A Robust Elicitation Algorithm for Discovering DNA Motifs Using Fuzzy Self-Organizing Maps

Dianhui Wang, *Senior Member, IEEE*, and Sarwar Tapan

***Abstract*— It is important to identify DNA motifs in promoter regions to understand the mechanism of gene regulation. Computational approaches for finding DNA motifs are well recognized as useful tools to biologists, which greatly help in saving experimental time and cost in wet laboratories. Self-organizing maps (SOMs), as a powerful clustering tool, have demonstrated good potential for problem solving. However, the current SOM-based motif discovery algorithms unfairly treat data samples lying around the cluster boundaries by assigning them to one of the nodes, which may result in unreliable system performance. This paper aims to develop a robust framework for discovering DNA motifs, where fuzzy SOMs, with an integration of fuzzy *c*-means membership functions and a standard batch-learning scheme, are employed to extract putative motifs with varying length in a recursive manner. Experimental results on eight real datasets show that our proposed algorithm outperforms the other searching tools such as SOMBRERO, SOMEA, MEME, AlignACE, and WEEDER in terms of the *F*-measure and algorithm reliability. It is observed that a remarkable 24.6% improvement can be achieved compared to the state-of-the-art SOMBRERO. Furthermore, our algorithm can produce a 20% and 6.6% improvement over SOMBRERO and SOMEA, respectively, in finding multiple motifs on five artificial datasets.**

***Index Terms*— Computational motif discovery, DNA sequences, fuzzy self-organizing maps, robust elicitation algorithm.**

I. Introduction

Transcription factor binding sites (TFBSs) are small DNA segments (usually ≤ 30 bp in length) that interact with the transcription proteins known as the transcription factors (TFs) to initiate gene transcription, which is the first step of gene expression [1]. A collection of binding sites that bind to a specific TF from a set of co-expressed genes is termed as a motif and characterizes the binding preferences of that TF. Discovering novel motifs in co-expressed genes or finding new binding sites associated with known TFs is crucial in understanding gene regulatory mechanisms [2].

Experimental approaches for finding DNA motifs, e.g., ChIP-chip [3], ChIP-seq, and micro-array technology [4], are still laborious, time consuming, and expensive. Hence, computational approaches have received considerable attention in the last two decades [5]. Over these years, many motif search algorithms and Web-based tools have been developed based on computational intelligence systems and data mining techniques, e.g., [6]–[19]. However, according to some surveys [20]–[22], the existing tools are still unable to achieve satisfactory performance in terms of accuracy, reliability, and scalability. Thus, effective motif discovery still remains challenging despite the enormous number of attempts over the past years, which necessitates the exploration of possibilities for improved developments.

Being one of the powerful data mining tools, clustering techniques can be used to group unlabeled data for subtle pattern recognition in biological sequences [23]–[25]. The basic idea behind clustering-based DNA motif discovery algorithms is to extract a set of clusters, which are composed of a similar type of same length (k-length) subsequences (known as k-mers), followed by merging and optimizing some potential clusters with a considerable degree of presence of the functional motif properties [26]. The clustering process iteratively updates the centroids with preferably random initialization to become useful representations of the potential clusters with motif properties. Clustering algorithms are capable of finding motifs embedded in DNA sequences because of some inherent properties of motifs, such as the positional reservation, rareness in respect to the background, and overrepresentation nature [27]. Also, the use of domain-specific scoring functions for data assignment plays a key role in clustering-based motif extraction exercises [24]. Note that a generic clustering approach for DNA motif discovery may not be effective unless it is equipped with specific mechanisms for forming clusters of similar k-mers from the input sequences, ensuring some degree of presence of the functional motif characteristics in the clusters. The appropriateness of the similarity metrics and the type of centroids are the two key factors in clustering-based motif discovery.

Self-organizing maps (SOMs) [28] can be employed for motif discovery through clustering similar k-mers extracted from DNA sequences [13], [14]. Its competitive learning with neighborhood-based cooperation between nodes introduces a useful generality during the initial learning phase, which distinguishes SOMs from a typical winner-takes-all type k-means clustering algorithm in terms of significantly improved DNA motif discovery results [13]. Furthermore, the use of SOMs enables a useful refinement of putative clusters (nodes) with the contribution from the grid-based immediate neighboring clusters that often preserve considerably similar or partially similar motif patterns. Although the conditions for keeping the topology preservation in SOMs are rarely examined in practice [29], the applicability of SOM-based clustering techniques for DNA motif discovery is undoubted.

SOMs perform hard partitions in the k-mer dataset, and each node in post-training SOMs encodes a nonoverlapping

Manuscript received September 8, 2012; revised April 24, 2013; accepted July 26, 2013. Date of publication August 15, 2013; date of current version September 27, 2013.

The authors are with the Department of Computer Science and Computer Engineering, La Trobe University, Melbourne 3086, Australia (e-mail: dh.wang@latrobe.edu.au; mstapan@students.latrobe.edu.au).

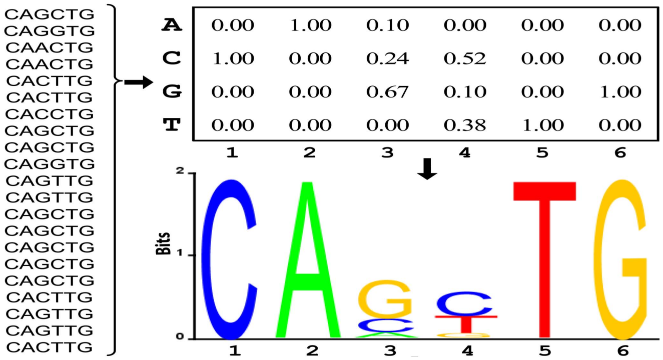
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2275733

area in the input distribution. Despite its proven usefulness in a wide range of applications, it has limited suitability for the DNA motif discovery task, since the hard partitions push the data samples lying around the cluster boundaries to belong only to the best matching cluster even though the matching is insignificant. This can have a considerable negative impact on the DNA motif discovery performance, especially in finding weak motifs. Consequently, it is inevitable and usually unavoidable to have some true but weak binding sites distributed among the neighboring clusters of a true motif, because of the fact that the functional binding sites can often be considerably degenerated in functional motifs. This implies that a motif extracted from a crisp partitioning approach often fails to attach weak binding sites, which eventually degrades the performance and reliability of the SOM-based motif finding tools.

To overcome the problem mentioned above, this paper presents a fuzzy SOMs (FSOMs) based approach for finding motifs, where the FSOM performs a fuzzy c-means (FCMs) type soft clustering [30]. Here, each cluster in the FSOMs has a global relationship to all k-mers in the dataset, which increases the chances of attaching degenerated instances to the functional motifs. Motivated by an incremental learning algorithm of fuzzy SOMs proposed in [31], a batch learning algorithm for training FSOM networks is developed in this paper. It should be pointed out that such an extension is necessary and meaningful for motif discovery. Indeed, the necessity comes from the nature of modeling motifs, i.e., each node as a whole stores a motif candidate. The incremental learning schemes, in this case, fail to comply with the requirements of modeling motifs since a single k-mer is incapable of characterizing the properties of functional motifs. Also, such type of learning schemes slow down the processing of finding motifs. In this paper, a robust framework, named robust elicitation algorithms for discovering (READ) DNA motifs, is proposed by utilizing multiple FSOMs and some innovative heuristics for motif candidate optimization.

An important issue related to SOM-based motif discovery tools is about the robustness of performance with respect to the map size. An SOM network with an improper number of nodes may produce inappropriate partitions of the dataset and degrade the motif-like clusters and eventually degrade the motif finding performance. Being data-specific in nature, setting a proper map size in SOMs is mostly an empirical task that needs considerable human intervention and careful attempts. To minimize the impact of an improper map size setting on the motif mining performance, two complementary strategies are adopted in the READ framework. First, FSOMs are employed, which naturally have a better tolerance in handling an improper map size setting due to its soft-partitioning mechanism. Second, the cluster quality degradation due to an inappropriate map size setting is compensated by an improved postprocessing scheme that aims to regain the lost information on the degraded motif-like clusters. As observed in our experiments, the proposed READ framework performs favorably and robustly compared to the FCM-based and crisp SOM-based algorithms.



	1	2	3	4	5	6
A	0.00	1.00	0.10	0.00	0.00	0.00
C	1.00	0.00	0.24	0.52	0.00	0.00
G	0.00	0.00	0.67	0.10	0.00	1.00
T	0.00	0.00	0.00	0.38	1.00	0.00

Fig. 1. Collection of binding sites (6 bp length) of MyoD TF [9] represented as a motif PFM and visualized using an entropy-based logo.

The remainder of this paper is organized as follows: Section II gives some preliminaries including motif modeling and evaluation, which will be used in the sequential section for model selection and rankin. Section III describes the proposed batch learning algorithm for the FSOMs. Section IV details our proposed READ framework with an improved refinement scheme. Section V reports the experimental results on eight real and five artificial datasets, including some comparisons with five representative tools and a robustness analysis. Section VI concludes this paper with remarks on further studies.

II. Motif Modeling

In computational practice, a DNA motif can be modeled by a set of k-mers extracted from a collection of promoter sequences of a set of co-regulated genes, such that an alignment of the subsequences gives a significant positional conservation of the nucleotides, which is usually expressed by using a position frequency matrix (PFM) or other variants of it. A motif can be visualized using a logo [32], as shown in Fig. 1, representing the entropy of each column in the PFM.

