Motif Discoveries in Unaligned Molecular Sequences Using Self-Organizing Neural Networks

Derong Liu, *Fellow, IEEE*, Xiaoxu Xiong, *Student Member, IEEE*, Bhaskar DasGupta, *Senior Member, IEEE*, and Huaguang Zhang, *Senior Member, IEEE*

***Abstract*— In this paper, we study the problem of motif discoveries in unaligned DNA and protein sequences. The problem of motif identification in DNA and protein sequences has been studied for many years in the literature. Major hurdles at this point include computational complexity and reliability of the search algorithms. We propose a self-organizing neural network structure for solving the problem of motif identification in DNA and protein sequences. Our network contains several layers with each layer performing classifications at different levels. The top layer divides the input space into a small number of regions and the bottom layer classifies all input patterns into motifs and non-motif patterns. Depending on the number of input patterns to be classified, several layers between the top layer and the bottom layer are needed to perform intermediate classifications. We maintain a low computational complexity through the use of the layered structure so that each pattern's classification is performed with respect to a small subspace of the whole input space. Our self-organizing neural network will grow as needed (e.g., when more motif patterns are classified). It will give the same amount of attention to each input pattern and it will not omit any potential motif patterns. Finally, simulation results show that our algorithm outperforms existing algorithms in certain aspects. In particular, simulation results show that our algorithm can identify motifs with more mutations than existing algorithms and our algorithm works well for long DNA sequences as well.**

***Index Terms*— DNA sequences, motif finding, neural networks, protein sequences, self-organization, subtle signals.**

I. Introduction

DNA, RNA and proteins are important molecules that support life on Earth. There are 4 different kinds of nucleotides (A, C, G and T) that make up the DNA of all the organisms. These are the four base letters that constitute the alphabets of DNA. The four base letters of RNA are A, C, G and U, where the T in DNA is replaced by U in RNA. On the other hand, proteins of all the organisms are made up of 20 different kinds of amino acids (letters).

DNA, RNA and protein sequences can be thought of as being composed of motifs interspersed in relatively unconstrained sequence. A motif is a short stretch of a molecule that forms a highly constrained sequence [2]. The expression of a motif can be in one of the following forms.

1) Use an actual sequence as the description of a motif. Such a sequence is also called a consensus sequence [13], [28], [56]. Each position of the consensus sequence is the letter that appears most frequently in all known examples of that motif, e.g., $ACTTATAA$ and $AGTTATAA$ are two examples of consensus sequence of a motif.
2) Use a so-called "degenerate" expression to show all possible letters for each position of a motif [18], [26]. For example, the expression

$$A-[CG]-T-T-[AC]-[TCG]-A-A \quad (1)$$

indicates that $AGTTCTAA$ and $ACTTAGAA$ are two of the possible occurrences; see, for example, [40] for similar concepts used in the design of degenerate primers [33].
3) Use a more biologically plausible representation to describe a motif. In this case, a probability matrix can be used to assign a different probability to each possible letter at each position in the motif [4], [5], [21]. For example, Table I shows a probability matrix representation of the motif given by (1). This matrix representation not only gives the possibility of which letter can appear in each position of the motif, but also shows the probability of their appearances. For example, the sixth position of this motif will have letters C, G, and T appearing with probabilities of 20%, 30%, and 50%, respectively.

TABLE I
FREQUENCY OF EACH LETTER APPEARING IN EVERY POSITION OF A MOTIF

	1	2	3	4	5	6	7	8
A	1.0	0.0	0.0	0.0	0.67	0.0	1.0	1.0
C	0.0	0.5	0.0	0.0	0.33	0.2	0.0	0.0
G	0.0	0.5	0.0	0.0	0.0	0.3	0.0	0.0
T	0.0	0.0	1.0	1.0	0.0	0.5	0.0	0.0

4) Hidden Markov model (HMM) can also be used to describe motifs [15], [22]. An HMM is obtained by a slight modification of the Markov model. Based on HMM algorithm, an output matrix Π can be formed by the state transition matrix and the probability vector of A, G, C, and T associated with each state [36], [49]. It is a probabilistic model for motifs when we have

Manuscript received Oct. 19, 2004; revised Sept. 22, 2005. B. DasGupta was supported in part by NSF grants CCR-0296041, CCR-0206795, CCR-0208749 and IIS-0346973.

D. Liu and X. Xiong are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA (email: dliu@ece.uic.edu, xxiong@cil.ece.uic.edu). B. DasGupta is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA (email: dasgupta@cs.uic.edu). H. Zhang is with the School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110004, P. R. China (email: hgzhang@ieee.org).

Digital Object Identifier 10.1109/TNN.2006.123456

prealigned sequences [55] that are known to share some common blocks.

Understanding what motifs mean is a major part of research in bioinformatics. In order to understand motifs, one needs first to identify and locate them in DNA and protein sequences. By one way or another, biologists have identified some motifs [57]. They can explain their structures, common locations and certain functions. They are usually the beginning of translation of DNA to protein [44]. A protein binds optimally to places with some specific patterns (e.g., motifs) and it can still bind effectively even if one or more positions in the binding site sequence deviate from its ideal binding site sequence [1], [23], [34]. This means that a motif may have slightly different appearances at different locations [41]. The goal of this paper is to develop algorithms that can identify and locate motifs, if any, given a set of DNA or protein sequences.

Generally speaking, the motif finding problem in DNA sequences can be described as follows: Given a set of unaligned DNA or protein sequences, project the length of motifs and locate all motifs with the projected length that these sequences hold [6]. It is not necessary for all the sequences to have the same motif. Some sequences may have more than one repetition of a motif and some motifs may not show up in every sequence. The appearances of the same motif in the sequences are not necessarily the same. A subsequence[1] is determined to be a motif if it matches a possible appearance indicated by (1) or by the matrix representation in Table I. Obviously, information provided in Table I is more than that in (1). Here the frequency or probability of letters in each position of a motif is in $[0, 1]$. Usually the frequency of the letter that appeared most frequently should be larger than 40% [32], [48].

References [16], [35], [45] presented an unsupervised learning method for finding contiguous motifs. This kind of motifs has some biological properties of interest such as being DNA binding sites for a regulatory protein. The work in [16], [35], [45] showed that unsupervised learning method is a good approach for dealing with the problem of finding motifs. An algorithm called MEME is proposed in [2], [3] for identifying contiguous motifs. This algorithm is an extension to the expectation maximization algorithm for motif finding. The Gibbs sampling algorithm [38], [46], [56] uses a Monte Carlo procedure and it assumes motifs are ungapped sequence blocks. The algorithm tries to converge to a conserved block if it exists. Experimental results showed that the Gibbs sampling method misses motifs when the number of mutations is relatively large [55]. In this paper, we will develop an algorithm based on a new structure of self-organizing neural networks [19] and we will compare the performance of our algorithm with that of [2] and [38]. For motif identification, we will project the length of motifs as well as the maximum number of letters that can be mismatched in a pattern [48]. In this case, the target patterns to be found are described by a given length and by how many letters that can be mismatched.

[1]By subsequence we mean a *contiguous* part of the sequence; this is more commonly called "substring" in the string matching research community (e.g., see [24], [47]).

