Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 853461, 10 pages
http://dx.doi.org/10.1155/2015/853461

Research Article

An Affinity Propagation-Based DNA Motif Discovery Algorithm

Chunxiao Sun, Hongwei Huo, Qiang Yu, Haitao Guo, and Zhigang Sun

School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Hongwei Huo; hwhuo@mail.xidian.edu.cn

Received 3 January 2015; Revised 10 June 2015; Accepted 11 June 2015

Academic Editor: Graziano Pesole

The planted (l, d) motif search (PMS) is one of the fundamental problems in bioinformatics, which plays an important role in locating transcription factor binding sites (TFBSs) in DNA sequences. Nowadays, identifying weak motifs and reducing the effect of local optimum are still important but challenging tasks for motif discovery. To solve the tasks, we propose a new algorithm, APMotif, which first applies the Affinity Propagation (AP) clustering in DNA sequences to produce informative and good candidate motifs and then employs Expectation Maximization (EM) refinement to obtain the optimal motifs from the candidate motifs. Experimental results both on simulated data sets and real biological data sets show that APMotif usually outperforms four other widely used algorithms in terms of high prediction accuracy.

1. Introduction

Transcription factor binding sites (TFBSs) are short and conserved nucleotide fragments (usually ≤ 30 bps) in the cis-regulatory regions of genes in DNA sequences. They interact with transcription factors (TFs) and affect the gene expression. Identification of TFBSs, that is, motif discovery [1], is a fundamental problem for its importance to understand the structure and function of gene expression.

In this paper, we focus on the planted (l, d) motif search (PMS) problem [2], a widely accepted formulation of motif discovery problem. Given a set of input n-length DNA sequences $X = \{X_1, X_2, \ldots, X_t\}$ and two nonnegative integers l and d, the aim of the PMS is to find an l-mer M (an l-length string), which occurs in each of the t sequences with up to d mutations. The l-mer M is called a (l, d) motif and each mutation of M is called a motif instance.

The existing algorithms to solve PMS problem include two main categories. One is exact algorithms, most of which use consensus sequences [3] to represent motifs. The exact algorithms are guaranteed to obtain the optimal motif. Recently, the research of exact algorithms mainly concentrates on pattern-driven algorithms. All the l-length string patterns are taken as candidate motifs, and the string patterns occurring in all input sequences with up to d mutations are the motifs. Typical pattern-driven algorithms use various means to reduce time complexity [4–10]. PairMotif [4] selects multiple pairs of l-mer with relatively large distance from the input sequences to restrict the search space. Compared with recently proposed algorithms, PairMotif requires less storage space and runs faster on most PMS problems. PMS5 [7] computes the common d-neighbors of three l-mers using integer programming formulation, which is an efficient algorithm for solving the difficult instances of PMS: (21, 8) and (23, 9). Some other pattern-driven algorithms index the input sequences with a suffix tree to speed up the search of candidate motifs [11–14]. RISOTTO [11] is the fastest algorithm in the family of suffix tree algorithms for PMS problem and can solve the instance (15, 5) in 100 minutes. The initial search space of pattern-driven algorithms is $O(4^l)$. Therefore, pattern-driven algorithms are feasible for small motif length l ($l \leq 20$), but they will take long running time or have high space requirement with the increase of the motif length.

The other category is approximate algorithms, which commonly use position weight matrixes (PWMs) [15] to represent motifs. They can report results in a short time but often get trapped in local optimal solutions. Most approximate algorithms attempt to maximize the score function of how likely a subsequence of an input sequence is a motif instance, using statistical analysis [16–23]. MEME [18] and Gibbs sampling [20] are well-known approximate algorithms. MEME finds motifs by optimizing the PWMs using the Expectation Maximization (EM). Based on MEME, there are some extension algorithms like Projection [21] and MCEMDA [22]. Projection projects all l-mers from the input sequences onto many buckets by hashing and then derives the consensus sequences to select some valid buckets. After the effective initialization step, EM algorithm is used for refinement. MCEMDA is a modification of the EM algorithm in that the expectation in the E-step is computed numerically through Monte Carlo simulation. Gibbs sampling is a Markov Chain Monte Carlo (MCMC) approach. Based on Gibbs sampling strategy, there are some modifications that have also been described [24, 25]. One that stands out is AlignACE [25], which is a Gibbs sampling algorithm for identifying the over-represented motifs in a set of DNA sequences. Furthermore, some graph-theoretic methods either based on clustering or on heuristic search have also been introduced in the field of motif discovery [26–28]. CRMD [26] uses an entropy-based clustering to find good starting candidate motifs from the input sequences and then employs an effective greedy refinement to search for optimal motifs from the candidate motifs. VINE [28] is a graph clustering algorithm for motif discovery by finding t-cliques in a t-graph in polynomial time. Generally, the approximate algorithm has speedy runtime and minimal memory consumption. Sometimes, however, they cannot converge to the global optimal.

In this paper, we propose a new algorithm, APMotif, to solve motif discovery problem. APMotif first applies Affinity Propagation (AP) [29] clustering in DNA sequences to find highly conserved candidate motifs. APMotif then employs an effective EM refinement to search for optimal motifs from the candidate motifs. Experimental results show that APMotif has competitive prediction accuracy compared to that of previously developed algorithms.

2. Materials and Method

Here, we first briefly describe the original Affinity Propagation clustering and Expectation Maximization algorithms used in the remainder of the paper. We then construct the similarity matrix for motif discovery. Finally, we describe the APMotif algorithm.

2.1. Affinity Propagation (AP). Compared with other clustering approaches, AP clustering is an effective and fast clustering algorithm, especially for large data sets. Given a set of data points $X = \{X_1, X_2, \ldots, X_t\}$, AP clustering takes as input a collection of real valued similarities $s(i,k)$ between the pairs X_i and X_k, $i,k \in \{1,2,\ldots,t\}$. According to the similarities between data points, AP clustering recursively calculates two types of messages: the responsibility $r(i,k)$, reflecting the suitability of point X_k as the exemplar for point X_i, and the availability $a(i,k)$, indicating how appropriate it would be for point X_i to choose point X_k as its exemplar:

$$\begin{aligned} r(i,k) &= s(i,k) - \max_{X_{k'} \neq X_k} \left\{ a\left(i,k'\right) + s\left(i,k'\right) \right\}, \\ a(i,k) &= \min \left\{ 0, r(k,k) + \sum_{X_{i'} \neq \{X_i, X_k\}} \max \left\{ 0, r\left(i',k\right) \right\} \right\} \quad \text{if } X_i \neq X_k, \\ a(k,k) &= \sum_{X_{i'} \neq X_k} \max \left\{ 0, r\left(i',k\right) \right\}. \end{aligned} \tag{1}$$

Upon convergence, AP clustering selects a subset of data points as exemplar and assigns every nonexemplar point to exactly one exemplar. The exemplar $e(i) = X_k$ associated with point X_i is finally defined as follows:

$$e(i) = \arg\max_{X_k} \{ r(i,k) + a(i,k) \}. \tag{2}$$

The AP clustering is terminated when the exemplar remains unchanged for a user-set number of iterations.

