (0.3333)
Hb	10	I	III (IV)	0 (0.1333)	0 (0.2142)	**0.1667 (0.25)**
HSTF	15	VI	**VI (VI)**	0.0909 (0.2222)	0.1111 **(0.25)**	**0.1429** (0.1667)
Kr	10	I	II (VI)	0 (0.2857)	**0.0833 (0.2667)**	0 (0.25)
Sc	8	III	**III (III)**	0 (0.6667)	**0.6667 (0.6667)**	0 (0.5)
Sn	13	I	IV (IV)	0 (0.2727)	**0.2857 (0.5)**	0.0667 (0.3333)
Su_Hw	12	I	- (IV)	0 (0.25)	0 **(1)**	0 (0.5)
TAB	15	-	- (II)	0 (0.2857)	0 **(0.5)**	0 (0.3333)
TBP	7	-	- (I)	0 (0.2)	0 **(0.25)**	0 **(0.25)**
TII	8	II	I (VI)	0 (0.1111)	**0.1176** (0.1176)	0.0526 **(0.1667)**
Ttk69k	8	I	IV **(I)**	0.0909 (0.3333)	0 (0.4286)	**0.2143** (0.2143)
Ubx_a	19	VI	II (II)	0.25 (0.25)	**1 (1)**	**1 (1)**
Zen-1	8	VI	IV **(VI)**	0 (0.1818)	0 (0.2222)	**0.0435 (0.2353)**
Zen-2	8	VI	**VI (VI)**	0.1429 (0.375)	**0.1 (0.5)**	0.05 (0.1667)
Zeste	11	V	IV (I)	0.0192 (0.1224)	0.05 (0.2)	**0.4222 (0.4222)**
Zeste_b	11	**-**	IV (I)	0.0192 (0.1224)	0.05 (0.2)	**0.4222 (0.4222)**
Average score				0.0998 (0.2761)	**0.2501 (0.4471)**	0.1925 (0.3141)

4. Experimental Results

We have implemented DIMDom using C++ and have compared its performance

with that of the popular motif discovery algorithm MEME [2], which is also based on an EM approach, on real biological motif from the TRANSFAC database (http://www.gene-regulation.com). For each transcription factor with at least one known binding site in fruit fly (Drosophila), we searched for all genes regulated by that transcription factor and used the 450 bp (base pairs) upstream and 50 bp downstream of the transcriptional start site of these genes as the input sequences.

We set $l' = 8$ when constructing seed matrices and considered a substring X_i as a binding site if $1 - Z_i \geq 0.9$ for a 90% confidence. Higher thresholds, such as 0.95 and 0.99, failed to give satisfactory results as the number of predicted binding sites decreased sharply to almost zero.

A score for each predicted motif is defined as:

$$\text{score} = \frac{|\text{predicted sites} \cap \text{published sites}|}{|\text{predicted sites} \cup \text{published sites}|}$$

A published binding site is *correctly predicted* if that binding site overlaps with at least one predicted binding site. The score is in the range of [0,1]. When all the published binding sites are correctly predicted without any mis-prediction, score = 1. When no published binding site is predicted correctly, score = 0.

The value of the threshold β used in calculating probability $P(M \mid g)$ was determined by performing tests on another set of real data from the SCPD database (http://rulai.cshl.edu/SCPD/) for yeast (Saccharomyces cerevisiae). DIMDom had the highest average score when $\beta = 0.9$. A smaller value of β did not give better performance because the values of $\log(P(M \mid g))$ were similar for different motif classes. As a result, DIMDom could not take much advantage of different motif classes and motifs from class V were predicted most of the time.

Table 2 shows the performance of MEME [2] and DIMDom on two types of output, only one predicted motif and 30 predicted motifs (from now on, all results related to outputs with 30 predicted motifs will be parenthesised). In order to have a fair comparison with our experiments, we have ignored the known prior probabilities of different motif classes and set them all equal. We have also performed experiments on a version of DIMDom which considers only the class V (*basic EM-algorithm*) so as to illustrate the improvement in performance by introducing the knowledge of different motif classes. It is not surprising to find that MEME (with average score 0.1925 (0.3141)) performed better than the basic EM-algorithm (with average score 0.0998 (0.2761)). However, after introducing the five motif classes, DIMDom (with average score 0.2501 (0.4471)) outperformed MEME when the same set of parameters were

used. Note that DIMDom was about 1.5 times more accurate than MEME when 30 predicted motifs could be outputted.