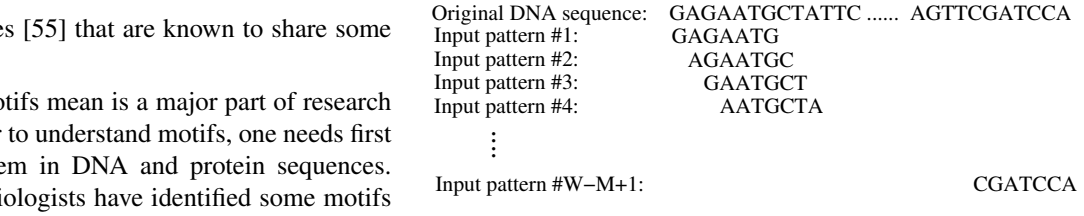

Fig. 1. An illustration on how to obtain input patterns ($M = 7$) from a given DNA sequence

Multiple sequence alignment method such as CLUSTALW [51], ITERALIGN [7] and PROBE [43] can also serve as motif identification tools. CLUSTALW aligns multiple sequences by calculating the global similarity among sequences. ITERALIGN and PROBE produce aligned blocks that are separated by variable-length unaligned segments. Sequence blocks in the alignment results of these methods can be treated as motif sets [43]. Usually these methods work on prealigned sequences and the conserved blocks they find have some limits, such as that the blocks must be in alignable position and at most one pattern from each sequence can be included in a motif set.

II. Self-Organizing Neural Networks for Motif Identification

A. Subsequences and Encoding

We consider the case where all motifs to be identified from a given set of DNA or protein sequences have the same length [42]. In general, the consensus sequence of a motif and the motif itself are not known *a priori* and we have to obtain them by using identification algorithms. What one obtains after the use of identification algorithms are specific appearances of a motif, usually with a few mismatched letter positions comparing to the motif consensus sequence. For a given set of DNA or protein sequences, in order to identify motifs in these sequences, we have to specify the maximum number of letter mismatches that can be tolerated (comparing to the consensus form) in addition to projecting the length of motifs to be found.

Test patterns, which we call input sequences or input patterns [30], can be obtained from the given set of DNA or protein sequences once the projected length of motifs is given. Fig. 1 shows a sketch of how input patterns are obtained from a DNA sequence. In the figure, the projected length of motifs is $M = 7$. All subsequences of seven connected letters obtained using a sliding window (see Fig. 1) from the given DNA or protein sequences will form the set of input patterns. For a DNA sequence of length W, we can obtain $W - M + 1$ input patterns if the projected length of motifs is M.

Letters used in DNA or protein sequences will be encoded using binary numbers [20]. All letters will be encoded using binary code with the same length, for example, four for DNA and RNA sequences and 20 for protein sequences. Table II shows an example of binary codes designed for DNA sequences. There are four letters in this case and each letter is encoded by flipping one bit of the standard code '1 1 0 0.' Letters coded this way will have exactly the same Hamming distance between any pair of letters [17], [52]. Also, the

TABLE II
ENCODER TABLE FOR DNA LETTERS

Standard	1	1	0	0
A	1	1	0	1
C	1	1	1	0
G	1	0	0	0
T	0	1	0	0

scheme shown in Table II can also guarantee that 1's and 0's will appear on average the same number of times. The coding scheme we used in the present paper is similar to [29] even though in reality certain pairs of letters may appear closer than others, e.g., in protein sequences, L and I are more similar than L and R [50].

B. A New Structure of Self-Organizing Neural Networks

This subsection describes the structure of our self-organizing neural networks for subtle signal discovery. The basic structure forms the subnetworks used in our self-organizing neural networks and contains two layers, i.e., an input layer and an output layer [8]–[12]. The number of output neurons of a subnetwork is the same as the number of categories classified by this subnetwork and the number of input neurons equals the projected length of motifs. The input patterns are obtained from the given DNA or protein sequences by taking all subsequences with the same length as the length of projected motifs (often in terms of the number of binary digits after encoding) [54]. Each output neuron represents a category that has been classified by a subnetwork and each output category is represented by the connection weights from all input neurons to the corresponding output neuron. Subnetworks perform functions of classification in a hierarchical manner. The first subnetwork is placed at the top layer and it performs a very rough classification, e.g., divide the input space into 4–8 categories. The second subnetwork is placed at the next layer and it usually divides the input space into 16–32 categories which indicates a slightly more detailed classification of the input space. The last subnetwork in our self-organizing neural network will be placed at the lowest layer and it classifies all the input patterns into either a motif or a non-motif category with one or a few patterns [37]. Typically, the number of output neurons will be large for the last subnetwork and gradually reduced to a small number for the first subnetwork. Fig. 2 shows the structure of our self-organizing neural network with three subnetworks. In the structure shown in the figure, there are four input neurons and three subnetworks. The first subnetwork has 3 output neurons, the second subnetwork has 5 output neurons, and the third subnetwork has 10 output neurons. Each of the output neurons represents a category that has been created and it is represented by the connection weights to the output neuron. The output category α of the first subnetwork contains two patterns (a and b), the output category β contains two patterns (c and d), and the output category γ contains one pattern (e). The output category a of the second subnetwork contains three patterns (1, 2, and 3), the output category b contains one pattern (4), the output category c contains two patterns (5 and 6), the output category d contains two patterns (7 and 8), and the output category e contains two patterns (9 and 10). The output categories 1, 2, 4 and 7 of the third subnetwork represent motifs while categories 3, 5, 6, 8–10 are not motifs (if we desire to have at least three appearances for each motif identified).

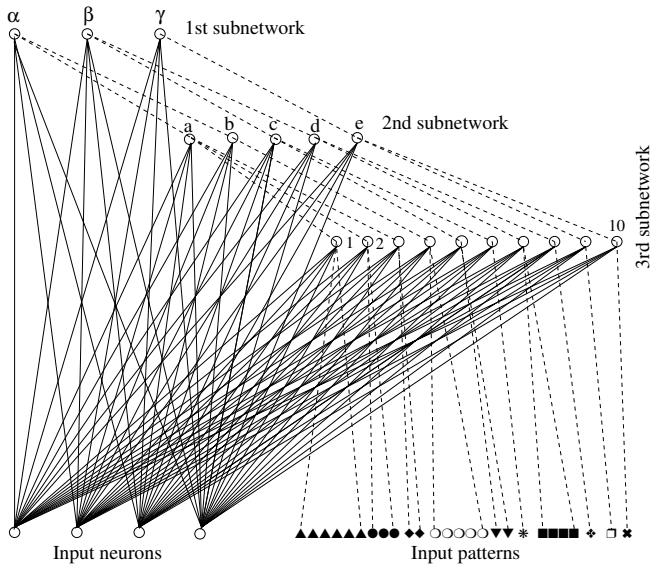

Fig. 2. Structure of the present self-organizing neural network

We can also illustrate the structure in Fig. 2 using a tree of sorting bins as shown in Fig. 3. In the figure, there are sorting bins at each level of the tree. From one level down to the next, the number of bins increases. At the lowest level, bins will be divided into motifs and non-motif categories. Fig. 3 also shows an example of how a new input pattern is sorted into a category. The new input pattern is first sorted by the bin at the top level. Then it is distributed to a suitable bin at the next level, and this process continues until the pattern reaches the lowest level where it is classified into a motif category or a non-motif category. By using the present neural network structure, the identification of motifs can be completed in one cycle of sorting (presenting all input patterns to the network). Multiple categories (at the lowest level) as shown in Fig. 3 will be generated in one cycle. On the other hand, existing methods for motif discoveries, such as MEME and Gibbs sampling methods, only sort the input patterns in each cycle into two groups: a motif category and a group containing all other patterns, as shown in Fig. 4. Using these algorithms, multiple trials will have to be employed so that multiple motifs can be discovered.