The following notation is introduced and used in this paper. Let $D = \{s_1, s_2, \ldots, s_n\}$ be a dataset with n sequences that are believed to contain regulatory motifs. The PFM-based motif model [33], denoted by M, is described as a matrix, i.e., $M = [f(b_i, i)]_{4\times k}$, where $b_i \in \chi = \{A, C, G, T\}$, and $f(b_i, i)$ represents a relative frequency of a nucleotide b_i appearing at the ith position. Similarly, a k-mer, $K_s = q_1 q_2 \ldots q_k$, can be encoded as a binary matrix $K = [k(b_i, i)]_{4\times k}$ with $k(q_i, i) = 1$ and $k(b_i, i) = 0$ for $b_i \neq q_i$. For example, the k-mer $K_s = AGCGTGT$ is encoded as

$$K = e(K_s) = \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,1\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,1 \end{bmatrix}_{4\times k}.$$

A. Motif Modeling Using MISCORE

Functional DNA motifs are typically characterized by model conservation, background rareness, and compositional complexity. Because of their role in gene regulation, motifs are mostly evolutionarily conserved. Thus, motif instances (binding sites) appear to be rather similar to each other despite having variability in their nucleotide composition [34]. Also, motifs are subtle signals in DNA sequences which are rarely found in the sequence backgrounds (i.e., background rareness). Furthermore, a functional motif usually demonstrates a complex composition of nucleotides known as motif complexity [13].

Mismatch-based matrix similarity score (MISCORE) [35] is a new and effective scheme for measuring the similarity between a k-mer and a motif model that uses a combined characterization of functional motif features without any assumption on nucleotide dependency [36]. Given a PFM model M and a set of background sequences, MISCORE can be computed by

$$r(K, M) = \frac{d(K, M)}{d(K, M_{\text{ref}}) + c(K)} \tag{1}$$

where M_{ref} represents a background reference model, $c(K)$ is a compositional complexity of the K, which is given by

$$c(K) = \frac{4}{3}\left[1 - \frac{1}{k^2}\sum_{\forall b_i \in \chi}\left(\sum_{i=1}^{k} k(b_i, i)\right)^2\right] \tag{2}$$

and $d(K, M)$ is a generalized Hamming distance between the K and the model M, which can be defined as,

$$d(K, M) = 1 - \frac{1}{k}\sum_{i=1}^{k}\sum_{\forall b_i \in \chi} f(b_i, i)k(b_i, i). \tag{3}$$

Using MISCORE, a metric for quantifying the potentiality of a candidate motif to be a functional one can be defined [35]. Given a set of k-mers S, its PFM model can be computed as $M_S = 1/|S| \sum_{\forall K \in S} e(K)$, and the MISCORE-based motif score (MMS), denoted by $R(S)$, is defined as

$$R(S) = \frac{1}{|S|}\sum_{\forall K \in S} r(K, M_S) \tag{4}$$

where $| * |$ is the set cardinality.

Binding sites are evolutionarily constrained with limited mutations. Hence, a K may be a putative motif instance if $r(K, M)$ takes a smaller value for a given motif reference M. In practice, this implies a smaller mismatch to a putative model M_S than the background reference model M_{ref} constructed by all k-mers from the background sequences. Note that the model M_{ref} can be constructed by computing the background frequency of bases in each column. Large sequence portions with minimal chances of having true binding sites can be used as backgrounds, e.g., random chunks of large genomic portions or a large collection of upstream regions from relevant species.

Usually, a motif model constructed by a set of k-mers with more repetitive nucleotides cannot be functional [13]. Unfortunately, computational tools often return and highly rank such model candidates. In order to improve the discriminative power of our MMS metric on separating the functional motifs from the random ones, the complexity quantifier $c(K)$ is integrated in (1), which helps in automatically eliminating low-complex motif models from a top ranked list. Therefore, a complexity-based filtering of the candidates, as applied in [13] and [14], can be avoided in the READ system.

B. Localized MMS for Model Assessment

Transcription proteins rarely contact a single nucleotide without interacting with the adjacent bases in the binding process. Hence, the positions with a higher binding energy given by IC (and also a lower binding energy) are usually clustered as local information blocks in the PFM of functional motifs [38]. Position-specific similarity metrics assign equal weights to every position in the model and ignore the variability among the local blocks of a functional motif PFM. Since a motif PFM can be regarded as a descriptor of its binding preferences, the underlying nucleotide blocks are believed to carry some useful information that constitutes the overall characterization on motifs. Therefore, a motif PFM should be decomposed into a set of local blocks with meaningful weights according to its merit of being functional.

MISCORE is then extended as localized-MISCORE with denotation $r_l(K, M_S)$

$$r_l(K, M_S) = \sum_{j=1}^{k-w+1} g_j\left(\frac{d\left(\beta_j(K), \beta_j(M_S)\right)}{d\left(\beta_j(K), \beta_j(M_{\text{ref}})\right)}\right) \tag{5}$$

where $\beta_j(K)$, $\beta_j(M_S)$, and $\beta_j(M_{\text{ref}})$ denote the jth local block in the K, the M_S, and the reference model M_{ref}, respectively. A w-length local block $\beta_j(\cdot)$ can be produced by shifting a small matrix window $\beta_{[4\times w]}$ $(2 \le w < k)$ in K, in M_S, and in M_{ref} so that $k - w + 1$ number of blocks can be produced.

The weight g_j for the jth block in M_S (i.e., $\beta_j(M_S)$) can be assigned as

$$g_j = \frac{G(\beta_j(M_S))}{\sum_{q=1}^{k-w+1} G(\beta_q(M_S))} \tag{6}$$

where $G(\beta_j(M_S))$ is a modified Gini purity index (a complement of the Gini impurity index), which can be written as

$$G(\beta_j(M_S)) = \frac{1}{w}\sum_{i=j}^{j+w-1}\sum_{\forall b_i \in \chi}\left(\frac{f(b_i, i)}{p(b_i)}\right)^2 \tag{7}$$

where $p(b_i)$ is the background frequency of base b_i.

Then, a localized-MMS (denoted by $R_l(S)$) can be defined and employed to evaluate a candidate motif

$$R_l(S) = \frac{1}{|S|}\sum_{\forall K \in S} r_l(K, M_S). \tag{8}$$

In our proposed framework, it is employed to assess and rank the motif candidates extracted from FSOMs. Indeed, such a localized version aims at improving the system performance on weak motifs. From the experimental results reported in our previous work [35], the localized-MMS performs better in distinguishing weak motifs, but demonstrates equivalent recognition capability for stronger motifs.

III. Fuzzy Self-Organizing Maps

FSOMs can be regarded as an improved version of the classical SOMs, where it allows data samples to belong to the nodes with a distributed degree of membership [30], [39]–[42]. This section extends the work reported in [31] and presents a batch learning algorithm for the FSOMs. Note that such an

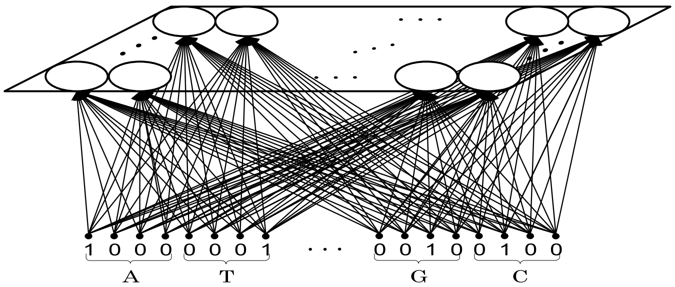


Fig. 2. Binary encoded k-mer $= `AT\dots GC'$ presented to the FSOM.

extension is necessary and meaningful to the implementation of our READ system for motif discovery. Fig. 2 depicts the used FSOM with an encoded k-mer input.

A. SOM Batch Learning

Let Γ be a set of n-dimensional data samples, where the jth sample is denoted as $x_j = \left[x_{j1}, x_{j2}, \dots, x_{\mathrm{jn}}\right]$; N is the number of nodes of a SOM network, where the ith node has a low-dimensional (e.g., 2-D) grid location as $v_i = [v_{i1}, v_{i2}]$ and an n-dimensional prototype vector as $w_i = [w_{i1}, w_{i2}, \dots, w_{\mathrm{in}}]$. In our task of DNA motif discovery, the dataset Γ is a set of k-mers extracted from the input sequences and the ith node in the SOM network stores a node-PFM (M_i). Using these notations, a standard batch-learning algorithm [28] of the SOM network can be described as follows:

$$M_i(t+1) = \frac{\sum_{j=1}^{|\Gamma|} h_{ic_j}(t) K_j}{\sum_{j=1}^{|\Gamma|} h_{ic_j}(t)} \tag{9}$$

where c_j is the best matching node for K_j at iteration t

$$c_j = \arg\min_q \{r(K_j, M_q(t))\} \tag{10}$$

where $r\left(K_j, M_q(t)\right)$ is the MISCORE that gives the similarity between K_j and $M_q(t)$, and $h_{ic_j}(t)$ is the neighborhood function expressed with a shrinking neighborhood range $\sigma(t)$ as

$$h_{ic_j}(t) = \exp\left\{ -\frac{\|\mathrm{v_i} - \mathrm{v_{c_j}}\|^2}{2\sigma(\mathrm{t})^2} \right\} \tag{11}$$

where $\| * \|$ is the Euclidean distance between the coordinates of the two nodes in the 2-D grid.