2.2. Expectation Maximization (EM). For EM algorithm, given the DNA sequences $X = \{X_1, X_2, \ldots, X_t\}$, each sequence consists of two components which model the motif and nonmotif ("background") positions in the sequence. The starting positions of the motif in each sequence are unknown and represented by the variables ("missing data") $Z = \{Z_{i,j} \mid 1 \leq i \leq t, 1 \leq j \leq n-l+1\}$, where $Z_{i,j} = 1$ if a motif starts at position j in the sequence X_i, and $Z_{i,j} = 0$ otherwise.

EM algorithm attempts to maximize the expectation of the logarithm of the joint likelihood of the model.

The main procedure of EM algorithm repeats iteratively the following two steps:

$$\text{E-step:} \quad Z^{(T)} = \underset{(Z|X,\theta^{(T)})}{E} [Z], \tag{3}$$

$$\text{M-step:} \quad \theta^{(T+1)} = \arg\max_{\theta} \underset{(Z|X,\theta^{(T)})}{E} \left[\log p(X, Z \mid \theta) \right]. \tag{4}$$

In (4), the logarithm of the joint likelihood of the model is defined as follows:

$$\log p(X, Z \mid \theta) = \sum_{i=1}^{t} \sum_{j=1}^{n-l+1} Z_{i,j} \log p\left(X_i \mid Z_{i,j} = 1, \theta\right), \tag{5}$$

where

$$\theta = [\theta_0, \theta_1] = [P_0, P_1, P_2, \ldots, P_l] = [P_{w,m}]_{4\times(l+1)} \tag{6}$$

is the vector containing all the parameters of the model and $P_{w,m}$ is the probability of the character $w \in \{A, T, C, G\}$ occurring at either a background position ($m = 0$) or a motif position ($1 \leq m \leq l$).

In (5), the conditional probability for a sequence containing a motif is defined as follows:

$$\log p\left(X_i \mid Z_{i,j}=1,\theta\right) = \sum_{k=0}^{l-1} I\left(i, j+k\right)^T \log P_k + \sum_{k\in\Delta_{i,j}} I\left(i,k\right)^T \log P_0, \tag{7}$$

where $I(i,j)$ indicates a vector whose entries are all zeros except the one corresponding to the character at position j in the sequence X_i. $\Delta_{i,j}$ is the set of positions of the background in the sequence X_i.

2.3. Construction of Similarity Matrix for Motif Discovery. In the original AP clustering, given two random l-mers x_i and x_k from t DNA sequences $X = \{X_1, X_2, \ldots, X_t\}$, the similarity is set as the negative Hamming distance between l-mers x_i and x_k; that is, $s(i,k) = -d_H(x_i, x_k)$ [29], which cannot describe the property of DNA sequences clustering effectively. According to the feature of PMS that two motif instances of the same motif cannot differ by more than $2d$ positions, and the maximum similarity principle, we employ pairwise constraints and variable-similarity measure [30] to modify the similarity as follows:

$$s\left(i,k\right) = -\rho \times d_H\left(x_i, x_k\right) \times L\left(x_i, x_k, X\right), \tag{8}$$

where

$$\rho = \begin{cases} R_1 & \text{if } d_H\left(x_i, x_k\right) \in (0, d] \\ R_2 & \text{if } d_H\left(x_i, x_k\right) \in (d, 2d] \\ +\infty & \text{if } d_H\left(x_i, x_k\right) \in (2d, 4d], \end{cases}$$
$$L\left(x_i, x_k, X\right) = \begin{cases} +\infty & \text{if } x_i \in_l X_p,\ x_k \in_l X_q,\ p = q \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

$R_1 \in (1, +\infty)$, $R_2 \in (0, 1]$, and $x_i \in_l X_p$ denotes x_i is an l-mer of the sequence X_p.

Based on the similarity in (8), the similarity between data points is more accurate and only tiny subsets of the data points are required to exchange messages, so AP clustering can not only increase clustering accuracy but also decrease runtime. Its theoretical analyses are shown in Section 3.1.

According to the two similarities: $s(i,k) = -d_H(x_i, x_k)$ and $s(i,k) = -\rho \times d_H(x_i, x_k) \times L(x_i, x_k, X)$, take the PMS instance (15, 4) with 20 sequences of different length between 100 and 1000 as an example; we show the comparison of runtime and clustering accuracy in Figure 1.

2.4. APMotif Algorithm. Under the assumption of exactly one occurrence of motif instance per sequence (OOPS) [1], to find the motif instances from the input DNA sequences $X = \{X_1, X_2, \ldots, X_t\}$, APMotif algorithm consists of the following stages:

(1) *Constructing Clusters*. Select the sequence X_1 as the reference sequence, for each l-mer x_k ($k = 1, 2, \ldots, n - l + 1$) in X_1 (reference subsequence), and construct cluster $C(x_k, X)$, which is the set composed by all the l-mer x' in $X - \{X_1\}$ that $d_H(x_k, x') \le 2d$ and the l-mer x_k.

(2) *Extracting Clusters*. For each cluster $C(x_k, X)$, use AP clustering and a filtering rule to generate a highly conserved cluster $C'(x_{k1}, X)$.

(3) *Refining Clusters*. For each filtered cluster $C'(x_{k1}, X)$, use EM refinement to obtain the distribution θ_{k1} and the objective function Q_{k1} of each cluster $C'(x_{k1}, X)$.

(4) *Verifying Motif Instances*. With the maximum distribution $\theta_{\max}$ and the maximum objective function $Q_{\max}$, the l-mer y having the maximum log-likelihood $\log p(y \mid \theta_{\max})$ in each sequence is verified as a motif instance.

Based on the four stages, the APMotif algorithm is presented as in Algorithm 1.

In line (1), the set of the (l, d) motif instances M is initialized to an empty set. Lines (2)-(3) show the stage of constructing clusters. Lines (4)-(5) show the stage of extracting clusters. Lines (6)-(7) show the stage of refining clusters. Lines (8)–(12) show the stage of verifying motif instances. APMotif can discover the (l, d) motif instances in high prediction accuracy and output them in line (13).

Next, we explain each stage in detail.