C. Rules for Weight Update and Output Node Creation

When an input pattern is applied to our self-organizing neural network, it will be classified to an output category by every subnetwork. An output category of a lower layer subnetwork is said to belong to an output category of a higher layer subnetwork if one or more input patterns are classified to belong to these two output categories. The connection weights for each category of the last subnetwork (at the lowest layer)

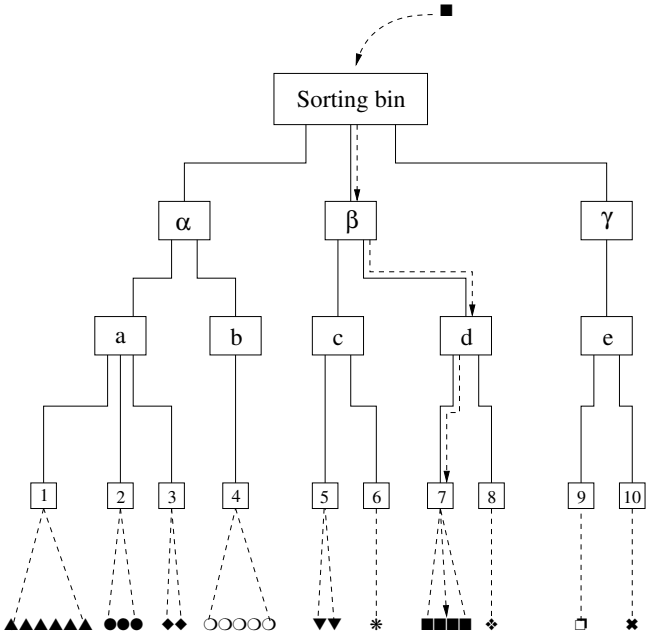

Fig. 3. Sorting strategy of the self-organizing neural network method

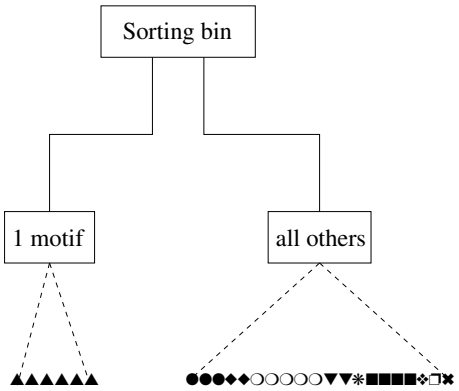

Fig. 4. Sorting strategy of the MEME and Gibbs methods

are calculated as the center of the category, i.e., the geometric center of all input patterns that are currently classified into the category associated with the corresponding output neuron. The connection weights for an output category of all other subnetworks (except the last subnetwork) are calculated as the geometric center of all categories from the lower layer subnetwork that belongs to this category.

When a new input pattern is applied to a subnetwork, its classification to an output category of every subnetwork involves the following two steps.

1) The distance between the input pattern and each output category is calculated by comparing the input pattern with the connection weights from the input neurons to that category. The minimum of these distances is determined and thus a winning category is also determined. This step works similarly to the winner-take-all networks [25]. These winning neurons form the tree of classification as in Fig. 2. For the example network shown in Fig. 2, an input pattern will be first compared to the three categories $\{\alpha\}$, $\{\beta\}$ and $\{\gamma\}$ at the first layer. At the next layer, it will be either compared to $\{a, b\}$, $\{c, d\}$ or $\{e\}$ depending on which of the three output categories at the first layer becomes the winning category.
2) Within the winning category, the similarity of all patterns in this category including the new pattern will be calculated and compared to a threshold value. If the similarity value is less than the threshold, the new pattern will be classified into the winning category. Otherwise, the new pattern cannot be classified into the winning category.

Assume that there are a total of L subnetworks for $l = 1, 2, \cdots, L$. Assume that there are M input neurons and the lth subnetwork has N_l output neurons. The input patterns obtained from the given DNA or protein sequences are used as motif candidates and are provided to each subnetwork of our self-organizing neural network. The outputs of the last subnetwork correspond to classifications of all input patterns into motifs and non-motif categories. The projected length of motifs possibly existing in the input sequences is the same as M.

We denote the input patterns as x^i, $i = 1, 2, \cdots$. Suppose that t input patterns have been presented to the network and have been classified. When a new input pattern, i.e., the $(t+1)$st pattern x^{t+1}, is introduced to the lth subnetwork, the distances from the new input pattern to those categories of the lth subnetwork that belong to the $(l-1)$st winning category W_q^{l-1} is calculated as

$$y_n^l = \sum_{m=1}^{M} |x_m^{t+1} - w_{mn}^l|, \text{ for } n \in W_q^{l-1}$$

where x_m^{t+1} is the mth component of the input pattern x^{t+1} and w_{mn}^l is the connection weight of the lth subnetwork from the mth input neuron to the nth output neuron after the presentation of the tth input pattern. Denote

$$y_q^l = \min_{n \in W_q^{l-1}} \{y_n^l\}$$

i.e., the qth output category of the lth subnetwork is the winning category that has the smallest distance to the new input pattern. Assume that the qth output category of the lth subnetwork contains p_q^l patterns from the $(l+1)$st subnetwork. Within this winning category q, we will calculate the similarity value of all the $p_q^l + 1$ patterns including the new input pattern. The similarity value of a group of patterns is calculated as the maximum of the pairwise distance [27] between all pairs of patterns in the group.

For the winning category q determined above, we calculate the distances from the new input pattern to all other patterns in the category as

$$d_j^l = \sum_{m=1}^{M} \left| x_m^{t+1} - e_{mj}^{l+1} \right|, \ j = 1, 2, \cdots, p_q,$$

where

$$e_{mj}^{l+1} = \begin{cases} x_m^j, & \text{if } l = L \text{ and } x_m^j \text{ belongs to the} \\ & \text{category } q \text{ of the } (l-1)\text{st layer} \\ w_{mj}^{l+1}, & \text{if } 1 \le l < L \text{ and } w_{mj}^{l+1} \text{ belongs} \\ & \text{to the category } q \text{ of the } l\text{th layer.} \end{cases} \quad (2)$$

We then perform the following threshold tests. If

$$\max_{1\le j\le p_q}\{d_j^l\} < \rho_l \tag{3}$$

then this new input pattern will be classified into the category q of the lth subnetwork. Otherwise, the new input pattern cannot be classified into any existing category at this layer. The threshold value ρ_l in (3) will be determined later and it takes different values for different subnetworks. We note that all pairwise distances in this category will be less than the threshold ρ_l if (3) is satisfied for the new input pattern since all other patterns are previously classified into this category using the same threshold test.