Among the 47 data sets, both DIMDom and MEME failed to predict any published binding sites in 19 (9) data sets and DIMDom had a better performance (higher score) for 17.5 (27.5) data sets while MEME had a better performance for 10.5 (10.5) data sets only. When the output has 30 predicted motifs, DIMDom outperformed MEME with 2.5 times in the number of successes. In 5.5 out of 10.5 cases for which MEME could do better than DIMDom, MEME predicted only 1 or 2 out of many not-so-similar binding sites because of the high threshold (0.9) used by DIMDom.

Even with a simple description of motif classes, DIMDom can correctly predict the motif classes in 9 (12) out of 21 (25) instances. We expect better prediction results if more parameters are used to describe motif classes [17]. However, more training data are needed for tuning these parameters.

5. Conclusion

We have incorporated biological information, in terms of prior probabilities and pattern characteristics of possible motif classes, into the EM algorithm for discovering motifs and binding sites of transcription factors. Our algorithm DIMDom was shown to have better performance than the popular software MEME. DIMDom will have potentially even better performance if more motif classes are known and included in the algorithm. Like many motif discovery algorithms, DIMDom will work without the length of the motif being given. When the length of the motif is specified, DIMDom will certainly have better performance than when the length is not given and the likelihoods of motifs of different lengths must be compared.

References

1. W. Atchley and W. Fitch, *Proc. Natl Acad. Sci.*, **94**, 5172-5176 (1997).
2. T. Bailey and C. Elkan, *ISMB*, 28-36 (1994).
3. F. Chin, H. Leung, S.M. Yau, T.W. Lam, R. Rosenfeld, W.W. Tsang, D. Smith and Y. Jiang, *RECOMB04*, 125-132 (2004).
4. E. Eskin, *RECOMB04*, 115-124 (2004).
5. S. Keles, M. Lann, S. Dudoit, B. Xing and M. Eisen, *Statistical Applications in Genetics and Molecular Biology*, **2**, Article 5 (2003).
6. C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald and J. Wootton, *Science*, **262**,208-214 (1993).

7. C. Lawrence and A. Reilly, *Proteins: Structure, Function and Genetics*, **7**,41-51 (1990).
8. H. Leung and F. Chin, *JBCB*, **4**, 43-58 (2006).
9. H. Leung and F. Chin, *WABI*, 264-275 (2005).
10. H. Leung and F. Chin, *Bioinformatics*, 22(supp 2), ii86-ii92 (2005).
11. H. Leung and F. Chin, *Bioinformatics* (to appear)
12. H. Leung, F. Chin, S.M. Yiu, R. Rosenfeld and W.W. Tsang, *JCB*, 12(6), 686-701 (2005).
13. M. Li, B. Ma, and L. Wang, *Journal of Computer and System Sciences*, **65**, 73-96 (2002).
14. J.S. Liu, A.F. Neuwald and C.E. Lawrence, *Journal of the American Statistical Association*, **432**, 1156-1170 (1995).
15. K. Maclsaac, D. Gordon, L. Nekludova, D. Odom, J. Schreiber, D. Gifford, R. Young and E. Fraenkel, *Bioinformatics*, **22(4)**, 423-429 (2006).
16. N.J. Mulder et al, *cleic Acids Res.*, **31**, 315-318 (2003).
17. L.Narlikar, R. Gordan, U. Ohler and A. Hartemink, *Bioinformatics*, **22(14)** e384-e392 (2006).
18. L. Narlikar and A. Hartemink, *Bioinformatics*, **22(2)**, 157-163 (2006).
19. C. Pabo and R. Sauer, *Annu. Rev. Biochem.*, **61**, 1053-1095 (1992).
20. P. Pevzner and S.H. Sze, *ISMB*, 269-278 (2000).
21. A. Sandelin and W. Wasserman, *JMB*, **338**, 207-215 (2004).
22. S. Sinha and M. Tompa, *BIBE*, 214-220 (2003).
23. S. Wolfe, L. Nekludova and C.O. Pabo, *Annu. Rev. Biomol. Struct.*, **3**, 183-212 (2000).
24. E. Xing and R. Karp, *Nati. Acad. Sci.*, **101**, 10523-10528 (2004).
25. J. Zilliacus, A.P. Wright, D.J. Carlstedt and J.A. Gustafsson, *Mol. Endocrinol.*, **9**, 389-400 (1995).