B. Fuzzy SOM Batch Learning

The batch-learning rule given in (9) can be rewritten as follows with singleton membership functions

$$M_i(t+1) = \frac{\sum_{j=1}^{|\Gamma|} \sum_{k=1}^{N} U_{kj}(t) h_{\mathrm{ik}}(t) K_j}{\sum_{j=1}^{|\Gamma|} \sum_{k=1}^{N} U_{kj}(t) h_{\mathrm{ik}}(t)} \tag{12}$$

where the U is a binary membership matrix, representing the belonging relationship between K_j and $M_k(t)$, which can be defined by

$$U_{kj}(t) = \begin{cases} 1, \; k = \arg\min_q \{r\left(K_j, M_q(t)\right)\}, \\ 0, \; \text{otherwise.} \end{cases} \tag{13}$$

The crisp membership matrix U in (12) can be replaced by a fuzzy membership matrix μ as in FCM algorithms [30], where $\mu_{kj}(t)$ represents the degree of membership of K_j with respect to the model $M_k(t)$

$$\mu_{kj}(t+1) = \left[\sum_{l=1}^{N} \left(\frac{r\left(K_j, M_k(t)\right)}{r\left(K_j, M_l(t)\right)} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{14}$$

where m is the fuzziness regulator.

In this paper, we propose a generalized batch-learning algorithm for SOM networks with fuzzy membership involved in the learning process. The SOM networks with such learning techniques are termed as fuzzy SOM networks. Concretely, from (12), the node-PFM (prototype) M_i can be updated by using the following fuzzy SOM batch learning algorithm:

$$M_i(t+1) = \frac{\sum_{j=1}^{|\Gamma|} \sum_{k=1}^{N} \mu_{kj}^m(t)\, h_{\mathrm{ik}}(t) K_j}{\sum_{j=1}^{|\Gamma|} \sum_{k=1}^{N} \mu_{kj}^m(t)\, h_{\mathrm{ik}}(t)} \tag{15}$$

where $h_{\mathrm{ik}}(t)$ is given by a neighborhood function

$$h_{\mathrm{ik}}(t) = \exp\left\{ -\frac{\|\mathrm{z_i} - \mathrm{z_k}\|^2}{2\sigma(\mathrm{t})^2} \right\}. \tag{16}$$

The neighborhood range $\sigma(t)$ can be monotonically shrunken using the following criterion mentioned in [43]:

$$\sigma(t+1) = \sigma(t_0)\exp\left\{ -2\sigma(\mathrm{t_0}) \frac{\mathrm{t}}{\mathrm{t_{max}}} \right\} \tag{17}$$

where $\sigma(t_0)$ is the fairly large initial σ and $t_{\max}$ is the maximum epoch set by the user.

For implementation convenience, the ith node is associated with two computing components, namely $\Delta M_i(t)_{[4\times k]}$ and $\Delta h_i(t)$, for tracing the contribution of each K_j to the updates of each $M_i(t)$. Note that the entries in $\Delta M_i(t)_{[4\times k]}$ and $\Delta h_i(t)$ are initialized to zeros at the beginning of each epoch. The cumulative updates for ith node, $1 \le i \le N$, can be then traced by

$$\Delta M_i(t) \Leftarrow \Delta M_i(t) + \sum_{l=1}^{N} \mu_{lj}^m(t)\, h_{il}(t)\, K_j$$
$$\Delta h_i(t) \Leftarrow \Delta h_i(t) + \sum_{l=1}^{N} \mu_{lj}^m(t)\, h_{il}(t). \tag{18}$$

Finally, the cumulative updates can be assigned to $M_i(t+1)$ at the end of tth training epoch as

$$M_i(t+1) = \frac{\Delta M_i(t)}{\Delta h_i(t)}. \tag{19}$$

Note that (15) is a neighborhood-incorporated expression of the classical prototype updating rule in FCM [30]. Hence, after eliminating the effect of the neighborhood function $h_{\mathrm{ik}}(t)$ from (15), it reads the same as the FCM prototype updating rule [30]

$$M_i(t+1) = \frac{\sum_{j=1}^{|\Gamma|} \mu_{\mathrm{ij}}^m(t) K_j}{\sum_{j=1}^{|\Gamma|} \mu_{\mathrm{ij}}^m(t)} \tag{20}$$

where the exponential term $m > 1$ essentially controls the amount of fuzziness in μ. As $m \to \infty$, $\mu_{jk}(t) \to 1/N$ and a meaningful fuzzy partitioning requires $m \to 1$ [40].

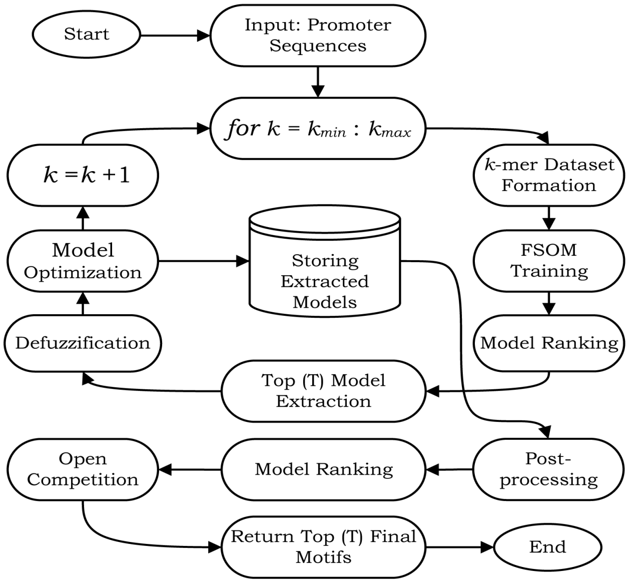


Fig. 3. READ framework overview. Multiple FSOMs are trained for variable k-mer lengths ($k_{\min} \leq k \leq k_{\max}$) and top T candidates from multiple maps are returned as the final motifs after being postprocessed.

IV. Robust Motif Discovery Framework

Our READ framework performs multiple FSOM training for $k_{\min} \leq k \leq k_{\max}$. The FSOMs are trained by all k-mers extracted from a given dataset, and the top-ranked T candidate motifs are generated as initial models. Then, a merging process is used to retrieve some possibly missing motif instances. Next, an improved postprocessing is applied to refine the candidate motif models. Finally, the top-ranked T candidate motifs are returned as the final motifs through open competition among the candidate motifs with variable lengths. An overview of the READ framework is presented in Fig. 3, which consists of three major components: (C1) system training; (C2) motif extraction; and (C3) post-processing.

A. System Training

System training includes three tasks: 1) initialization; 2) learning; and 3) termination.

1) *Initialization:* system initialization needs the map size (i.e., $N =$ number of nodes) and the k-mer dataset Γ to be known. The ith node in the FSOM network is then initialized with a random PFM model, M_i.
2) *Fuzzy SOM learning:*
   a) *Membership computation:* The fuzzy membership for all $K \in \Gamma$ to each node-PFM $M_i(t)$ is computed using (14).
   b) *Prototype updating:* Each node-PFM $M_i(t)$ is updated using (15). In implementation, all K can be fed to the network first. Then, the cumulative updates for each $M_i(t)$ can be traced and finally assigned to $M_i(t+1)$ at the end of each training epoch.
3) *Termination:* The learning process will be terminated as the maximum number (default setting is 100 in READ) of epochs is reached. However, early termination can be applied using the following criteria. After the neighborhood range σ is sufficiently shrunken, the FSOM becomes FCM-type clustering. So the standard FCM objective function [30] can be employed as an indicator

$$J_m(t) = \sum_{j=1}^{|\Gamma|} \sum_{i=1}^{N} r\left(K_j, M_i(t)\right) \mu_{\text{ij}}^m(t). \tag{21}$$

If $J_m(t)$ moves steadily while σ is reasonably shrunken, the updating process can be stopped. Alternatively, the goodness of the fuzzy partitioning can be monitored in order to detect an early termination using a partitioning coefficient $pc\left(\mu(t)\right)$

$$pc(\mu(t)) = \frac{1}{|\Gamma|} \sum_{j=1}^{\Gamma} \sum_{i=1}^{N} \mu_{\text{ij}}^2(t). \tag{22}$$

A value close to 1 is necessary for termination since it refers to an average minimal fuzziness in $\mu(t)$.

B. Motif Extraction

Now, the post-training M_i becomes a fuzzy-PFM of the ith node, where all k-mers belong to it with certain degrees. In order to rank these fuzzy-PFM models, we generalize (4) and define the following metric to evaluate the fuzzy motif models with membership information:

$$R_f(M_i) = \frac{\sum_{j=1}^{|\Gamma|} r(K_j, M_i)\mu_{\text{ij}}^m}{\sum_{j=1}^{|\Gamma|} \mu_{\text{ij}}^m} \tag{23}$$

where M_i, Γ, K, μ, m and $r(K_j, M_i)$ are as defined earlier.

A smaller $R_f(M_i)$ score indicates a better potential for the M_i to be a putative motif. The candidate motifs (i.e., the prototypes) are then ranked according to their R_f scores and the top-ranked T candidates will be selected for further data processing. Each selected fuzzy candidate is then converted to a crisp motif model through a defuzzification operation, i.e., selecting the Ks with the highest membership values to retain in the model.

It is observed that the immediate neighboring models (nodes) of the top-ranked model are often found to share considerably similar motif patterns. This property offers us a way to find weak motif signals from the k-mers contained in the neighboring models through a heuristic optimization process. In this paper, we propose a method to select a k-mer from each immediate neighboring node (denoted by S_{ngh}) of a top-ranked candidate model S_i. The following steps give the way to define the k-mers to be merged into the S_i.

1) For each $K \in S_{\text{ngh}}$, we first define a small region $\Theta(K) = [post(K) - k, post(K) + k]$ in its sequence. If $S_i \cap \Theta(K) = \emptyset$, then a possible k-mer (denoted as K^*) from the $\Theta(K)$ will be picked out to be merged to the candidate model S_i according to the criteria in 2) below. In the case that the two criteria cannot be satisfied simultaneously, no action will be applied to S_i.
2) a) $K^* = \arg\min_j \{r(X_j, M_i)\}$ for all $X_j \in \Theta(K)$ and
   b) $R(S_i \cup K^*) \leq R(S_i)$.