We describe in the following some more details about our calculation procedure.

a) We start from the top layer, i.e., the first subnetwork, and work down the layers one by one, when classifying a new input pattern. After a winning category has been determined at the lth layer, the input pattern will only be compared to those patterns at the $(l+1)$st layer that are classified to belong to the winning category at the lth layer and the winning category is denoted by W_q^l.

b) If the threshold tests in (3) are successful for $l = 1, 2, \cdots, L$, we perform the following updates for the Lth subnetwork:

$$\begin{aligned} w_{mq}^L &:= \frac{1}{p_q^L+1}\sum_{j=1}^{p_q^L+1} x_m^j \\ &= \frac{1}{p_q^L+1}\left[p_q^L \times w_{mq}^L + x_m^{t+1}\right], \\ & m = 1, 2, \cdots, M, \\ p_q^L &:= p_q^L + 1, \end{aligned}$$

where $x^{p_q^L+1}$ indicates the new input pattern x^{t+1} for convenience. We perform the following updates for the rest of subnetworks:

$$w_{mq}^l := \frac{1}{p_q^l}\sum_{j=1}^{p_q^l} w_{mj}^{l+1},$$

$$m = 1, 2, \cdots, M,\ l = L-1, L-2, \cdots, 2, 1.$$

c) If the threshold tests in (3) are successful for $l = 1, 2, \cdots, L_1$, where $L_1 < L$, we will add an output neuron to subnetworks $L_1+1, L_1+2, \cdots, L$. Each of these newly added categories will contain only one pattern and the weights of the new categories are chosen as

$$w_{mn}^l = x_m^{t+1},$$

$$m = 1, 2, \cdots, M,\ n = N_l + 1,$$

$$l = L_1 + 1, L_1 + 2, \cdots, L.$$

We also update the number of output neurons for these subnetworks as

$$N_l := N_l + 1,\ p_{N_l}^l = 1,\ l = L_1+1, L_1+2, \cdots, L.$$

In this case, it is not necessary to perform threshold tests for subnetworks $L_1+1, L_1+2, \cdots, L$ anymore. For subnetworks $1, 2, \cdots, L_1$, we will perform the following updates:

$$p_q^{L_1} := p_q^{L_1} + 1$$

$$w_{mq}^l := \frac{1}{p_q^l}\sum_{j=1}^{p_q^l} w_{mj}^{l+1},$$

$$m = 1, 2, \cdots, M,\ l = L_1, L_1-1, \cdots, 2, 1.$$

D. Order Randomization and Recycling of Input Patterns

After one cycle of the motif identification procedure, our neural network is able to identify most of the patterns belonging to some motifs. However, there might still be some missing ones. That is because that the classification of an input pattern to a category using the present self-organizing neural network will be affected by the order in which input patterns are presented to the network. The new input pattern will only be tested in existing categories in the network. If the pairwise test wins in an earlier category, the pattern will not be included in categories built later. Fig. 5(a) shows a case where an input pattern is placed in a non-motif category A (e.g., a category with less than two members). After that, the same pattern may not be considered to belong to a motif category B that is created after. In Fig. 5(b), the same input pattern is classified to the motif category B since in this case the category B is created before A.

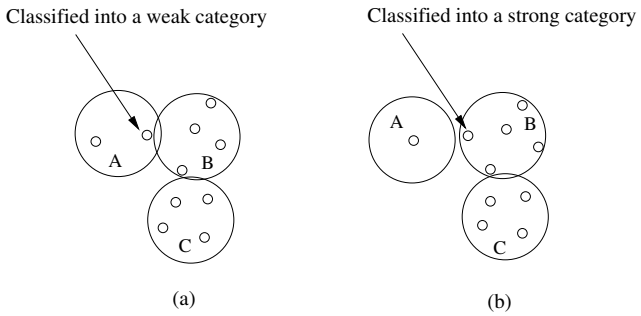

Fig. 5. (a) A new input pattern fails to be classified into category B. (b) The classification succeeded in a different trial.

To avoid the problem shown in Fig. 5(a), we use the following procedure. After the first trial, we keep all motif categories and recycle all input patterns in non-motif categories to determine whether we have misclassified any patterns during the first trial. (1) Initial trial: We randomly select the order of presentation of all input patterns, and run our algorithm to identify motif categories. (2) Recycling input patterns: Keep all motif categories including all patterns belonging to these categories, remove all non-motif categories, randomly select the order of presentation of input patterns from non-motif categories, and run our algorithm. After the second trial with recycled input patterns, the problem shown in Fig. 5(a), if any, will be resolved. Thus, it is likely some motif categories will get new members to join. It is also likely that some more motif categories will be created.

In our simulation studies, we have used 3 as the threshold to determine whether a category is a motif or not, i.e., if a

category contains three members or more, it is classified as a motif category and otherwise, it is not. Our simulation results also indicate that two trials are enough to identify all motif categories since additional trials have not produced anything new.

III. Simulation Results

We will compare our algorithm with existing algorithms in the present simulation studies. We will use both randomly generated and real DNA sequences to test our algorithm. In each example, input patterns to our self-organizing neural network will be obtained from DNA or protein sequences as described in Section 2.1.