Stage 1 (construct clusters). The construction of clusters keeps the following simple observation that the Hamming distance between two motif instances of the same motif must be less than or equal to $2d$. Generally, we choose the first sequence X_1 as the reference sequence. As we do not know in advance which l-mer x_k in X_1 is the motif instance, all the l-mers x_k ($k = 1, 2, \ldots, n - l + 1$) in X_1 are regarded as the reference subsequences. Given an l-mer x_k in X_1, the selected l-mers x' in other sequence X_i ($i = 2, \ldots, n - l + 1$) should satisfy $d_H(x_k, x') \le 2d$, denoted as $B(x_k, X_i) = \{x' : x' \in_l X_i, d_H(x_k, x') \le 2d\}$, where $x' \in_l X_i$ denotes that x' is an l-mer of X_i. The cluster corresponding to the reference subsequence x_k is denoted as

$$C\left(x_k, X\right) = \{x_k\} \cup \bigcup_{i=2}^{n-l+1} B\left(x_k, X_i\right). \tag{10}$$

The average number of l-mers in the cluster $C(x_k, X)$ is $p_{2d} \times t \times (n - l + 1)$, where

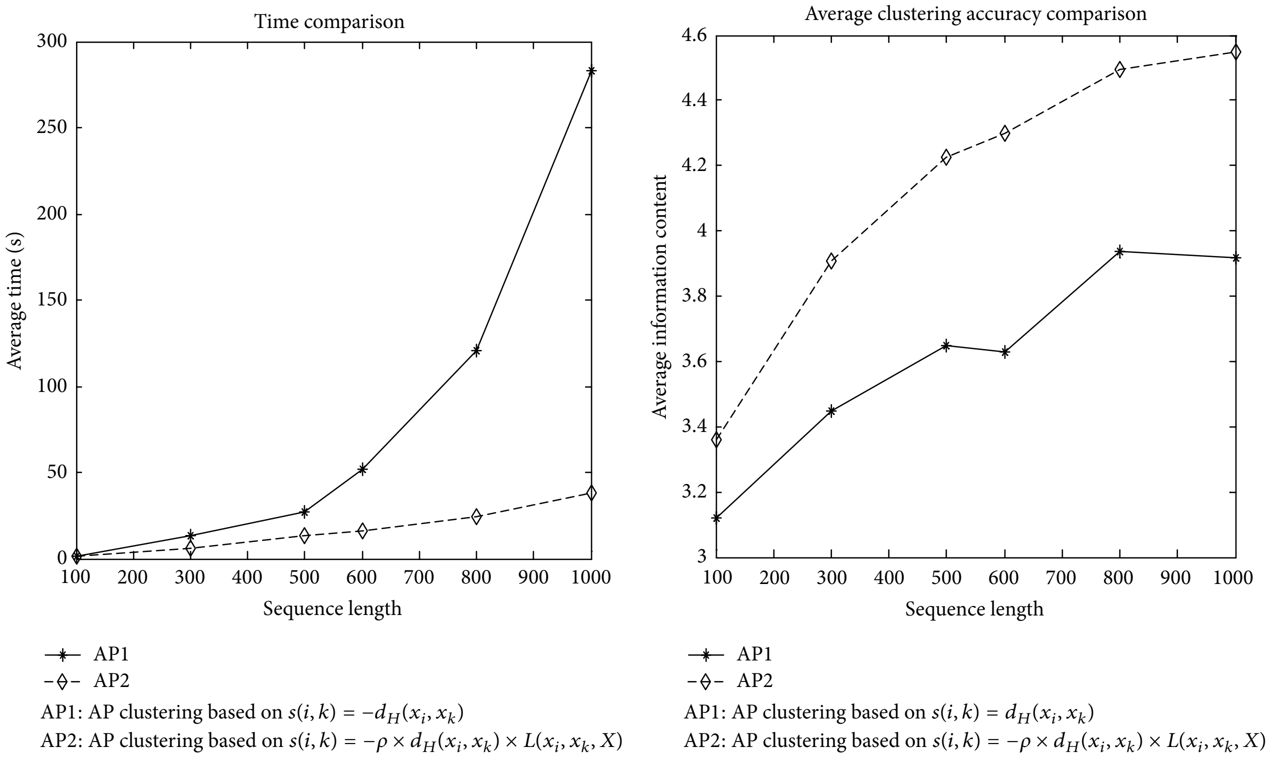
$$p_{2d} = \sum_{i=0}^{2d} \binom{l}{i} \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^{l-i} \tag{11}$$

is the probability that the Hamming distance between two random l-mers is at most $2d$.

Stage 2 (extract clusters). For each cluster subset $C(x_k, X)$, we use the AP clustering to produce the high conserved cluster $C'(x_k, X)$ that contains the reference subsequence x_k. If one of the reference subsequences x_k ($k = 1, 2, \ldots, n - l + 1$) is a motif instance, the corresponding cluster $C'(x_k, X)$ may be the true motif model.

```
Input: $l, d, X = \{X_1, X_2, \ldots, X_t\}$
Output: $(l, d)$ motif instances set $M$
$M \leftarrow \Phi$
for each $l$-mer $x_k \in X_1$, $1 \le k \le n - l + 1$ do
    Construct cluster $C(x_k, X)$
for each $C(x_k, X)$ do
    Use AP clustering and a filtering rule to generate cluster $C'(x_{k1}, X)$
for each $C'(x_{k1}, X)$ do
    Use EM algorithm to generate $Q_{k1}$ and $\theta_{k1}$
Calculate $Q_{\max} \leftarrow \max\{Q_{k1}\}$, $\theta_{\max} \leftarrow \max\{\theta_{k1}\}$
for $i \leftarrow 1$ to $t$ do
    for each $l$-mer $y \in X_i$ do
        Calculate $\arg\max_y \log p(y \mid \theta_{\max})$
        Add $y$ to $M$
Output $M$
```

Algorithm 1: APMotif.



(a)

(b)

Figure 1: AP clustering results.

For each cluster $C'(x_k, X)$, two types of metrics, information content (IC) and complexity scores [31], are employed to assess the quality of the cluster. The information content of the cluster $C'(x_k, X)$ is defined as

$$Q_k = \mathrm{IC}\left(C'\left(x_k, X\right)\right) = \sum_{m=1}^{l} \sum_{w=1}^{4} p_{w,m} \log \frac{p_{w,m}}{p_{w,0}}, \qquad (12)$$

where $p_{w,m}$ represents the probability of each character $w \in \{A, T, C, G\}$ appearing at the position m of the l-mer, and where $p_{w,0}$ is the background probability of character w. A higher IC value indicates a stronger potential of a cluster to be the true motif model.

The complexity score of the cluster $C'(x_k, X)$ is defined as

$$J\left(C'\left(x_k, X\right)\right) = \left(\frac{1}{4}\right)^{l} \prod_{w=1}^{4} \left(\frac{l}{\sum_{m=1}^{l} p_{w,m}}\right)^{\sum_{m=1}^{l} p_{w,m}}. \qquad (13)$$

Note that the IC value cannot completely reflect the conservation of the motif model. The reason is that many noninformative repeated l-mers may lead to a higher IC value. Fortunately, these false positive clusters have lower complexity scores and they can be effectively filtered out.

Taking these into account, we propose the following rule to filter out some unqualified clusters.