This scheme ensures that only a putative K from a new location is added to the candidate motif S_i with the constraint that an amendment does not degrade its motif-like quality quantified by $R(S_i)$ given in (4). The best K among a set of k-mers from the new sequence region is added to the candidate motif S_i in order to ensure a better alignment between the new K and S_i. Finding a new region in the sequences increases the chances of finding a possibly missed instance of the motif candidate S_i and avoids duplication of instances in the candidate, since multiple k-mers pointing to the same small region in the sequence brings no practical benefits to the discovery results. It has been practically observed that this optimization of a candidate is capable of attaching a small number of new instances to the candidate, as anticipated. Also, it enables a k-mer to belong to multiple candidate motifs, which is useful in distributing degenerated motif instances to multiple candidate clusters that share the complete or partial motif pattern of the functional motif. In a typical clustering approach, e.g., c-means or fuzzy-c-means, this heuristic refinement is complicated because of their topologically unordered representation of the clusters.

C. Postprocessing

Postprocessing in READ aims to refine the extracted candidates to improve system robustness and motif discovery performance. First, the scope of a target motif needs to be set, i.e., the maximum (minimum) $P_{\max}(P_{\min})$ number of predicted motif instances per promoter sequence. Each extracted candidate S_i is then processed, i.e., adding new putative Ks or removing some Ks (located in the same sequence) from the model in order to meet the user defined scope using Algorithm 1.

Second, by applying Algorithm 2, some weak motif instances may be retrieved (one at a time) from the k-mer dataset through adding or replacing the existing predictions from each promoter sequence in S_i. Simultaneously, it removes noise-like predictions (one at a time) from S_i under the user-defined model scope provided that each amendment can increase the motif qualities of the candidate S_i, quantified by the localized-MMS given in (8).

This postprocessing scheme is routinely fast and capable of quickly turning a suboptimal guess of a motif into a target one. Hence, this refinement scheme can be employed for the effective recovery of the noisy candidate motifs that are produced by an improper map size setting or an improper training of the FSOMs. This postprocessing scheme was initially proposed and reported as the most-one-in-out (MOIO) scheme in [10], which has been significantly improved in this paper.

Algorithm 1: Motif Scope Per Promoter Sequence

```
input: $S_i$, $P_{min,\max}$
ensure: $P_{\min} \leq P_{\max}$
for each promoter sequence $seq_j$ do
  Find set $S_k = \{K | K \in S_i \cap seq_j\}$
  if $|S_k| < P_{\min}$ then
    Add $(P_{\min} - |S_k|)$ number of the lowest scoring $K$s such that $(K \in seq_j \setminus S_i)$ using $r_l(K, M_i)$ as given in (5).
  end if6
  if $|S_k| > P_{\max}$ then
    Remove $(|S_k| - P_{\max})$ number of the highest scoring $K$s such that $(K \in S_i \cap seq_j)$ from $S_i$ using $r_l(K, M_i)$ as scoring function.
  end if
end for
return $S_i$.
```

Algorithm 2: Motif Refinement

```
input: $S_i$, $P_{min,\max}$ , $\Gamma$
call: Algorithm 1 for motif scoping.
set: $\Delta \Longleftarrow 0.1$
while $\Delta > 0.0$ do
  Task: weak binding site acquisition
  1. Find a best $K_t = \arg\min_p\{r_l(K_p, M_i), \forall K_p \in \Gamma \setminus S_i\}$.
  2. Get $seq_j$ as the promoter sequence where $K_t$ is located.
  3. Find $S_k = \{K | \forall K \in S_i \cap seq_j\}$.
  4. $S_{temp} \Longleftarrow S_i \setminus S_k$
  5. $S_k \Longleftarrow S_k \cup \{K_t\}$
  6. if $|S_k| > P_{\max}$ then
       6.1 Compute $r_l(K, M_i), \forall K \in S_k$.
       6.2 $S_k$ only keeps the $P_{\max}$ number of lowest scoring $K$s.
     end if
  7. $S_{temp} \Longleftarrow S_{temp} \cup S_k$
  8. $\Delta \Longleftarrow R_l(S_i) - R_l(S_{temp})$
  9. if $\Delta \geq 0$ then, $S_i \Longleftarrow S_{temp}$, end if
  Task: noise elimination
  1. Find an worst $K_t = \arg\max_p\{r_l(K_p, M_i), \forall K_p \in S_i\}$.
  2. Get $seq_j$ as the promoter sequence where $K_t$ is located.
  3. Find $S_k = \{K | K \in S_i \cap seq_j\}$
  4. if $|S_k| > P_{\min}$, then $S_{temp} \Longleftarrow S_i \setminus \{K_t\}$, end if
  5. $\Delta \Longleftarrow R_l(S_i) - R_l(S_{temp})$
  6. if $\Delta \geq 0$ then, $S_i \Longleftarrow S_{temp}$, end if
end while
return $S_i$.
```

Notations: S_i: motif to refine; M_i: motif-PFM; Γ: k-mer dataset; $P_{\min}(P_{\max})$: minimum (maximum) number of predicted sites per sequence; '$\Longleftarrow$': assignment by erasing previous contents; S_k: a k-mer set; $| * |$: set cardinality; K_t: a temporary k-mer; and '\': set minus notation.

1) Remarks: In computational exercises for DNA motif discovery, the signal-to-noise ratio is rather low, especially for large datasets. Therefore, mining functional motifs seems like looking for a needle in a haystack. Technically speaking, it is possible to obtain improved results if an effective filtering system can be placed before running READ. However, such preprocessing in DNA motif discovery must ensure no dismissals of any true binding sites, which is quite difficult to achieve. It should be pointed out that the filtering must be done piecewise in sequences rather than at the k-mer level; otherwise, many existing tools stop working. Although it is a meaningful job to improve the computational environment in finding motifs, there is no report available on it to date.

Another important issue related to our algorithm development is the use of *a priori* knowledge on motifs. Suppose that a reference motif model (PFM or PWM) is available, which may come from some relevant species or a public database. Then, it will be beneficial to use it as a supervised signal in the learning process. With the assistance of *a priori* knowledge, a hierarchical READ can be further developed.

V. Performance Evaluation

This section contains some results and comparisons on eight benchmarked real single motif datasets and five artificial datasets for evaluating the performance of our proposed algorithm. To see the merits of the framework, the existing SOM-based tools, e.g., SOMBRERO [13] and SOMEA [14], and other prominent tools, e.g., MEME [8], AlignACE [17], and WEEDER [18], are used in the comparison. In this section, a robustness analysis on the system performance with respect

TABLE I

POSITIONAL OVERLAP BETWEEN TRUE AND PREDICTED BINDING SITES

	Sequence	(Predicted sites, pos)	True and predicted binding sites
>>	AF188709	(ATGACGTC, 109)	...ggccaaagtaaagccctctttctca**aTGACGTC**Aagatctttaccaagattaggctttca...
>>	M26065	(ATGACATC, 028)	...tccaactctgaaaattcctgtGATTCG**ATGACATC**AGTACGGTGaataatcaaactggcg...
>>	X13257	(ATGACGTC, 020)	...tctacttcaactcccactg**atgacgtc**catGTGTCATTAGTGCCAATTAGAGgagggcag...
>>	X56849	(GTGATGTC, 095)	...ttaagctctgtgagaatcctgggagttg**gTGATGTC**Agactagttgggtcatttgaaggt...
>>	X56850	(GTGATGTC, 124)	...cgctgtgagaatcctgggagttg**gTGATGTC**Agactggttgggtcatttgaaggttagca...
>>	K03021	(ATGACATC, 087)	...ttccaaattcctgcgattcA**ATGACATC**ACGGCtgtgaataatcagcctggcccgaagcc...
>>	M26179	(GTGACGTC, 272)	...cagcatctcttttgttcgctgcgaacccacagtccccc**gTGACGTC**Acccggagcccggg...
>>	M31184	(ATGACGTC, 016)	...aattcctgtGATTCG**ATGACGTC**AGTACGGTGaataatcaagctggcgtcaagccaagag...
>>	S99616	(CTGACGAC, 002)	...g**cTGACGac**caaggagatcttcccacagacccagcaccagggaaatggtccggaaattgc...
>>	X04724	(TTGACGTC, 022)	...ccctaggactaagtagaggtg**tTGACGtc**caatgagcgctttctgcagacctagcaccag...

Note: Positional overlap between predicted sites (bold faced) and the true binding sites (upper case) are shown using a selected portion of a set of promoter sequences of CREB [9] TF. Sequence $X13257$ (3^{rd} from top) produced a zero TP count as no overlap between the true and the predicted sites are found. The header 'pos' indicates the position of the predicted site from the first nucleotide in the sequence in forward direction.

to the map size variation is also given, which is related to the algorithm reliability.

The well-known precision (P), recall (R), and F-measure (F) scores are employed as the metrics for performance evaluation in this paper, i.e., $P = TP/(TP + FP)$, $R = TP/(TP + FN)$, and $F = 2PR/(P + R)$, where TP, FP, and FN are the true-positive, false-positive, and false-negative predictions, respectively. TP refers to the number of true binding sites that are overlapped by at least one predicted binding sites with at least x nucleotides such as, $x = 2$ if $L_{\text{tbs}} \leq 6$ and $x = 4$ if $L_{\text{tbs}} > 6$, where L_{tbs} represents the average length of the true binding sites of the known motif; FP is the number of predicted binding sites that do not sufficiently overlap with any true binding site to meet the TP definition; and FN is the number of true binding sites that are not sufficiently overlapped by any predicted site according to the TP definition.