Example 1: In this example, we will apply our algorithm to motif discoveries in Ornithine Carbamoyltransferase family protein sequences (OTCase family). We choose 9 OTC samples from SwissProt gene library. The lengths of these sequences are between 305 and 340 letters. The average length is 322. We project the length of the target motif to be 17 and the maximum number of mismatched letters to be 4. A total of 2754 input patterns are obtained from the 9 protein sequences. We choose to use three levels of subnetworks with one output neuron initially at each level. After the presentation of all input patterns in random order to the network, we obtain 8 motif sets each has at least 5 appearances. Results are shown in Fig. 6. The motif sets are marked with different underlines or blocks. The consensus forms of the motif sets are used to summarize

```
                1
OTC2_ECOLI   SDLYKKHFLKLLDFTPAQFTSLLTLAAQLKADKKNGKEVQKLTGKNIALIFEKDSTRTRCSFEVAAFDQGARVTYL
OTC1_PSESH   NARHFLSMMDYTPDELLGLIRRGVELKDLRIRGELFEPLKNRVLGMIFEKSSTRTRLSFEAGMIQLGGQAIFLSHR
OTC1_ECOLI   SGFYHKHFLKLLDFTPAELNSLLQLAAKLKADKKSGKEEAKLTGKNIALIFEKDSTRTRCSFEVAAYDQGARVTYL
OTC1_LACLA   MFQGRSFLKEIDFSKDELLYLIDFAIHLKKLKKEHIQHKYLLDKNIALIFEKTSTRTRAAFTTAAVDLGAHPEFLG
OTC2_LACLA   MVTTNKRDFITTEDYTKEEILDIVTLGLKIKAAIKNGYYPPLLKNKSLGMIFQQTSTRTRVSFETAMTQLGGHAEY
OTC2_PSESF   KITSLKNRNLLTMNEFNQSELSHLIDRAIECKRLKKDRIFNLGLNHLNICGIFLKPSGRTSTSFVVASYDEGAHFQ
OTC_BACAN    MSTVQVPKLNTKDLLTLEELTQEEIISLIEFAIYLKKNKQEPLLQGKILGLIFDKHSTRTRVSFEAGMVQLGGHGM
OTC_ANASP    MAALLGRDLLSLADLTPTELQELLQLATQLKSQQLKLRCNKVLGLLFSKASTRTRVSFTVAMYQLGGQVIDLNPNV
OTC_AQUAE    MKRDFVDLWDLSPKEAWEIVKKTLKVKKGEEELGKPLSGKTIALLFTKPSTRTRVSFEVGIYQLGGNSLFFQEKEL

                77
OTC2_ECOLI   GPSGSQIGHKESIKDTARVLGRMYDGIQYRGHGQEVVETLAQYAGVPVWNGLTNEFHPTQLLADLMTMQEHLPGKA
OTC1_PSESH   DTQLGRGEPIADSAKVMSRMLDAVMIRTYAHSNLTEFAANSRVPVINGLSDDLHPCQLLADMQTFLEHRGSIKGKT
OTC1_ECOLI   GPSGSQIGHKESIKDTARVLGRMYDGIQYRGYGQEIVETLAEYASVPVWNGLTNEFHPTQLLADLLTMQEHLPGKA
OTC1_LACLA   PNDIQLGKKESISDTAKVLGSMFDGIEFRGFKQSDVEILAKDSGRPVWNGLTDVWHPTQMLADFMTIKEHFGHLQD
OTC2_LACLA   LAPGQIQLGGHETIEDTSTVLSRLLDIIMARVDRHESVNNLAKHTTIPVLNGMSDYNHPTQEVGDLTTMIEHLPAG
OTC2_PSESF   FFPADNIRFGHKESIKDFARVVGRLFDGIAFRGFEHEVAEELAKHSGIPVWNALTDTHHPTQVLADVMTVKEEFGR
OTC_BACAN    FLNGKEMQMGRGETVSDTAKVLSHYIDGIMIRTFSHADVEELAKESSIPVINGLTDDHHPCQALADLMTIYEETNT
OTC_ANASP    TQVSRGEPVQDTARVLERYLDVLAIRTFEQQELATFAEYAKIPVINALTDLEHPCQILADLLTVQECFDSISGLTL
OTC_AQUAE    QVSRGEDVRDTARTLSKYVDGVIVRNHSHTWLKEFANFASVPVINALTNMSHPCQILSDVFTLYEHYGEELKNLKV

                153
OTC2_ECOLI   FNEMTLVYAGDARNNMGNSMLEAAALTGLDLRLLAPKACWPEESLVAECSALAEKHGGKITLTEDVAAGVKGADFI
OTC1_PSESH   VAWIGDGNNMCNSYIEAAIQFDFQLRVACPAGYEPNPEFLALAGERVTIVRDPKAAVAGAHLVSTDVWTSMGQEEE
OTC1_ECOLI   FNEMTLVYAGDARNNMGNSMLEAAALTGLDLRLVAPQACWPEAALVTECRALAQQNGGNITLTEDVAKGVEGADFI
OTC1_LACLA   LTLAYVGDGRNNVANSLLVTGAILGVNITIISPESLQPALEIQKLARKYAMKSRSKISIRTDLNGLENADIVYTDV
OTC2_LACLA   KKLEDCKVVFVGDATQVCFSLGLIATKMGMHFVHFGPKGYQLNEEHQAKLAANCEVSGGTYEVTDDEESIVGADFL
OTC2_PSESF   IEGVTIAYVGDGRNNMVTSLAIGALKFGYNLRIIAPNALHPTDAVLAGIYEQTPERNGSIEIFTEVAAGVHQADVI
OTC_BACAN    FKGIKLAYVGDGNNVCHSLLLASAKVGMHMTVATPVGYKPNEEIVKKALAIAKETGAEIEILHNPELAVNEADFIY
OTC_ANASP    TYVGDGNNVANSLMLGCALAGMNVRIATPSGYEPNPQVVAQAQAIADGKTEILLTNDPDLATKGASVLYTDVWASM
OTC_AQUAE    AYVGDGNNVCNTLMVGAGMFGLKLFVATPEGYEPNSYYYKKALEFSKENGGSVELTNNPVESVKDADVVYTDVWVS

                229
OTC2_ECOLI   YTDVWVSMGEAKEKWAERIALLRGYQVNAQMMALTDNPNVKFLHCLPAFHDDQTTLGKQMAKEFDLHGGMEVTDEV
OTC1_PSESH   TARRMALFAPFQVTRASLDLAEKDVLFMHCLPAHRGEEISVDLLDDSRSVAWDQAENRLHAQKALLEFLVAPSHQR
OTC1_ECOLI   YTDVWVSMGFAKEKWAERIALLREYQVNSKMMQLTGNPEVKFLHCLPAFHDDQTTLGKKMAEEFGLHGGMEVTDEV
OTC1_LACLA   WVSMGEEAQTAKRIKLLKSYQINQKVVEKIINKNFIFMHCLPSFHDLNTEVMKEIKENYNLNELEVTDEVFNSKNS
OTC2_LACLA   YTDVWYGLYDAELSEEERLAIFFPKYQVTPEMMAKAGAHTKFMHCLPASRGEEVVDAVIDGPNSICFDEAENRLTS
OTC2_PSESF   YTDVWISMGESVSVEERIALLKPYKVTEKMMALTGKADTIFMHCLPAFHDLDTEVARETPDLVEVEDSVFEGPQSR
OTC_BACAN    TDVWMSMGQEGEEEKYTLFQPYQINKELVKHAKQTYHFLHCLPAHREEEVTGEIIDGPQSIVFEQAGNRLHAQKAL
OTC_ANASP    GQFAEADDRFPIFQPYQISEQLLSLAEPNAIVLHCLPAHRGEEITEEVIEGSQSRVWQQAENRLHVQKALLASILG
OTC_AQUAE    MGEENKNIEAFLPYQVNEKLLSFAKSSVKVMHCLPAKKGQEITEEVFEKNADFIFTQAENRLHTQKTLMEFLFREP

                305
OTC2_ECOLI   FESAASIVFDQAENRMHTIKAVMMATLGE
OTC1_PSESH   A
OTC1_ECOLI   FESAASIVFDQAENRMHTIKAVMVATLSK
OTC1_LACLA   VVFEQAENRMHTIKEVMAATLGDLFIPKI
OTC2_LACLA   IRALLVWLMSDYAEKNPYDLKAQAKAKAELEAYLAK
OTC2_PSESF   VFDQGENRMHTIKALMLETVVP
OTC_BACAN    LVSLFKNVEELS
OTC_ANASP    AE
OTC_AQUAE    QA
```

Motif consensus form:

```
Motif 1:  PDELLHLIDRAIELKRL      Motif 5:  GLTLAYVGDGRNNMNNS
Motif 2:  NKNIGLIFEKPSTRTRV      Motif 6:  KGADVIYTDVWVSMGEE
Motif 3:  QFGHKESIKDTARVLGR      Motif 7:  EEEKRIALFRPYQVNKK
Motif 4:  WNGLTDDHHPTQLLADL      Motif 8:  VKFMHCLPAFHDDETTE
```

Fig. 6. The motif discovery results in OTCase family proteins

TABLE III

COMPARISON OF THE MOTIF SETS FROM SELF-ORGANIZING NEURAL NETWORK METHOD AND MEME METHOD, NOS = NUMBER OF SAMPLES IN THE MOTIF SET.

ID	Our Result		MEME Result	
	Consensus	NOS	Consensus	NOS
1	GTACAGTTTGTTTATAC	9	GTACAGTTTGTTTATAC	9
2	ACCTTCCACTCAGGATG	11	ACCTTCCACTCAGGATG	11
3	CAAAAGAACAATAATCA	8	CAACATATTCATAGTCT	7
4	CACAGAGGCACCAATTT	7	CACACCCCGAGCATTCT	6
5	CTTCTTGGAATCCTCTG	7	CTTCTTGGAATCCTCTG	7
6	TTCTAATATTTATTGCT	7	TTCTAATATTTATTGCT	4
7	TGGACTTGGAACCTATA	7	GACTCGCAACCTACAAA	4
8	AGCTTTCTGAATAAAAG	7	ACTTTCTGAATAAAAGA	6
9	TACAATAACAATACCTT	7	TACAGGAAAGATACCTT	5
10	GGTGAGTCTGTGCATTT	7	AATGAGTCTGTGCATAT	7
11	TTCCAATACATTAATAT	7	CCCTTTTCTCTACATTT	7
12	CCATCGATCGAACGATT	7	CGATCAATCGAACGATT	4
13	CCCCCACCTCTCATCAG	7	CATCTCCCATCAGTCAT	4
14	ACTTTCTGAATAAAAGA	6	GACAATCAAAGGAAACA	5
15	GGGTTCGGCAGATGTTT	6	GGGTGGGGCAGTTGTTT	4
16	TTTGTGGTTCAAAATAT	13		
17	AGGAATTTAAAACAAAT	11		
18	AATAATAAAGTAAAAAA	11		
19	ATATTTTTTTTCTTCAG	10		
20	ACCTTTCGGATAAAACC	6		

the results.