Rule. If $J(C'(x_{k1}, X)) > (1/(n-l+1))\sum_{k=1}^{n-l+1} J(C'(x_k, X))$ and $\mathrm{IC}(C'(x_{k1}, X)) > (1/(n-l+1))\sum_{k=1}^{n-l+1} \mathrm{IC}(C'(x_k, X))$, the cluster $C'(x_{k1}, X)$ will be stored as a candidate motif model.

According to the rule, a few clusters $C'(x_{k1}, X)$ with high IC value and high complexity scores are stored for EM refinement in Stage 3.

Stage 3 (refine clusters). It is important to note that AP clustering is primarily an initialization strategy that produces starting points for EM refinement. Taking each cluster $C'(x_{k1}, X)$ as a starting point, we use the modified EM refinement to search for a motif model.

The E-step of EM calculates the expected value of the missing information $Z_{i,j}$, which is the probability that a motif starts in position j of sequence X_i.

E-Step. Consider

$$Z_{i,j}^{(T)} = \frac{p\left(X_i \mid z_{i,j} = 1, \theta^{(T)}\right)}{\sum_{j=1}^{n-l+1} p\left(X_i \mid z_{i,j} = 1, \theta^{(T)}\right)}. \quad (14)$$

The M-step of EM reestimates distribution θ by maximizing the expected log-likelihood.

M-Step. Consider

$$\begin{aligned} p_{w,m}^{(T)} &= \frac{c_{w,m} + \xi_m}{\sum_{w\in\Omega}\left(c_{w,m} + \xi_m\right)} \quad w \in \Omega = \{A, T, C, G\}, \\ c_{w,m} &= \sum_{i=1}^{t}\sum_{j=1}^{n-l+1} z_{i,j}^{T} I\left(i, j+k-1\right), \\ c_{w,0} &= c_w - \sum_{m=1}^{l} c_{w,m}, \end{aligned} \quad (15)$$

where ξ_m is the pseudocount to deal with the zero frequencies and c_w is the total number of the character w in all sequence X.

The EM algorithm is terminated when the object function Q, that is, Information Content, remains unchanged. After EM refinement, we can obtain the distribution θ_{k1} and the objective function Q_{k1} of each cluster $C'(x_{k1}, X)$.

Stage 4 (select motif instances). Comparing each distribution θ_{k1}, we find the maximum one $\theta_{\max}$. For the distribution $\theta_{\max}$, an l-mer y in one sequence with the maximum log-likelihood that is considered as a candidate motif instance:

$$\log p\left(y \mid \theta_{\max}\right) = \max \sum_{m=1}^{l} \log p_{w,m}. \quad (16)$$

Meanwhile, a candidate motif instance y should satisfy $d_H(y, x_{\mathrm{motif}}) \leq d$, where x_{motif} is the motif by using $\theta_{\max}$ as the consensus.

Thus, the l-mer y that has the maximum log-likelihood under the distribution $\theta_{\max}$ and satisfies $d_H(y, x_{\mathrm{motif}}) \leq d$ is stored in the set of motif instances M.

3. Results and Discussion

Here, we first theoretically analyze the probability of $s(i,k) = -\infty$ and give its formula. We then show the experimental results of APMotif both on simulated data sets and real biological data sets.

3.1. Analysis of Similarity Matrix. It has been pointed out in [29] that the sparsity of the similarity matrix will lead to fast calculation since the information propagation needs not be performed if $s(i,k) = -\infty$.

Given two random l-mers x_i and x_k, coming from different sequences, which differ from the same l-mer x_0 with up to $2d$ positions, the distance relationships between x_i, x_k, and x_0 satisfy $0 \leq d_H(x_0, x_i) \leq 2d$ and $0 \leq d_H(x_0, x_k) \leq 2d$. Let $p(\alpha, \beta)$ represent the probability of $d_H(x_0, x_i) = \alpha$ and $d_H(x_0, x_k) = \beta$ corresponding to a sample space $\Omega = \{\langle\alpha, \beta\rangle : 0 \leq \alpha \leq 2d, 0 \leq \beta \leq 2d\}$. Because $d_H(x_0, x_i) = \alpha$ and $d_H(x_0, x_k) = \beta$ are independent of each other, $p(\alpha, \beta)$ can be calculated as follows:

$$\begin{aligned} p\left(\alpha, \beta\right) &= p\left(d_H\left(x_0, x_i\right) = \alpha, d_H\left(x_0, x_k\right) = \beta\right) \\ &= p\left(d_H\left(x_0, x_i\right) = \alpha\right) \times p\left(d_H\left(x_0, x_k\right) = \beta\right), \end{aligned} \quad (17)$$

$$\begin{aligned} p\left(d_H\left(x_0, x_i\right) = \alpha\right) &= \binom{2d}{\alpha}\frac{3^{\alpha}}{4^{2d}}, \\ p\left(d_H\left(x_0, x_k\right) = \beta\right) &= \binom{2d}{\beta}\frac{3^{\beta}}{4^{2d}}. \end{aligned} \quad (18)$$

Let $p(d_H(x_i, x_k) > 2d)$ represent the probability that the Hamming distance between two random l-mers x_i and x_k is more than $2d$.

Using Theorem of Total Probability, we have

$$\begin{aligned} p\left(d_H\left(x_i, x_k\right) > 2d\right) = p\left(d_H\left(x_i, x_k\right)\right. \\ \left. > 2d \mid d_H\left(x_0, x_i\right) = \alpha, d_H\left(x_0, x_k\right) = \beta\right) \times p\left(\alpha, \beta\right), \end{aligned} \quad (19)$$

where $p(d_H(x_i, x_k) > 2d \mid d_H(x_0, x_i) = \alpha, d_H(x_0, x_k) = \beta)$ represents the conditional probability of $d_H(x_i, x_k) > 2d$ given $d_H(x_0, x_i) = \alpha$ and $d_H(x_0, x_k) = \beta$.

Next, we discuss how to calculate the conditional probability $p(d_H(x_i, x_k) > 2d \mid d_H(x_0, x_i) = \alpha, d_H(x_0, x_k) = \beta)$.

According to $d_H(x_i, x_k) \leq d_H(x_0, x_i) + d_H(x_0, x_k)$ and $d_H(x_i, x_k) > 2d$, we can obtain

$$d_H\left(x_0, x_i\right) + d_H\left(x_0, x_k\right) > 2d. \quad (20)$$

For $0 \leq d_H(x_0, x_i) \leq 2d$ and $0 \leq d_H(x_0, x_k) \leq 2d$, (20) can be written as

$$\begin{aligned} d_H\left(x_0, x_i\right) + d_H\left(x_0, x_k\right) &= 2d + 1 + c \\ c &= 0, 1, \ldots, 2d - 1. \end{aligned} \quad (21)$$

Given d, for each c, we can find all the 2-tuple $\langle d_H(x_0, x_i), d_H(x_0, x_k)\rangle = \langle \alpha, \beta\rangle$ that satisfy (21).