A demonstration on the overlap between the true and predicted binding sites is presented in Table I.

A. Comparison Using Real DNA Datasets

These eight datasets are composed of a set of promoters of co-regulated genes that contain known motifs associated with the following TFs: ERE, MEF2, SRF, CREB, E2F, MyoD, CRP, and GCN4. These datasets have been collected from [9] and [44]. The locations of all binding sites in these datasets are known.

READ was run with a random map size between 10×10 to 20×20 on each dataset. For each run, the expected motif width was set as $(k_{\min}, k_{\max}) = (l - 3, l + 3)$, where l was the consensus length of the known motif in the dataset. It was set to return the top 10 candidates during each run. The initial neighborhood range $\sigma(t_0) = 3$ and the fuzziness regulator $m = 1.025$ were set. Maximum epoch $t_{\max} = 100$ was set; however, the following early termination criteria was used, i.e., $J_m(t-1) - J_m(t) \leq 0.0001$ or $pc(\mu(t)) \geq 0.95$ when $\sigma(t) \leq 0.05$. This is because the neighborhood cooperation rate for the nearest neighboring nodes then becomes $h = \exp(-1/(2 * 0.05^2)) = 1.3839\text{e-}087$, indicating considerably less cooperation in the final stage of training. In this paper, the motif scopes were set as $(P_{\min}, P_{\max}) = (1, 3)$.

SOMEA was run with a fixed learning rate and the neighborhood range set at 0.005 and 3.0, respectively [14]. The stand-alone SOMBRERO tool was then run on each dataset, where the complexity threshold was empirically selected as 0.1. For a fair comparison between the SOM-based and FSOM-based tools, READ, SOMBRERO, and SOMEA were allowed to have the same map size, random initialization of the nodes, the same number of maximum training epochs, and the same expected k-mer length, and each tool was set to return the top 10 candidate motifs during each run on a dataset.

The WEEDER tool was run with the following options: sites might appear more than once, both strands, and normal or complete scan. MEME was run with 'any number' model option and the expected motif lengths were set similar to READ. AlignACE was run online with default arguments in most cases. These tools were set to return the top 10 motifs during each run on each dataset for a fair comparison. For details of parameter settings, refer [14].

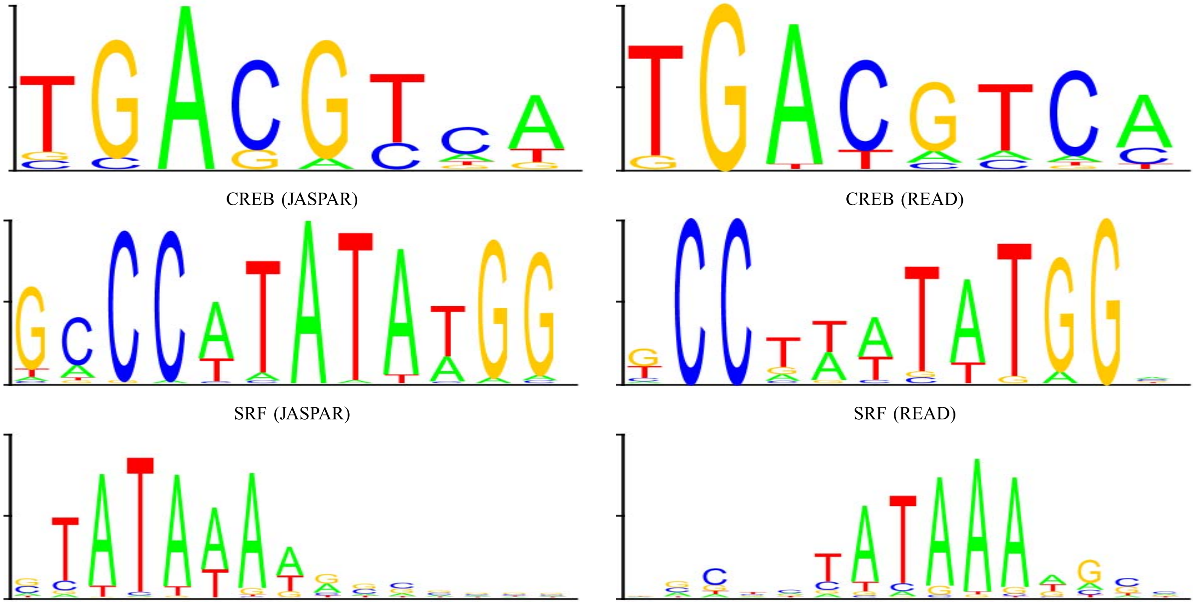
The best motif models in terms of F-measure, obtained from the search tools during each run, were collected for comparison purposes. The average over 10 runs of the R, P, and F-measure obtained by each tool on each dataset is presented in Table II, showing that READ can significantly outperform the state-of-the-art SOM-based SOMBRERO on CRP, MEF2, SRF, CREB, and MyoD datasets. Also, as shown in Table II, READ can achieve remarkable improvements in terms of the average recall (10.39%), precision (30.99%) and F-measure (24.66%) compared to the performances of SOMBRERO.

The higher recall rate of READ reflects that our FSOM-based approach is better in retrieving weak motif signals than the classical SOM-based approaches, while the higher precision rate is the effect of the postprocessing scheme employed in the READ framework. The crisp SOM-based approaches naturally tend to push the weak binding sites to the neighboring nonmotif clusters, which most likely has caused SOMBRERO to have a lower average recall rate (0.69) than that of READ (0.77). SOMEA is found to have a better performance than SOMBRERO due to its heterogeneous modeling scheme [14]. However, READ outperforms SOMEA on at least six datasets.

It has been observed that READ has achieved an average F-measure of 0.73, which is the best among the tools, includ-

TABLE II
PERFORMANCE COMPARISON USING EIGHT REAL DNA DATASETS

	The average of F-measure (F), recall (R) and precision (P) rates over 10 runs																	
	READ			SOMEA			SOMBRERO			MEME			AlignACE			WEEDER		
TF	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
CRP	0.76	0.84	0.80	0.91	0.89	**0.90**	0.83	0.43	0.56	0.59	0.88	0.69	0.83	0.98	**0.90**	0.75	0.83	0.79
GCN4	0.48	0.70	0.55	0.69	0.45	0.54	0.80	0.41	0.53	0.52	0.52	0.52	0.61	0.62	0.60	0.64	0.87	**0.73**
ERE	0.92	0.59	0.71	0.74	0.58	0.65	0.80	0.59	0.67	0.72	0.82	**0.77**	0.75	0.77	0.76	0.76	0.54	0.63
MEF2	0.96	0.87	**0.91**	0.81	0.99	0.89	0.35	0.22	0.27	0.92	0.80	0.85	0.86	0.87	0.86	0.88	0.88	0.88
SRF	0.91	0.77	**0.83**	0.84	0.74	0.79	0.67	0.83	0.74	0.87	0.72	0.79	0.83	0.71	0.77	0.83	0.71	0.76
CREB	0.82	0.78	**0.81**	0.89	0.67	0.77	0.83	0.43	0.56	0.59	0.88	0.69	0.52	0.66	0.57	0.79	0.71	0.75
E2F	0.69	0.74	0.71	0.82	0.64	0.71	0.76	0.67	0.71	0.68	0.64	0.65	0.75	0.68	0.71	0.89	0.67	**0.76**
MyoD	0.65	0.42	**0.51**	0.66	0.39	0.49	0.50	0.32	0.39	0.23	0.38	0.27	0.34	0.31	0.32	0.43	0.50	0.46
avg	0.77	0.71	**0.73**	0.80	0.67	0.72	0.69	0.49	0.55	0.64	0.71	0.65	0.69	0.70	0.69	0.75	0.71	0.72



CREB (JASPAR)

CREB (READ)

SRF (JASPAR)

SRF (READ)

TBP (JASPAR)

TBP (READ)

Fig. 4. Logos of the verified motifs of CREB, SRF, and TBP collected from the JASPAR database [45] and compared with the logos generated by READ on sample runs.

ing WEEDER (0.72), MEME (0.65), and AlignACE (0.69). These promising results imply good potential of READ in finding DNA motifs. The motif logos generated by READ are visually compared with the verified logos from the JASPAR database [45] in Fig. 4, showing a significant logo similarity. Certain dissimilarities between the logos come from the variations of the promoter sequences used.

B. Comparison Between SOMs, FSOMs and FCMs

In order to evaluate the effectiveness of the FSOM-based approach with respect to the crisp SOM-based approaches in DNA motif discovery, a parallel framework to READ is developed by only replacing the FSOMs with the classical SOM algorithm shown in (12), termed as READ$_s$. It was run on eight real DNA datasets with similar parameter settings as used in READ. The post-training crisp clusters were evaluated and ranked using MMS given in (4). The top 10 candidate motifs were extracted, and the best motif was selected using F-measure in each run.

It is also interesting to learn the benefits of employing FSOMs rather than FCM algorithms in DNA motif discovery. Hence, another variant of READ, termed as READ$_f$, is developed, where multiple runs of a standard FCM algorithm are conducted for varying k-mer lengths. The rest of the operations in READ$_f$ are exactly the same as READ, except for neighborhood-based candidate optimization, which cannot be conducted in READ$_f$ as there is no topological ordering of clusters in the FCM outputs. READ$_f$ was run on these datasets with the same number of cluster centroids assigned as the lattice size used in READ. The maximum number of FCM cycles was set to 100, but $J_m(t-1) - J_m(t) \leq 0.0001$ or $pc(\mu(t)) \geq 0.95$ conditions are checked for suitable early termination. The top 10 candidates are extracted after being ranked by the fuzzy MMS metric. The candidate motifs obtained from multiple FCMs for different k-mer lengths are refined with the same post-processing scheme as used in READ. Table III shows the results from the average over 10 runs for READ, READ$_s$, and READ$_f$.