Example 2: In this example, we will test our algorithm on a group of DNA sequences that share strong and weak motifs [31], [39]. The target samples are ancient conserved untranslated sequences (ACUTS). The DNA samples are obtained from the ACUTS database [58]. The ACUTS DNA sequences are usually used in identifying new regulatory elements in untranslated regions of protein-coding genes [14]. We pick the ACTAC_3UT entries which included 7 pieces of sequences. The lengths of the sequences are between 98 and 1866 residues and the average length is 525. The projected length of target motifs is 17 and the maximum number of mismatched letters is 6. A total of 3225 input patterns are obtained from the 7 DNA sequences. After applying the input patterns to our neural network, we obtain a total of 20 motif sets. In order to compare our algorithm with MEME method, we apply the same DNA sequences to the MEME online server (http://meme.sdsc.edu/meme/website/meme.html). The MEME method finds 15 motif sets. Table III shows a comparison between the motif sets we found and MEME results. In the table we list the consensus sequence of each motif set obtained by both self-organizing neural network method and MEME method. We list the number of patterns that are found for each motif set. The first 15 motif sets are those found by both our algorithm and MEME method. Compared with MEME method, our algorithm finds more patterns for most of these motif sets. Motif sets 16 to 20 are found by our method only.

Example 3: In this example, following [53], we generate i.i.d. samples of DNA sequences with certain lengths. Motifs with random mismatch letters at randomly chosen positions are implanted in these sequences. The performance of the algorithm is defined as follows:

$$P_{erf} = \frac{|R \cap T|}{|R \cup T|} \tag{4}$$

where R is the motif set generated, T is the motif set identified, and $|\cdot|$ indicates the cardinality of a set. The numerator of the performance represents the number of motifs we found that are really motifs. The denominator represents the whole set of any motifs that are generated or found. In the figures shown in this example, the horizontal axis represents the percentage of mismatch of the motifs (i.e., ϵ/M, where ϵ is the number of letters that is tolerable as the representation of a motif), and the vertical axis indicates the performance averaged over 8 such simulations. A result that is closer to 1 implies a better performance.

Fig. 7 to 9 show the performance of the system on finding motifs of lengths 13, 15 and 17. From the figures we can see that the using the present self-organizing neural network, results are still acceptable even with the mismatch letters up to 30%. After that, the performance drops sharply. The reason of the sharp drop is that for the 4 letter DNA case, the total number of randomly generated sequences is not large enough, which makes the generated patterns to be often similar to noise. Comparing to the results obtained using MEME in [2] and using Gibbs in [38], our simulation results can find motifs with at least one more mismatch letter than the other two algorithms. For example, for motif length of 15, our algorithm achieved 100% performance (i.e., identified all motifs) when there are four letter mismatches allowed, while MEME and Gibbs algorithms both achieved less than 20% performance. We can conclude that in this aspect our algorithm outperforms the MEME and Gibbs algorithms. In this simulation example, we generated 10 DNA sequences with 200 letters in each sequence. The computation time of our algorithm is 3 minutes on a SUN Ultra 60 workstation. Compared with MEME (15 minutes) and Gibbs (12 minutes), our algorithm demands less computation time.

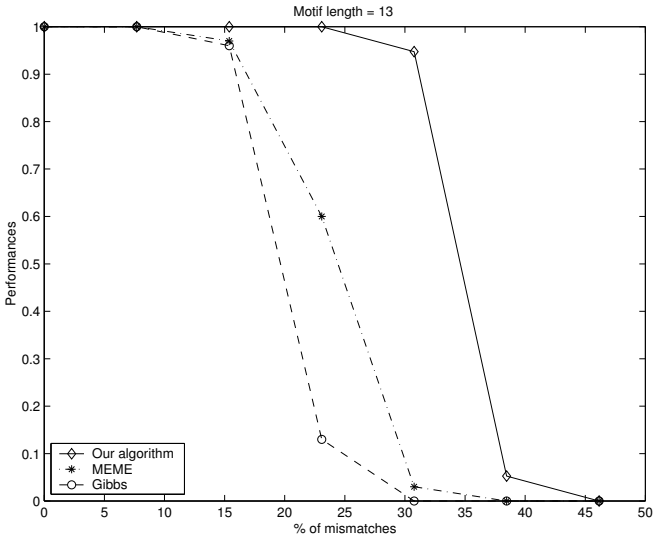

Fig. 7. Comparison results for motif length = 13

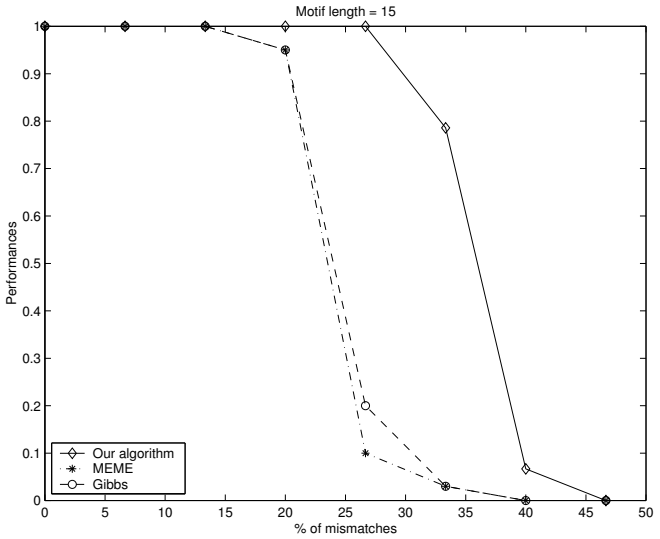

Fig. 8. Comparison results for motif length = 15

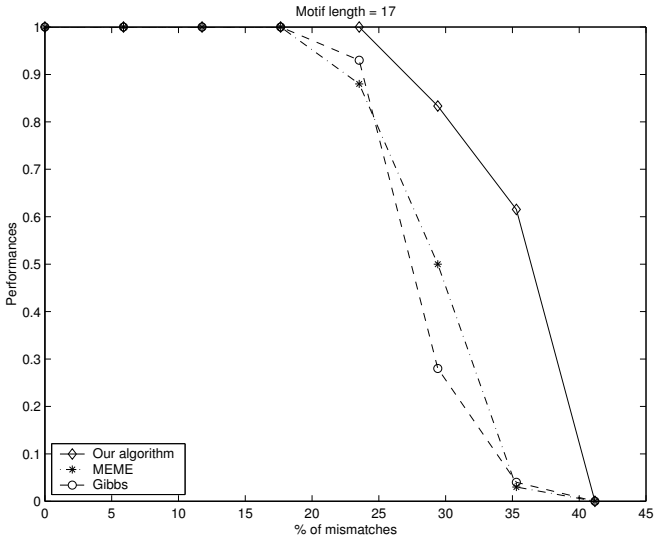

Fig. 9. Comparison results for motif length =17

Example 4: In this example, we study the performance of our algorithm with respect to protein sequences. We use the same strategy as defined in the last example. We generate i.i.d samples of protein sequences and certain number of protein motifs with mismatch letters. Fig. 10 shows performances of our algorithm for both DNA and protein sequences. The length of the motif patterns in both cases is chosen as 15. We can see that the performance of our algorithm for protein sequences is better than that for DNA sequences. One reason for this improved performance is the large number of random sequences that can be generated in the case of protein sequences due to large alphabet.