Given c, for each 2-tuple $\langle \alpha_0, \beta_0\rangle$, the conditional probability of $d_H(x_i, x_k) > 2d$ can be calculated as follows:

$$p\left(d_H\left(x_i, x_k\right) > 2d \mid d_H\left(x_0, x_i\right) = \alpha_0, d_H\left(x_0, x_k\right) = \beta_0\right) = \frac{\left(\sum_{i=0}^{c}\binom{\alpha_0}{i}\binom{l-\alpha_0}{\beta_0-i}\right)\times 3^{\beta_0}}{\binom{l}{\beta_0}\times 3^{\beta_0}}. \tag{22}$$

Considering all the values $c = 0, 1, \ldots, 2d - 1$ and all the 2-tuple $\langle\alpha, \beta\rangle$, we calculate the conditional probability of $d_H(x_i, x_k) > 2d$ as follows:

$$p\left(d_H\left(x_i, x_k\right) > 2d \mid d_H\left(x_0, x_i\right) = \alpha, d_H\left(x_0, x_k\right) = \beta\right) = \sum_{c=0}^{2d-1}\sum_{\langle\alpha,\beta\rangle}\frac{\left(\sum_{i=0}^{c}\binom{\alpha}{i}\binom{l-\alpha}{\beta-i}\right)\times 3^{\beta}}{\binom{l}{\beta}\times 3^{\beta}}. \tag{23}$$

According to (18) and (23), we can obtain

$$p\left(d_H\left(x_i, x_k\right) > 2d\right) = \left(\sum_{c=0}^{2d-1}\sum_{\langle\alpha,\beta\rangle}\frac{\left(\sum_{i=0}^{c}\binom{\alpha}{i}\binom{l-\alpha}{\beta-i}\right)\times 3^{\beta}}{\binom{l}{\beta}\times 3^{\beta}}\right)\times\binom{2d}{\alpha}\frac{3^{\alpha}}{4^{2d}} \times\binom{2d}{\beta}\frac{3^{\beta}}{4^{2d}}. \tag{24}$$

The probability $p(d_H(x_i, x_k) > 2d)$ is also the probability of $s(i, k) = -\infty$ corresponding to the condition that $d_H(x_i, x_k) \in (2d, 4d]$.

Meanwhile, when the two l-mers x_i and x_k are in the same sequence, the probability of $s(i, k) = -\infty$ in the similarity matrix is $1/(t - 1)$, where t is the sequence number.

For the (15, 4) problem instance with sequence number $t = 20$, the probability of $s(i, k) = -\infty$ obtained by (24) and t is 0.8405, when sequence length n varies from 100 to 1000.

In Table 1, by enumerating all the $s(i, k) = -\infty$ in the similarity matrix, the empirical result shows that the probability of $s(i, k) = -\infty$ accounts for more or less than 84% of the similarity matrix, which is consistent with the theoretical analysis.

3.2. Results on Synthetic Data Sets. We generate the synthetic data sets as follows: first, we generate a motif M of length l and t independent and identically distributed (i.i.d) sequences X of length n. Then, we implant (l, d) instance, which differs from the motif M with up to d positions, into a random position in each sequence.

The nucleotide level performance coefficient (nPC) defined by Pevzner and Sze [2] is used to evaluate the motif prediction accuracy:

$$\text{nPC} = \frac{|K \cap P|}{|K \cup P|}. \tag{25}$$

K is the set of $l \times t$ base positions in the t known motif instances, and P is the corresponding set of $l\times t$ base positions in the t predicted motif instances. The value of nPC is between 0 and 1; the larger the value of nPC, the higher the accuracy of the predicted motif.

Table 1: Number of $-\infty$ related to different sequence length n on (15, 4) instance.

n	Data size[a]	Numbers of $-\infty$	Percentage
100	93	$6.81e + 03$	78.75%
300	308	$8.01e + 04$	84.43%
500	523	$2.29e + 05$	83.89%
600	630	$3.41e + 05$	85.82%
800	846	$5.84e + 05$	81.55%
1000	1061	$9.53e + 05$	84.67%

[a] Data size: the number of all l-mers in one cluster.

Table 2: Prediction accuracy on different (l, d) instances.

(l, d)	nPC				
	Projection	MEME	VINE	Gibbs sampling	APMotif
(11, 3)	92%	65%	95%	56%	96%
(12, 3)	77%	84%	92%	3%	93%
(15, 4)	93%	86%	98%	19%	96%
(16, 5)	64%	71%	95%	2%	94%
(18, 6)	75%	79%	93%	3%	98%
(19, 7)	84%	77%	92%	4%	97%

Table 2 shows the comparison of the mean nPC obtained by APMotif and four other representative algorithms: MEME [16–18], Gibbs sampling [20], Projection [21], and VINE [28]. For each of the (l, d) combinations, all the five algorithms are run once on each of 10 randomly generated sets of input sequences ($t = 20$, $n = 600$). APMotif constitutes a simple and effective method which groups the significant l-mers to form the optimal clusters so that the motif instances can be predicted with high accuracy. APMotif has the highest mean nPC on the instances (11, 3), (12, 3), (18, 6), and (19, 7), and the second highest mean nPC on the instances (15, 4), (16, 5), which proves that APMotif is relatively robust in various problem instances.

In Table 3, we compare the nPC of APMotif on problem instances with longer background sequences. Since the longer a sequence is, the more noisy l-mers will be yielded, this makes it difficult to discover the true motifs. We fix the (l, d) instance as (15, 4) instance, one of the most popular benchmarks for motif discovery problem, and vary the sequence length n from 100 to 1000. For each setting, 10 i.i.d data sets are generated, each containing 20 sequences. The nPC of APMotif is over 95% for sequences of various lengths between 100 and 1000, much greater than that of Projection, MEME, Gibbs sampling, and VINE. The reason why the performance of APMotif is stable over the sequence length is that APMotif has strong ability of filtering noisy l-mers. With each sequence length n increasing ($n \geq 600$), APMotif still maintains its advantage in the prediction accuracy. For example, when the sequence length is 1000 bps, the nPC of Projection, MEME, VINE, and Gibbs sampling are 88%, 76%,

Table 3: Prediction accuracy of different sequence length n on (15, 4) instance.

n	nPC				
	Projection	MEME	VINE	Gibbs sampling	APMotif
100	96%	99%	99%	92%	100%
300	94%	98%	99%	58%	99%
600	89%	91%	97%	19%	98%
800	87%	90%	98%	14%	97%
1000	88%	76%	91%	8%	95%