Table III shows that READ with its FSOM-based approach produces better results than READ$_s$. Noticeably, READ's average recall rate (0.77) is found to be improved compared to that of READ$_s$ (0.66), which indicates the better ability

TABLE III

PERFORMANCE COMPARISON OF SOM, FCM, AND FSOM

	The average of R, P and F over 10 runs											
	READ				READ$_s$				READ$_f$			
TF	R	P	F	$\delta(F)$	R	P	F	$\delta(F)$	R	P	F	$\delta(F)$
CRP	0.76	0.84	**0.80**	(0.03)	0.68	0.73	0.70	(0.11)	0.61	0.77	0.68	(0.05)
GCN4	0.48	0.70	**0.55**	(0.10)	0.48	0.57	0.51	(0.08)	0.36	0.55	0.43	(0.05)
ERE	0.92	0.59	**0.71**	(0.06)	0.76	0.52	0.62	(0.14)	0.70	0.53	0.60	(0.08)
MEF2	0.96	0.87	**0.91**	(0.08)	0.70	0.63	0.66	(0.25)	0.68	0.60	0.64	(0.32)
SRF	0.91	0.77	**0.83**	(0.02)	0.79	0.78	0.78	(0.07)	0.58	0.67	0.62	(0.06)
CREB	0.82	0.78	**0.81**	(0.02)	0.79	0.76	0.78	(0.03)	0.77	0.71	0.74	(0.07)
E2F	0.69	0.74	**0.71**	(0.02)	0.56	0.60	0.58	(0.22)	0.63	0.50	0.56	(0.21)
MyoD	0.65	0.42	**0.51**	(0.09)	0.54	0.37	0.44	(0.10)	0.48	0.36	0.41	(0.15)
avg	**0.77**	**0.71**	**0.73**	(**0.05**)	0.66	0.62	0.64	(0.12)	0.60	0.59	0.58	(0.12)

Note: $\delta(F)$ denotes a standard deviation over 10 runs.

of the FSOM-based approach in motif discovery. Since these two frameworks only differ in terms of the SOM algorithm used, this comparison clearly indicates that the FSOM-based approach is more sensible in relation to the underlying fuzziness in the DNA datasets, and it is a potential way to improve the performance of traditional SOM-based DNA motif discovery tools.

Also, the use of FSOMs in generating candidates in the READ framework produces a significantly improved discovery performance than the use of FCMs, which indicates the practical benefits of employing the FSOMs over FCMs in DNA motif discovery. The introduced generality in the centroids by the neighborhood function in the early stage of FSOM training distinguishes the learning process of FSOMs from the clustering process of FCMs, which can be reasonably pointed out as one of the factors contributing to the improved performance of READ than READ$_f$. Additionally, the optimization of candidates using the immediate neighboring models in the FSOMs output lattice may be another reason behind the improved performance of READ. This observation is found complementary to the reported comparison between SOMs and the k-means clustering algorithm in [13], which points to the benefits of using SOMs in the SOMBRERO framework instead of similar size k-means clustering algorithms in DNA motif discovery.

The operational complexity of training SOMs, FSOMs, and FCM with N number of nodes on a dataset with M number of k-mers can be computed as follows: $\Omega_{\text{som}} = O(MN + MQP + N)$, $\Omega_{\text{fsom}} = O(MN + MQ_f N + N)$, and $\Omega_{\text{fcm}} = O(MN + MN)$, where the three subterms in Ω_{fsom} (Ω_{som}) refer to 1) the operations required for fuzzy (crisp) membership computation; 2) updating the temporary variables of each node by using N-size (P-size, $P < N$) neighborhood; and 3) updating each node PFM, respectively; while Ω_{fcm} excludes the neighborhood cooperating in adaptation. Note that $Q = 1$ in SOMs since each data sample has only one winner, and in practical implementation $Q_f (10 \sim 20 \ll N)$ is the number of centroids that have a considerable degree of fuzzy membership to a data sample while the other centroids have negligible close to zero membership, which can be ignored without losing the integrity of FSOMs. Also, in practice, the global neighborhood is mostly used in SOMs' implementation for computational convenience, i.e., $P = N$. These imply similar computational complexities between the practical implementation of SOMs and FSOMs. However, FSOMs naturally require more computational time than SOMs because of the exponential operations. A 10-run average training time of SOMs, FSOMs, and FCM on eight datasets were found as 99.20, 110.62, and 55.94 s, respectively, where the same number of nodes and a fixed number of cycles were set for comparison fairness using an Intel(R) Core(TM) i7-3612QM CPU @ 2.10 GHz machine.

C. Comparison Using Multiple Motif Datasets

In practice, multiple motifs are often found in the upstream regions of the co-expressed genes. Hence, it is interesting to see how these tools perform to simultaneously discover multiple motifs. For this purpose, five artificial datasets were generated from the Annotated Regulatory Binding Sites (ABS, v1.0) database [46], which have been used in our previous study [14]. Each dataset contains 20 sequences (500 bp each) with three real motifs (one for each TF) arbitrarily planted in the sequences.

READ, SOMEA, SOMBRERO, MEME, and WEEDER were run on these datasets with similar parameter settings as used in the experiments for single motif discovery. A 20×20 map size was used for the SOM-based tools. All tools were set to return the top-ranked 20 candidate motifs for each dataset. The best motif model associated with each TF in terms of F-measure was selected in each run for performance evaluation. The average of the R, P, and F-rates over 10 runs are presented in Table IV. Results produced by READ$_s$ on these datasets are also included in Table IV for comparison with the results of READ in order to see the improvement of the FSOM-based approach in finding DNA motifs. Clearly, the results presented in Tables III and IV demonstrate the benefit from the proposed FSOM learning over the classical SOM learning approach in DNA motif discovery.

Table IV shows that READ achieved a better average recall rate than SOMBRERO on Dataset$_{1,2,4,5}$. READ also produced a better recall rate than MEME and WEEDER on 11 out of 15 motifs. READ (0.45) obtained a remarkable 20% improvement over SOMBRERO (0.36) and a 6.6% improvement over SOMEA (0.42) in terms of the average F-measure computed using all datasets. Because of the better precision rate, MEME performed the best on the average F-measure. However, READ (0.52) obtained a noticeable 21.2% improved average recall rate over MEME (0.41), which is obviously advantageous in this complicated task of multiple motif discovery. Surprisingly, WEEDER performed poorly on these artificial datasets most likely due to the inability of its scoring function to discriminate the true motifs when planted in artificial sequences.

Note that in the multiple motif discovery tasks, the SOM-based tools face two major constraints. First, the SOM-based tools suffer from variations of the consensus length of the multiple motifs. Conducting multiple trainings using different k-mer lengths is a viable solution to resolve this problem and, indeed, it has been adopted in our READ framework to find motifs with variable k-mer lengths. Second, it is difficult to find an appropriate map size that can serve simultaneously

TABLE IV

PERFORMANCE COMPARISON USING ARTIFICIAL DATASETS WITH PLANTED MULTIPLE MOTIFS

		READ			SOMEA			SOMBRERO			MEME			WEEDER			READ (SOM)		
	3 TFs	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
$Dataset_1$	CREB	0.39	0.29	0.33	0.43	0.26	0.33	0.44	0.26	0.33	0.20	1.00	0.33	0.00	0.00	0.00	0.33	0.25	0.29
	MyoD	0.27	0.19	0.23	0.48	0.23	0.31	0.20	0.08	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.12	0.14
	TBP	0.28	0.20	0.23	0.36	0.21	0.26	0.20	0.12	0.15	0.07	0.50	0.12	0.00	0.00	0.00	0.19	0.14	0.16
	avg	0.31	0.23	0.26	0.42	0.23	**0.30**	0.28	0.15	0.20	0.09	0.50	0.15	0.00	0.00	0.00	0.23	0.17	0.20
$Dataset_2$	NFAT	0.36	0.29	0.32	0.39	0.27	0.31	0.36	0.21	0.26	0.44	0.78	0.56	0.00	0.00	0.00	0.30	0.24	0.26
	HNF4	0.85	0.64	0.73	0.57	0.40	0.47	0.63	0.39	0.48	0.60	0.82	0.69	0.40	1.00	0.57	0.73	0.55	0.63
	SP1	0.53	0.43	0.47	0.50	0.53	0.50	0.53	0.35	0.42	0.38	0.54	0.44	0.00	0.00	0.00	0.54	0.43	0.48
	avg	0.58	0.45	0.51	0.49	0.40	0.43	0.51	0.32	0.39	0.47	0.71	**0.56**	0.13	0.33	0.19	0.52	0.41	0.46
$Dataset_3$	CAAT	0.28	0.20	0.23	0.43	0.21	0.25	0.32	0.17	0.22	0.29	0.80	0.42	0.00	0.00	0.00	0.30	0.21	0.25
	SRF	0.49	0.35	0.40	0.70	0.40	0.50	0.59	0.28	0.38	0.29	0.57	0.38	0.00	0.00	0.00	0.36	0.25	0.30
	MEF2	0.61	0.46	0.52	0.79	0.45	0.57	0.65	0.31	0.27	0.80	0.57	0.67	0.27	1.00	0.42	0.53	0.40	0.45
	avg	0.46	0.33	0.38	0.64	0.35	0.44	0.52	0.25	0.29	0.46	0.65	**0.49**	0.09	0.33	0.14	0.39	0.29	0.33
$Dataset_4$	USF	0.61	0.52	0.56	0.68	0.39	0.48	0.73	0.48	0.57	0.41	0.88	0.56	0.00	0.00	0.00	0.35	0.30	0.32
	HNF3B	0.21	0.14	0.17	0.47	0.25	0.31	0.26	0.13	0.17	0.15	1.00	0.27	0.00	0.00	0.00	0.16	0.11	0.13
	NFKB	0.89	0.81	0.85	0.71	0.47	0.56	0.66	0.46	0.54	0.80	0.57	0.67	0.33	1.00	0.50	0.76	0.69	0.72
	avg	0.57	0.49	**0.53**	0.62	0.37	0.45	0.55	0.36	0.43	0.45	0.82	0.50	0.11	0.33	0.17	0.42	0.36	0.39
$Dataset_5$	GATA3	0.48	0.36	0.41	0.61	0.37	0.46	0.49	0.33	0.36	0.40	0.75	0.52	0.40	1.00	0.57	0.45	0.34	0.38
	CMYC	0.94	0.75	0.83	0.74	0.47	0.57	0.89	0.70	0.84	0.75	1.00	0.86	0.19	0.75	0.30	0.58	0.47	0.52
	EGR1	0.61	0.43	0.51	0.66	0.36	0.47	0.47	0.26	0.33	0.64	0.81	0.72	0.00	0.00	0.00	0.58	0.41	0.48
	avg	0.68	0.51	0.58	0.67	0.40	0.50	0.62	0.43	0.51	0.60	0.85	**0.70**	0.20	0.58	0.29	0.54	0.40	0.46
$avg\{5\ datasets\}$		0.52	0.40	0.45	0.57	0.35	0.42	0.49	0.30	0.36	0.41	0.71	**0.48**	0.11	0.32	0.16	0.42	0.32	0.37