Example 5: Existing algorithms such as MEME and Gibbs do not perform well for long DNA sequences. Based on the work in [48], the performances of these two algorithms are not good enough when the length of the sequence hits 500. In this example, we make a comparison for performance of motif identification using long DNA sequences. In the present

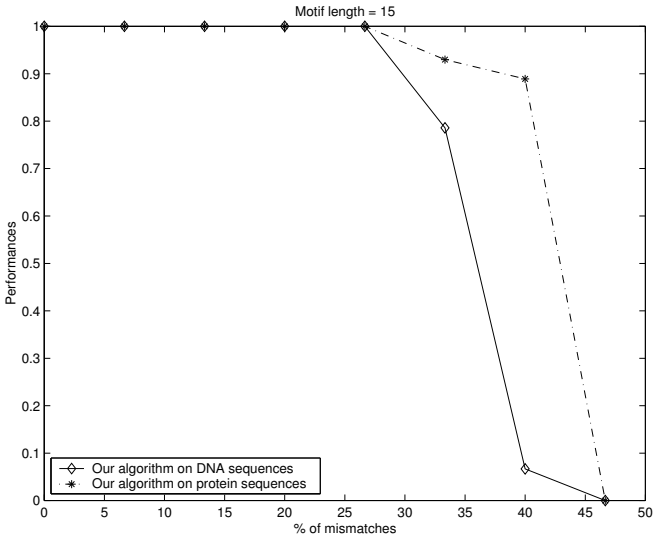

Fig. 10. Comparison results for DNA and protein sequences with motif length = 15

example, we generate 20 DNA sequences each with length of 1000. A total of 30 patterns of (15, 4) are implanted at random locations in these sequences, where 15 indicates the motif length and 4 represent the tolerable number of mismatch letters. We perform a total of 8 simulation runs. The average performance of our algorithm of the 8 runs is 90%. Compared with MEME (0.00) and Gibbs (12%), we can see that our algorithm significantly outperforms both the MEME and Gibbs algorithms for long DNA sequences.

IV. Conclusions

In this paper, we studied the problem of motif discoveries in unaligned DNA and protein sequences. We developed a self-organizing neural network structure for solving the problem of motif identification in DNA and protein sequences. Our network contains several layers with each layer performing classifications at different level. We maintain a low computational complexity through the use of the layered structure so that each pattern's classification is performed with respect to a small subspace of the whole input space. We also maintain a high reliability using our self-organizing neural network since it will grow as needed to make sure that all input patterns are considered and are given the same amount of attention. Simulation results show that our algorithm outperforms existing algorithms MEME and Gibbs in certain aspects. Our algorithm works well for long DNA sequences as well.

References

[1] T. L. Bailey and C. Elkan, "The value of prior knowledge in discovering motifs with MEME," *Proc. 3rd International Conference on Intelligent Systems for Molecular Biology*, Cambridge, UK, July 1995, pp. 21–29.

[2] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning,* vol. 21, no. 1–2, pp. 51–83, Oct./Nov. 1995.

[3] T. L. Bailey and M. Gribskov, "Combining evidence using p-values: Application to sequence homology searches," *Bioinformatics*, vol. 14, no. 1, pp. 48–54, Feb. 1998.

[4] A. Basu, P. Chaudhuri, and P. P. Majumder, "Identification of polymorphic motifs using probabilistic search algorithms," *Genome Research*, vol. 15, no. 1, pp. 67–77, Jan. 2005.

[5] K. Blekas, D. I. Fotiadis, and A. Likas, "A sequential method for discovering probabilistic motifs in proteins," *Methods of Information in Medicine*, vol. 43, no. 1, pp. 9–12, 2004.

[6] M. Boden and J. Hawkins, "Improved access to sequential motifs: a note on the architectural bias of recurrent networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 491–494, Mar. 2005.

[7] L. Brocchieri and S. Karlin, "A symmetric-iterated multiple alignment of protein sequences," *J. Molecular Biology,* vol. 276, no. 1, pp. 249–264, Feb. 1998.

[8] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern-recognition machine," *Computer Vision, Graphics, and Image Processing,* vol. 37, no. 1, pp. 54–115, Jan. 1987.

[9] G. A. Carpenter and S. Grossberg, "Search mechanisms for adaptive resonance theory (ART) architectures," *Proc. International Joint Conference on Neural Networks*, Washington, DC, June 1989, vol. 1, pp. 210–205.

[10] G. A. Carpenter and S. Grossberg, "ART-3: Hierarchical search using chemical transmitters in self-organizing pattern-recognition architectures," *Neural Networks,* vol. 3, no. 2, pp. 129–152, Mar. 1990.

[11] G. A. Carpenter and S. Grossberg, "A self-organizing neural network for supervised learning, recognition, and prediction," *IEEE Communications Magazine,* vol. 30, no. 9, pp. 38–49, Sept. 1992.

[12] G. A. Carpenter, S. Grossberg, and D. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Proc. International Joint Conference on Neural Networks*, Seattle, WA, July 1991, vol. 2, pp. 151–156.

[13] B. C. H. Chang, A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Particle swarm optimisation for protein motif discovery," *Genetic Programming and Evolvable Machines,* vol. 5, no. 2, pp. 203–214, June 2004.

[14] L. Duret and P. Bucher, "Searching for regulatory elements in human noncoding sequences," *Current Opinion in Structural Biology*, vol. 7, no. 3, pp. 399–406, June 1997.

[15] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, Oct. 1998.

[16] O. Emanuelsson, H. Nielsen, and G. von Heijne, "ChloroP, A neural network-based method for predicting chloroplast transit peptides and their cleavage sites," *Protein Science,* vol. 8, no. 5, pp. 978–984, May 1999.

[17] D. Frishman and P. Argos, "A neural network for recognizing distantly related protein sequences," in *Handbook of Neural Computation*, E. Fiesler and R. Beale, Eds., New York: IOP Publishing and Oxford University Press, pp. G4.4:1–8, 1997.

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

[19] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 10, pp. 1053–1074, Oct. 2001.

[20] S. Geva and J. Sitte, "Adaptive nearest neighbor pattern classification," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 318–322, Mar. 1991.

[21] P. Gonnet and F. Lisacek, "Probabilistic alignment of motifs with sequences," *Bioinformatics,* vol. 18, no. 8, pp. 1091–1101, Aug. 2002.

[22] W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker, "Meta-MEME: Motif-based hidden Markov models of protein families," *Computer Applications in the Biosciences*, vol. 13, no. 4, pp. 397–406, Aug. 1997.

[23] K. Gulukota, J. Sidney, A. Sette, and C. DeLisi, "Two complementary methods for predicting peptides binding major histocompatibility complex molecules," *J. Molecular Biology,* vol. 267, no. 5, pp. 1258–1267 Apr. 1997.

[24] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, New York: Cambridge University Press, 1997.

[25] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Upper Saddle River, NJ: Prentice Hall, pp. 443–483, 1999.

[26] 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, July/Aug. 1999.

[27] T. Hofmann and J. M. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 1–14, Jan. 1997.

[28] S. T. Jensen and J. S. Liu, "BioOptimizer: A Bayesian scoring function approach to motif discovery," *Bioinformatics*, vol. 20, no. 10, pp. 1557–1564, July 2004.

[29] E. Jeong, I. F. Chung, and S. Miyano, "A neural network method for identification of RNA-interacting residues in protein," *Proc. 15th International Conference on Genome Informatics*, Yokohama, Japan, Dec. 2004, pp. 105–116.

[30] F. Kanaya and S. Miyake, "Bayes statistical behavior and valid generalization of pattern classifying neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 4, pp. 471–475, July 1991.