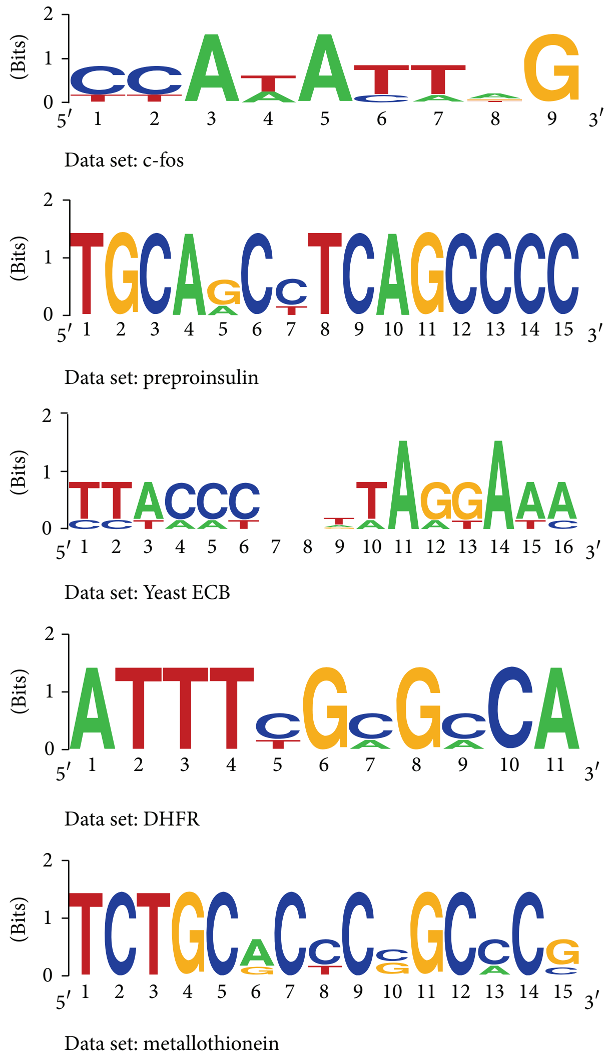
91%, and 8%, respectively, while APMotif algorithm shows its advantage with the nPC 95%.

3.3. Results on Real Biological Data Sets. At first, the performance of APMotif is evaluated on the five widely used real data sets discussed in [21], which are preproinsulin, dihydrofolate reductase (DHFR), c-fos, metallothionein, and Yeast ECB. Because no information about the l and the d of the true motif is known in advance, we select the (l, d) used for each real data set as follows: the value of l is fixed as the length of the reference motif; the value of d is minimum to ensure that the predicted (l, d) instance contains the reference motif. Table 4 shows that the predicted motifs returned by the APMotif algorithm are almost consistent with the reference motifs. In Figure 2, the software Weblogo [32] is used to show the sequence logos of the predicted motifs, which graphically shows the degree of motif conservation measured by relative entropy.

For the five real data sets, Figure 3 compares the nucleotide level performance coefficient of APMotif with that of other popular algorithms. For Yeast ECB, the nPC of APMotif, MEME, Projection, and VINE are 1, which indicates the prediction result is completely correct. For c-fos, preproinsulin, DHFR, and metallothionein, the nPC of APMotif is 0.28, 0.68, 0.37, and 0.82, respectively, greater than that of three other widely used motif finding algorithms.

In addition, we show the prediction performance of APMotif on Tompa data [33], which is set up as the benchmark for testing motif discovery algorithms. For most Tompa data, the distribution of motif in each sequence makes it difficult to report the motif occurrence positions. We select some Tompa data. When a sequence contains more than one motif, it is difficult to discover all the motifs. When some sequences do not contain any motif, it is difficult to discovery motifs in other sequences. Overall, most algorithms have very low prediction accuracy in Tompa data. To improve the prediction accuracy, different algorithms should be executed together to complement each other.

Figure 4 shows the prediction accuracy (nPC) of APMotif and MEME on each selected Tompa data. We observe that, for some data, such as hm08r, hm19r, mus03r, dm04r, and yst02r, the nPC of APMotif is better than that of MEME, but for some other data, such as hm23r, mus04r, dm01r, and yst06r, the nPC of MEME is better than that of APMotif. This phenomenon illustrates the practical significance in combining the results of APMotif and MEME to improve the



Data set: c-fos

Data set: preproinsulin

Data set: Yeast ECB

Data set: DHFR

Data set: metallothionein

Figure 2: Sequence logos of the predicted motifs.

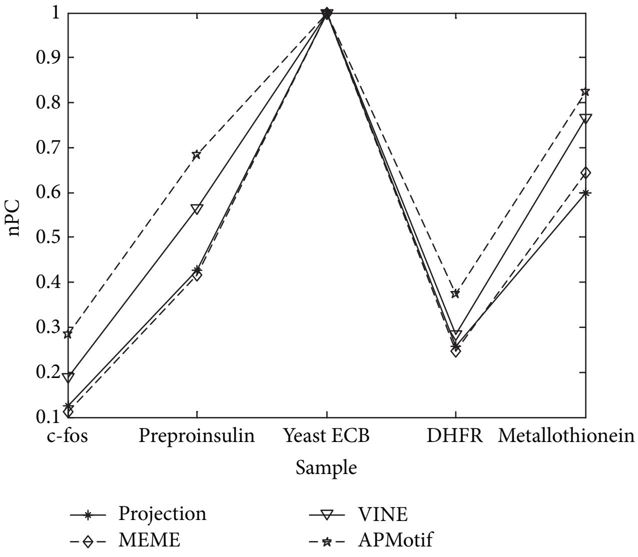


Figure 3: Prediction accuracy on real biological data.

Table 4: Results of APMotif on real biological data.

Data set	Predicted motif	Reference motif	(l, d)
c-fos	CCATATTAG	CCANATTNG	(9, 2)
Preproinsulin	TGCAGCCTCAGCCCC	CAGCCTCAGCCCCAT	(15, 2)
Yeast ECB	TTACCCNNTTAGGAAA	TTTCCCNNTNAGGAAA	(16, 3)
DHFR	ATTTCGCGCCA	ATTTCGCGCCA	(11, 2)
Metallothionein	TCTGCACCCGGCCCG	CTCTGACNCCGCCC	(15, 2)