TABLE V

ROBUSTNESS ANALYSIS OF SOM AND FSOMS BASED TOOLS USING DIFFERENT MAP SIZES

	The average of F-measure over 10 runs with different map sizes											
	$map\ size = 10 \times 10$			$map\ size = 15 \times 15$			$map\ size = 20 \times 20$			$standard\ deviation$		
TF	READ	SOMEA	SOMBRERO	READ	SOMEA	SOMBRERO	READ	SOMEA	SOMBRERO	READ	SOMEA	SOMBRERO
CREB	0.80	0.70	0.41	0.79	0.76	0.67	0.78	0.72	0.67	**0.008**	0.031	0.150
CRP	0.79	0.81	0.71	0.79	0.66	0.71	0.69	0.58	0.52	**0.060**	0.117	0.110
E2F	0.70	0.58	0.73	0.70	0.69	0.63	0.71	0.72	0.67	**0.004**	0.074	0.050
ERE	0.72	0.53	0.42	0.76	0.66	0.60	0.71	0.61	0.74	**0.028**	0.066	0.160
GCN4	0.53	0.41	0.44	0.50	0.51	0.52	0.49	0.58	0.60	**0.018**	0.085	0.080
MEF2	0.92	0.68	0.92	0.85	0.91	0.80	0.75	0.82	0.44	**0.087**	0.116	0.250
MyoD	0.50	0.32	0.23	0.52	0.49	0.42	0.44	0.47	0.49	**0.043**	0.093	0.135
SRF	0.82	0.70	0.67	0.81	0.77	0.72	0.72	0.71	0.71	0.055	0.038	**0.026**

the multiple motif discovery task. As can be seen, a smaller map size will produce a poor precision rate but an improved recall rate, while a bigger map size will produce a poor recall rate but an improved precision rate. As a matter of fact, all SOM-based tools are naturally sensitive to the map size setting even for single motif discovery. Such a problem is critical to multiple motif discovery tasks. Indeed, it is believed that the comparatively poor performance obtained by SOMBRERO comes from this size curse.

A large amount of training with different map sizes and varying k-mer lengths, both from a fairly large range, can be conducted using these SOM-based motif discovery tools in multiple motif discovery tasks. However, the computational burden becomes huge and longer postprocessing will have to be made. The proposed READ framework in this paper is able to automatically find a suboptimal solution through multiple training with varying k-mer lengths although the PFM prototypes are randomly initialized. Theoretically speaking, the random setting of initial weights can only result in locally optimal motifs before model merging and refinement. READ performs robustly because of the use of FSOM networks, the merging strategy, and the improved postprocessing scheme, which greatly contribute to handling the model uncertainty caused by map size setting and/or random model initialization.

D. Robustness Analysis

The number of clusters (prototypes) plays an important role in clustering algorithms, which greatly affects the clustering quality. Therefore, setting a proper map size is necessary for all SOM-based motif mining tools. This section reports some primary results on robustness analysis, which demonstrates the impact of the map size on the system performance.

Robustness can be defined as the ability of a system to maintain its performance when subjected to either noisy inputs or changes to the system's structure or parameters. In this paper, robustness refers to the sensitivity of discovery performances while different map sizes are set in SOM-based tools. In order to compare the robustness of READ in respect to SOMEA and SOMBRERO, we ran these tools on the eight real datasets using map sizes of 10 × 10, 15 × 15, and 20 × 20, respectively. Parameter settings were kept the same as those used in the single motif discovery task described earlier. During each run on each dataset, the best motif model selected from the top-ranked 10 candidate motifs in terms of F-measure was used to produce the results. The average figures of the F-measure rates over 10 runs are presented in Table V for comparison.

The last three columns in Table V give the standard deviation of the average F-measures obtained by each tool on these datasets with three different map sizes. It can be clearly seen that READ has the smallest std value in most of the cases, which indicates a better robustness than SOMBRERO and SOMEA in handling map size changes. A further justification of the robustness is shown in Table III, where our READ demonstrates much better reliability than READ$_s$ and READ$_f$.

With no universal solution to estimate an optimal map size (data dependent), usually the size setting is set through multiple empirical trials. This is a time-consuming process, particularly for large-scale datasets. We attempted to find a feasible means of reducing the impact of an improper map size setting on motif discovery performance. To a large extent, the merging cluster and the refinement scheme proposed in this paper greatly help in overcoming the negative impact on motif discovery performance, which may be caused by an improper map size setting in READ.

VI. Conclusion

Computational discovery of DNA motifs is a meaningful and challenging task in bioinformatics. From a learning-system perspective, finding putative motifs through learning is to mine a set of subtle patterns with some statistical significance. This cannot be done without a good understanding of DNA motifs, and, therefore, characterization on motifs and an appropriate similarity metric used in learning are essential to further improve the existing tools. Thus, there is still a plenty of room left in exploring unsupervised learning-based approaches for DNA motif discovery.

This paper contributes to the development of robust elicitation algorithms for DNA motif discovery using fuzzy SOM networks. A new batch learning algorithm for FSOMs was proposed by integrating the FCM membership function in the batch-learning algorithm of a standard SOM network. To achieve better and reliable performance, based on our previous work and the proposed learning scheme suggested in this paper, multiple FSOM networks are employed to extract candidate motifs with varying k-mer lengths from the input sequences followed by effective model merging and optimization processing. To fairly assess the merits and limitations of our proposed algorithm, a comprehensive performance comparison was carried out. Two SOM-based approaches, SOMBRERO and SOMEA, and other prominent tools including the well-known MEME, AlignACE, and WEEDER were tested using eight real datasets and five artificial datasets, respectively. The results indicated that our algorithm has good potential to favorably and robustly perform DNA motif discovery.

More research on this topic can be explored in many directions, e.g., development of adaptive SOM-based algorithms (i.e., the map size can be automatically adjusted during learning) for problem solving, investigations on semisupervised learning schemes, and case studies on weak motif discovery or on dealing with large-scale datasets in specific applications. It is also very interesting to see how to utilize the results obtained by other mining tools to further improve READ's performance.

Acknowledgment

The authors would like to thank the editors and anonymous reviewers for their constructive comments which greatly helped us to improve the quality of this paper.

References

[1] K. Yeung, M. Medvedovic, and R. Bumgarner, "From co-expression to co-regulation: How many microarray experiments do we need?" *Genome Biol.*, vol. 5, no. 7, p. R48, Jun. 2004.

[2] P. Cowie, R. Ross, and A. MacKenzie, "Understanding the dynamics of gene regulatory systems; Characterisation and clinical relevance of cis-regulatory polymorphisms," *Biology*, vol. 2, no. 1, pp. 64–84, Jan. 2013.

[3] T. I. Lee, R. G. Jenner, L. A. Boyer M. G. Guenther, S. S. Levine, R. M. Kumar, B. Chevalier, S. E. Johnstone, M. F. Cole, K. Isono, H. Koseki, T. Fuchikami, K. Abe, H. L. Murray, J. P. Zucker, B. Yuan, G. W. Bell, E. Herbolsheimer, N. M. Hannett, K. Sun, D. T. Odom, A. P. Otte, T. L. Volkert, D. P. Bartel, D. A. Melton, D. K. Gifford, R. Jaenisch, and R. A. Young, "Control of developmental regulators by Polycomb in human embryonic stem cells," *Cell*, vol. 125, no. 2, pp. 301–313, Apr. 2006.

[4] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young, "Genome-wide location and function of DNA binding proteins," *Science*, vol. 290, no. 5500, pp. 2306–2309, Dec. 2000.

[5] F. Zambelli, G. Pesole, and G. Pavesi, "Motif discovery and transcription factor binding sites before and after the next-generation sequencing era," *Briefings Bioinformat.*, vol. 14, no. 2, pp. 225–237, Mar. 2013.