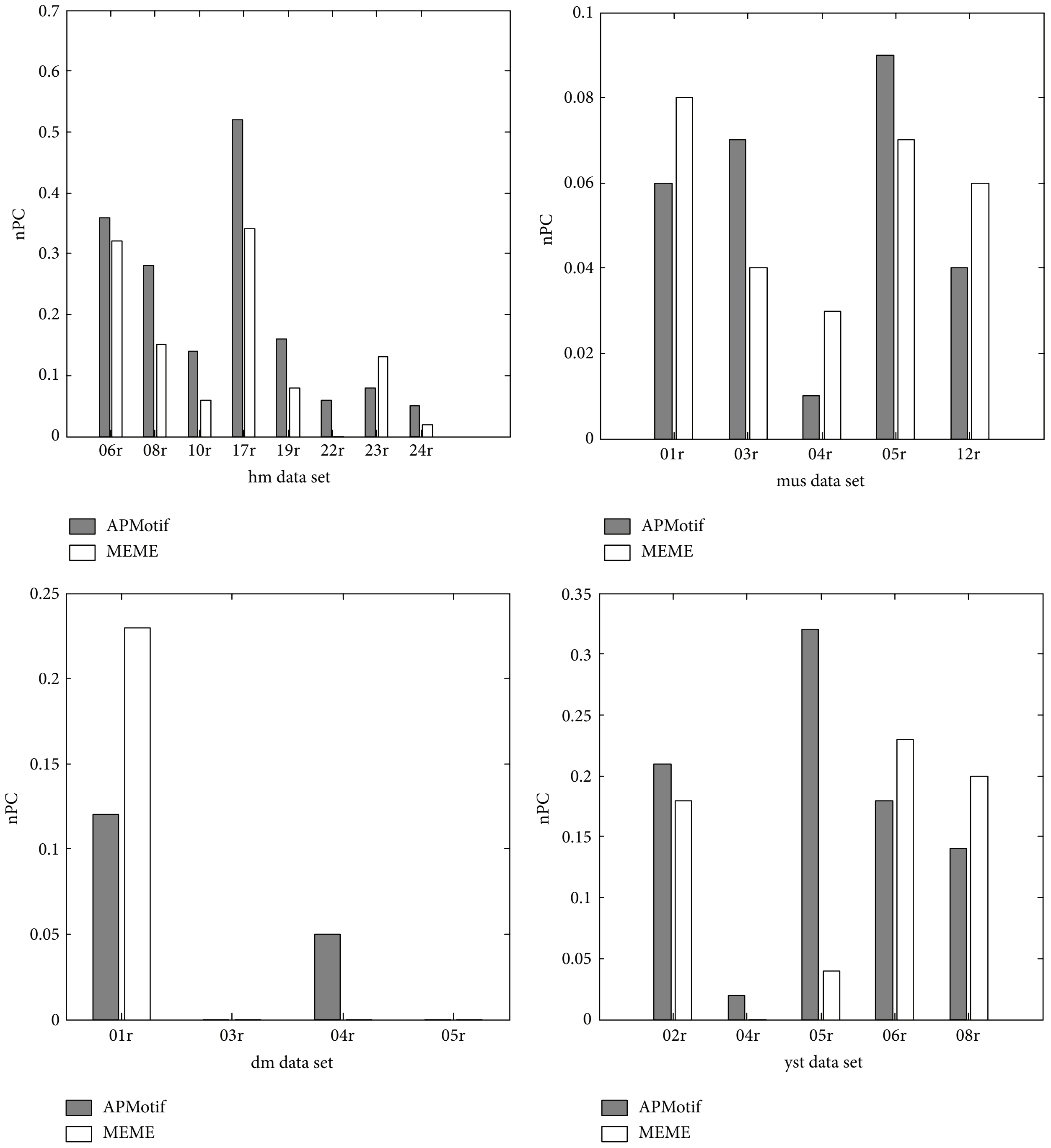


Figure 4: Prediction accuracy on Tompa data.

ability of motif discovery algorithms in identifying TFBSs in higher eukaryotes [33].

4. Conclusions

The planted (l, d) motif search (PMS) problem arises from the need to find transcription factor binding sites (TFBSs) in DNA sequences. In this paper, we propose a new approximate algorithm, APMotif, which overcomes the local maximum drawback to some extent that is inherent in the EM motif-finding algorithms and guarantees that most motifs can be discovered for specific (l, d) settings. APMotif first constructs clusters by computing the Hamming distance between each l-mer in the reference sequence and all the l-mers in other sequences, and then it uses AP clustering combined with two metrics to select the high conserved clusters for the EM refinement progress. After the EM refinement, the cluster with maximum information content is verified as the motif instances. The experimental results on the synthetic data sets show that APMotif indeed removes most useless background information to obtain motifs with high accuracy. The experimental results on the real biological data show that APMotif can discover all or a large part of the motif instances. In summary, the APMotif algorithm outperforms the compared algorithms with significant improvement in prediction accuracy.

In the last years, the introduction of ChIP-Seq data raises new challenges for motif discovery problem from the perspective of data scale. Most existing motif discovery algorithms proposed for small data set are inefficient in dealing with ChIP-Seq data. Since APMotif performs AP clustering for multiple times with each clustering independent of others, APMotif thus features the merit for parallel computing. In our future work, we plan to parallel the APMotif algorithm to be fastened, so that the improved APMotif algorithm can deal with large data set efficiently.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported in part by the National Natural Science Foundation of China (61173205 and 61373044).

References

[1] G. D. Stormo, "DNA binding sites: representation and discovery," *Bioinformatics*, vol. 16, no. 1, pp. 16–23, 2000.

[2] P. A. Pevzner and S.-H. Sze, "Combinatorial approaches to finding subtle signals in DNA sequences," in *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pp. 269–278, San Diego, Calif, USA, August 2000.

[3] T. D. Schneider, "Consensus sequence zen," *Applied Bioinformaitics*, vol. 1, no. 3, pp. 111–119, 2002.

[4] Q. Yu, H. Huo, Y. Zhang, and H. Guo, "PairMotif: a new pattern-driven algorithm for planted (l, d) DNA motif search," *PLoS ONE*, vol. 7, no. 10, Article ID e48442, 2012.

[5] F. Y. L. Chin and H. C. M. Leung, "Voting algorithms for discovering long motifs," in *Proceedings of the 3rd Asia-Pacific Bioinformatics Conference (APBC '05)*, pp. 261–271, January 2005.

[6] J. Davila, S. Balla, and S. Rajasekaran, "Fast and practical algorithms for planted (l, d) motif search," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 544–552, 2007.

[7] H. Dinh, S. Rajasekaran, and V. K. Kundeti, "PMS5: an efficient exact algorithm for the (ℓ, d)-motif finding problem," *BMC Bioinformatics*, vol. 12, article 410, 2011.

[8] E. S. Ho, C. D. Jakubowski, and S. I. Gunderson, "iTriplet, a rule-based nucleic acid sequence motif finder," *Algorithms for Molecular Biology*, vol. 4, no. 1, article 14, 2009.

[9] H. Dinh, S. Rajasekaran, and J. Davila, "qPMS7: a fast algorithm for finding (l, d)-motifs in DNA and protein sequences," *PLoS ONE*, vol. 7, no. 7, Article ID e41425, 2012.

[10] M. Nicolae and S. Rajasekaran, "Efficient sequential and parallel algorithms for planted motif search," *BMC Bioinformatics*, vol. 15, no. 1, article 34, 2014.

[11] N. Pisanti, A. M. Carvalho, L. Marsan, and M.-F. Sagot, "RISOTTO: fast extraction of motifs with mismatches," in *LATIN 2006: Theoretical Informatics*, vol. 7, pp. 757–768, Springer, 2006.

[12] A. Floratou, S. Tata, and J. M. Patel, "Efficient and accurate discovery of patterns in sequence data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 8, pp. 1154–1168, 2011.

[13] M. F. Sagot, "Spelling approximate repeated or common motifs using a suffix tree," in *Proceedings of the 3rd Latin American Symposium on Theoretical Informatics (LATIN '98)*, pp. 374–390, 1998.