[6] K. S. Leung, K. C. Wong, T. M. Chan, M. H. Wong, K. H. Lee, C. K. Lau, and S. K. W. Tsui, "Discovering protein-DNA binding sequence patterns using association rule mining," *Nucleic Acids Res.*, vol. 38, no. 19, pp. 6324–6337, Jun. 2010.

[7] 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, Oct. 1993.

[8] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learn.*, vol. 21, nos. 1–2, pp. 51–80, Oct./Nov. 1995.

[9] Z. Wei and S. T. Jensen, "GAME: Detecting cis-regulatory elements using a genetic algorithm," *Bioinformatics*, vol. 22, no. 13, pp. 1577–1584, Jul. 2006.

[10] D. Wang and X. Li, "iGAPK: Improved GAPK algorithm for regulatory DNA motif discovery," in *Proc. 17th Int. Conf. Neural Inf. Process.*, vol. 2, pp. 217–225, Nov. 2010.

[11] T. M. Chan, K. S. Leung, and K. H. Lee, "TFBS identification based on genetic algorithm with combined representations and adaptive post-processing," *Bioinformatics*, vol. 24, no. 3, pp. 341–349, Feb. 2008.

[12] C. Bates Congdon, J. C. Aman, G. M. Nava, H. R. Gaskins, and C. J. Mattingly, "An evaluation of information content as a metric for the inference of putative conserved non-coding regions in DNA sequences using a genetic algorithms approach," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 5, no. 1, pp. 1–14, Feb. 2008.

[13] S. Mahony, D. Hendrix, A. Golden, T. J. Smith, and D. S. Rokhsar, "Transcription factor binding site identification using the self-organizing map," *Bioinformatics*, vol. 21, no. 9, pp. 1807–1814, Jan. 2005.

[14] N. K. Lee and D. Wang, "SOMEA: Self-organizing map based extraction algorithm for DNA motif identification with heterogeneous model," *BMC Bioinformat.*, vol. 12, p. S16, Feb. 2011.

[15] D. Liu, X. Xiong, Z. G. Hou, and B. DasGupta, "Identification of motifs with insertions and deletions in protein sequences using self-organizing neural networks," *Neural Netw.*, vol. 18, nos. 5–6, pp. 835–842, Jun./Jul. 2005.

[16] D. Liu, X. Xiong, B. DasGupta, and H. Zhang, "Motif discoveries in unaligned molecular sequences using self-organizing neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 919–928, Jul. 2006.

[17] 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 quantization," *Nature Biotechnol.*, vol. 16, no. 10, pp. 939–945, Oct. 1998.

[18] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding signals of unknown length in DNA sequences," *Bioinformatics*, vol. 17, pp. S207–S214, Apr. 2001.

[19] J. K. Kim and S. Choi, "Probabilistic models for semisupervised discriminative motif discovery in DNA sequences," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 8, no. 5, pp. 1309–1317, Sep./Oct. 2011.

[20] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu, "Assessing computational tools for the discovery of transcription factor binding sites," *Nature Biotechnol.*, vol. 23, no. 1, pp. 137–144, Jan. 2005.

[21] J. Hu, B. Li, and D. Kihara, "Limitations and potentials of current motif discovery algorithms," *Nucleic Acids Research*, vol. 33, no. 15, pp. 4899–4913, Sep. 2005.

[22] D. Simcha, N. D. Price, and D. Geman, "The limits of de novo DNA motif discovery," *PLoS One*, vol. 7, no. 11, p. e47836, Nov. 2012.

[23] M. A. Lones and A. M. Tyrrell, "Regulatory motif discovery using a population clustering evolutionary algorithm," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 3, pp. 403–414, Jul. 2007.

[24] D. Wang and N. K. Lee, "Computational discovery of motifs using hierarchical clustering techniques," in *Proc. 8th IEEE Conf. Data Mining*, Dec. 2008, pp. 1073–1078.

[25] X. Ma, A. Kulkarni1, Z. Zhang, Z. Xuan, R. Serfling, and M. Q. Zhang, "A highly efficient and effective motif discovery method for ChIP-seq/ChIP-chip data using positional information,"*Nucleic Acids Res.*, vol. 40, no. 7, p. e50, Jan. 2012.

[26] G. Li, T. M. Chan, K. S. Leung, and K. H. Lee, "A cluster refinement algorithm for motif discovery," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 7, no. 4, pp. 654–668, Oct./Dec. 2010.

[27] M. C. Frith, Y. T. Fu, L. Q. Yu, J.-F. Chen, U. Hansen, and Z. Weng, "Detection of functional DNA motifs via statistical over-representation," *Nucleic Acids Res.*, vol. 32, no. 4, pp. 1372–1381, Feb. 2004.

[28] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1995.

[29] T. Villmann, R. Der, M. Herrmann, and T. Martinetz, "Topology preservation in self-organizing feature maps: Exact definition and measurement," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, pp. 256–266, Mar. 1997.

[30] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA USA: Kluwer, Jul. 1981.

[31] N. Chen, "Fuzzy classification using self-organizing map and learning vector quantization," in *Proc. Chin. Acad. Sci. Conf. Data Mining Knowl. Manag.*, Beijing, China, Jul. 2004, pp. 41–50.

[32] T. D. Schneider and R. M. Stephens, "Sequence logos: A new way to display consensus sequences," *Nucleic Acids Res.*, vol. 18, no. 20, pp. 6097–6100, 1990.

[33] G. D. Stormo and D. S. Fields, "Specificity, free energy and information content in protein-DNA interactions," *Trends Biochem. Sci.*, vol. 23, no. 3, pp. 109–113, Mar. 1998.

[34] A. Moses, D. Chiang, M. Kellis, E. Lander, and M. Eisen, "Position specific variation in the rate of evolution in transcription factor binding sites," *BMC Evol. Biol.*, vol. 3, no. 1, p. 19, Aug. 2003.

[35] D. Wang and S. Tapan, "MISCORE: A new scoring function for characterizing DNA regulatory motifs in promoter sequences," *BMC Syst. Biol.*, vol. 6, pp. S4–S18, Dec. 2012.

[36] F. Chin and H. C. Leung, "DNA motif representation with nucleotide dependency," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 5, no. 1, pp. 110–119, Mar. 2008.

[37] X. S. Liu, D. L. Brutlag, and J. S. Liu, "An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nature Biotechnol.*, vol. 20, no. 8, pp. 835–839, Aug. 2002.

[38] M. B. Eisen, "All motifs are not created equal: Structural properties of transcription factor—DNA interactions and the inference of sequence specificity," *Genome Biol.*, vol. 6, pp. 277–284, Mar. 2005.

[39] P. Vuorimaa, "Fuzzy self-organizing map," *Fuzzy Sets Syst.*, vol. 66, pp. 223–231, Sep. 1994.

[40] E. C.-K. Tsao, J. C. Bezdek, and N. R. Pal, "Fuzzy Kohonen clustering networks," *Pattern Recognit.*, vol. 27, no. 5, pp. 757–764, May 1994.

[41] R. D. Pascual-marqui, A. D. Pascual-montano, K. Kochi, and J. M. Carazo, "Smoothly distributed fuzzy *c*-means: A new self-organizing map," *Pattern Recognit.*, vol. 34, no. 12, pp. 2395–2402, Dec. 2001.

[42] W. Hu, D. Xie, T. Tan, and S. Maybank, "Learning activity patterns using fuzzy self-organizing neural network," *IEEE Trans. Syst., Man Cybern. B*, vol. 34, no. 3, pp. 1618–1626, Jun. 2004.

[43] M. M. Van Hulle, *Handbook of Natural Computing: Theory, Experiments, and Applications*. New York, NY, USA: Springer-Verlag, 2011.

[44] J. Zhu and M. Zhang, "SCPD: A promoter database of the yeast saccharomyces cerevisiae," *Bioinformatics*, vol. 15, no. 7, pp. 607–611, Dec. 1999.

[45] D. Vlieghe, A. Sandelin, P. J. De Bleser, K. Vleminckx, W. W. Wasserman, F. van Roy, and B. Lenhard, "A new generation of JASPAR, the open-access repository for transcription factor binding site profiles" *Nucleic Acids Res.*, vol. 34, pp. D95–D97, Jan. 2006.

[46] E. Blanco, D. Farre, M. M. Alba, X. Messeguer, and R. Guigo, "ABS: A database of annotated regulatory binding sites from orthologous promoters," *Nucleic Acids Res.*, vol. 34, pp. D63–D67, Jan. 2006.

Dianhui Wang (M'03–SM'05) received the Ph.D. degree from Northeastern University, Shenyang, China, in 1995.

He was a Post-Doctoral Fellow with Nanyang Technological University, Singapore, from 1995 to 1997, and a Researcher with The Hong Kong Polytechnic University, Hong Kong, from 1998 to 2001. He joined the Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia, in July 2001, and is currently a Reader and an Associate Professor. He is an Adjunct Professor at State Key Laboratory of Synthetical Automation of Process Industries, Northeastern University. His current research interests include data mining and computational intelligence systems for DNA motif discovery and transcription factor binding site detection, image recognition, restoration and retrieval, and engineering applications, such as big data modeling and classification for process industries.

Dr. Wang serves as an Associate Editor for several international journals, including *Information Sciences* and *Neurocomputing*.

Sarwar Tapan received the bachelor's degree in computer science from the University of Wollongong, Wollongong, Australia (INTI College Sarawak, Malaysia) in 2004, and the master's degree in cognitive sciences from the University of Malaysia Sarawak, Sarawak, Malaysia, in 2008. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia.

His current research interests include the applications of computational intelligence techniques in high dimensional data visualization and biological sequence analysis, with emphases on SOM-based approaches for regulatory motif discovery in promoter regions.