[14] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding signals of unknown length in DNA sequences," *Bioinformatics*, vol. 17, supplement 1, pp. S207–S214, 2001, Proceedings of the 9th International Conference on Intelligent Systems for Molecular Biology (ISMB '01).

[15] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673–4680, 1994.

[16] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," in *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pp. 28–36, 1994.

[17] D. Quang and X. Xie, "EXTREME: an online em algorithm for motif discovery," *Bioinformatics*, vol. 30, no. 12, pp. 1667–1673, 2014.

[18] T. L. Bailey, N. Williams, C. Misleh, and W. W. Li, "MEME: discovering and analyzing DNA and protein sequence motifs," *Nucleic Acids Research*, vol. 34, supplement 2, pp. W369–W373, 2006.

[19] G. Z. Hertz and G. D. Stormo, "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences," *Bioinformatics*, vol. 15, no. 7-8, pp. 563–577, 1999.

[20] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, "Detecting subtle sequence signals:

a gibbs sampling strategy for multiple alignment," *Science*, vol. 262, no. 5131, pp. 208–214, 1993.

[21] J. Buhler and M. Tompa, "Finding motifs using random projections," *Journal of Computational Biology*, vol. 9, no. 2, pp. 225–242, 2002.

[22] C. Bi, "A monte carlo em algorithm for de novo motif discovery in biomolecular sequences," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 3, pp. 370–386, 2009.

[23] Q. Yu, H. Huo, Y. Zhang, H. Guo, and H. Guo, "PairMotif+: a fast and effective algorithm for De Novo motif discovery in DNA sequences," *International Journal of Biological Sciences*, vol. 9, no. 4, pp. 412–424, 2013.

[24] W. Thompson, E. C. Rouchka, and C. E. Lawrence, "Gibbs recursive sampler: finding transcription factor binding sites," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3580–3585, 2003.

[25] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church, "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation," *Nature Biotechnology*, vol. 16, no. 10, pp. 939–945, 1998.

[26] G. Li, T.-M. Chan, K.-S. Leung, and K.-H. Lee, "A cluster refinement algorithm for motif discovery," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 4, pp. 654–668, 2010.

[27] S. van Dongen, *Graph clustering by flow simulation [Ph.D. thesis]*, University of Utrecht, 2000.

[28] C.-W. Huang, W.-S. Lee, and S.-Y. Hsieh, "An improved heuristic algorithm for finding motif signals in DNA sequences," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 4, pp. 959–975, 2011.

[29] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

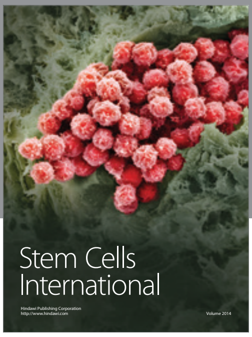
[30] M. Leone, Sumedha, and M. Weigt, "Clustering by soft-constraint affinity propagation: applications to gene-expression data," *Bioinformatics*, vol. 23, no. 20, pp. 2708–2715, 2007.

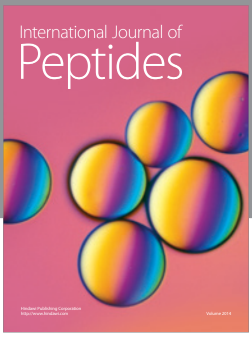
[31] D. Wang and N. K. Lee, "Computational discovery of motifs using hierarchical clustering techniques," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pp. 1073–1078, December 2008.

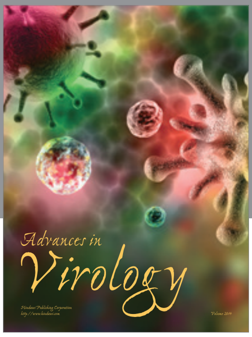
[32] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner, "WebLogo: a sequence logo generator," *Genome Research*, vol. 14, no. 6, pp. 1188–1190, 2004.

[33] M. Tompa, N. Li, T. L. Bailey et al., "Assessing computational tools for the discovery of transcription factor binding sites," *Nature Biotechnology*, vol. 23, no. 1, pp. 137–144, 2005.

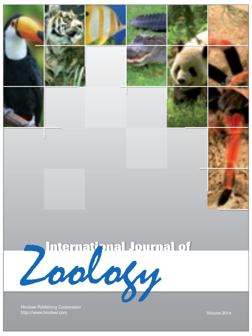
BioMed
Research International

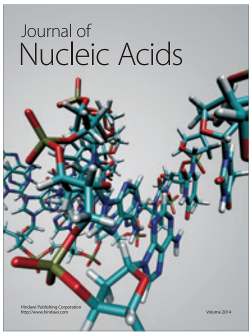

Stem Cells
International


International Journal of
Peptides


Advances in
Virology

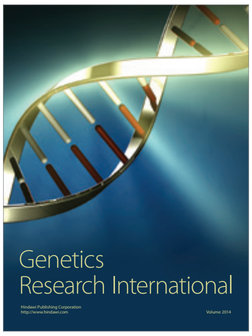
International Journal of
Genomics

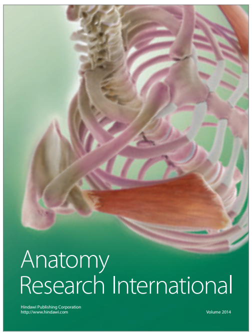

International Journal of
Zoology


Journal of
Nucleic Acids

Journal of
Signal Transduction

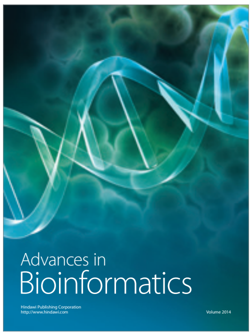
The Scientific
World Journal

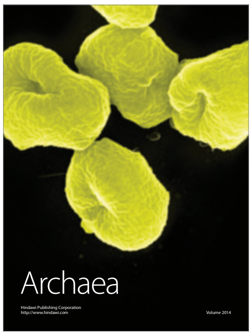

Genetics
Research International

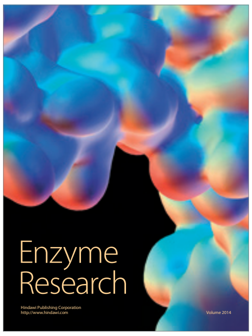

Anatomy
Research International

International Journal of
Microbiology

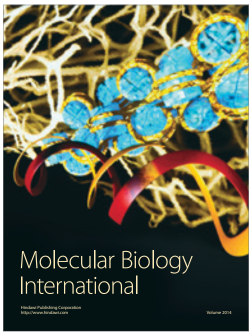
Biochemistry
Research International


Advances in
Bioinformatics


Archaea


Enzyme
Research

International Journal of
Evolutionary Biology


Molecular Biology
International

Journal of
Marine Biology