[31] U. Keich and P. A. Pevzner, "Finding motifs in the twilight zone," *Bioinformatics,* vol. 18, no. 10, pp. 1374–1381, Oct. 2002.

[32] U. Keich and P. A. Pevzner, "Subtle motifs: Defining the limits of motif finding algorithms," *Bioinformatics,* vol. 18, no. 10, pp. 1382–1390, Oct. 2002.

[33] S. Keles, M. J. van der Laan, and C. Vulpe, "Regulatory motif finding by logic regression," *Bioinformatics,* vol. 20, no. 16, pp. 2799–2811, Nov. 2004.

[34] J. T. Kim, J. E. Gewehr, and T. Martinetz, "Binding matrix: A novel approach for binding site recognition," *J. Bioinformatics and Computational Biology,* vol. 2, no. 2, pp. 289–307, June 2004.

[35] S. Knudsen, "Promoter2.0: For the recognition of PolII promoter sequences," *Bioinformatics,* vol. 15, no. 5, pp. 356–361, May 1999.

[36] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling," *J. Molecular Biology*, vol. 235, no. 5, pp. 1501–1531, Feb. 1994.

[37] P. Lavoie, J.-F. Crespo, and Y. Savaria, "Generalization, discrimination, and multiple categorization using adaptive resonance theory," *IEEE Trans. Neural Networks,* vol. 10, no. 4, pp. 757–767, July 1999.

[38] 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.

[39] S. Liang, M. P. Samanta, and B. A. Biegel, "cWINNOWER algorithm for finding fuzzy DNA motifs," *J. Bioinformatics and Computational Biology*, vol. 2, no. 1, pp. 47–60, Mar. 2004.

[40] C. Linhart and R. Shamir, "The degenerate primer design problem," *Bioinformatics*, vol. 18, Suppl. 1, pp. S172–S180, 2002.

[41] Y. Liu, X. S. Liu, L. Wei, R. B. Altman, and S. Batzoglou, "Eukaryotic regulatory element conservation analysis and identification using comparative genomics," *Genome Research*, vol. 14, no. 3, pp. 451–458, Mar. 2004.

[42] A. M. Moses, D. Y. Chiang, and M. B. Eisen, "Phylogenetic motif detection by expectation-maximization on evolutionary mixtures," *Proc. Pacific Symposium on Biocomputing*, Fairmont Orchid, HI, Jan. 2004, pp. 324–335.

[43] A. F. Neuwald, J. S. Liu, D. J. Lipman, and C. E. Lawrence, "Extracting protein alignment models from the sequence database," *Nucleic Acids Research,* vol. 25, no. 9, pp. 1665–1677, May 1997.

[44] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne, "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites," *Protein Engineering,* vol. 10, no. 1, pp. 1–6, Jan. 1997.

[45] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne, "A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites," *International J. Neural Systems,* vol. 8, no. 5—6, pp. 581–599, Oct./Dec. 1997.

[46] A. R. Ortiz, A. Kolinski, and J. Skolnick, "Nativelike topology assembly of small proteins using predicted restraints in Monte Carlo folding simulations," *Proc. National Academy of Sciences of the USA,* vol. 95, no. 3, pp. 1020–1025, Feb. 1998.

[47] B. Padmanabhan and A. Tuzhilin, "Pattern discovery in temporal databases: A temporal logic approach," *Proc. Second International Conference on Knowledge Discovery and Data Mining,* Portland, Oregon, Aug. 1996, pp. 351–354.

[48] P. A. Pevzner and S.-H. Sze, "Combinatorial approaches to finding subtle signals in DNA sequences," *Proc. 8th International Conference on Intelligent Systems for Molecular Biology*, San Diego, CA, Aug. 2000, pp. 269–278.

[49] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[50] W. R. Taylor and D. T. Jones, "Deriving an amino acid distance matrix," *J. Theoretical Biology,* vol. 164, no. 1, pp. 65–83, Sept. 1993.

[51] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTALW: 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, Nov. 1994.

[52] G. White and W. Seffens, "Using a neural network to backtranslate amino acid sequences," *Electronic J. Biotechnology [online]*, Dec. 1998, Vol. 1, No. 3. Available from: http://www.ejbiotechnology.info/content/vol1/issue3/full/5/index.html. ISSN 0717-3458

[53] C. T. Workman and G. D. Stormo, "ANN-Spec: A method for discovering transcription factor binding sites with improved specificity," *Proc. Pacific Symposium on Biocomputing,* Honolulu, HI, Jan. 2000, pp. 467–478.

[54] C. Wu, S. Shivakumar, H. P. Lin, S. Veldurti, and Y. Bhatikar, "Neural networks for molecular sequence classification," *Mathematics and Computers in Simulation,* vol. 40, no. 1–2, pp. 23–33, Dec. 1995.

[55] J. Xie, K. C. Li, and M. Bina, "A Bayesian insertion/deletion algorithm for distant protein motif searching via entropy filtering," *J. the American Statistical Association,* vol. 99, no. 466, pp. 409–420, June 2004.

[56] E. P. Xing, W. Wu, M. I. Jordan, and R. M. Karp, "Logos: A modular Bayesian model for de novo motif detection," *J. Bioinformatics and Computational Biology,* vol. 2, no. 1, pp. 127–154, Mar. 2004.

[57] R. Y. Zheng and R. Thomson, "Bio-basis function neural network for prediction of protease cleavage sites in proteins," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 263–274, Jan. 2005.

[58] http://pbil.univ-lyon1.fr/acuts/ACUTS_home.html.

Derong Liu (S'91-M'94-SM'96-F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, Indiana, in 1994; the M.S. degree in electrical engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1987; and the B.S. degree in mechanical engineering from the East China Institute of Technology (now Nanjing University of Science and Technology), Nanjing, China, in 1982. From 1982 to 1984, he was a product design engineer at China North Industries Corporation, Jilin, China. From 1987 to 1990, he was an instructor at the Graduate School of the Chinese Academy of Sciences, Beijing, China. From 1993 to 1995, he was a staff fellow at General Motors Research and Development Center, Warren, Michigan. From 1995 to 1999, he was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, New Jersey. He joined the University of Illinois at Chicago in 1999 where he is now an Associate Professor of Electrical and Computer Engineering, of Bioengineering, and of Computer Science. Since 2005, he serves as the Director of Graduate Studies in the Department of Electrical and Computer Engineering at the University of Illinois at Chicago. He is coauthor (with A. N. Michel) of the books *Dynamical Systems with Saturation Nonlinearities: Analysis and Design* (New York: Springer-Verlag, 1994) and *Qualitative Analysis and Synthesis of Recurrent Neural Networks* (New York: Marcel Dekker, 2002). He is coeditor (with P. J. Antsaklis) of the book *Stability and Control of Dynamical Systems with Applications* (Boston, MA: Birkhauser, 2003).

Dr. Liu was a member of the Conference Editorial Board of the IEEE Control Systems Society (1995–2000); and served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: FUNDAMENTAL THEORY AND APPLICATIONS (1997–1999), the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2001–2003), and the IEEE TRANSACTIONS ON NEURAL NETWORKS (2004–2006). Since 2004, he has been the Editor of the IEEE Computational Intelligence Society's Electronic Letter; and since 2006, he has been the Letter Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, an Associate Editor of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, and an Associate Editor of the Automatica. He is the Program Chair for the following three conferences: the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning; the 21st IEEE International Symposium on Intelligent Control (2006); and the 2006 International Conference on Networking, Sensing and Control. He has served and is serving as a member of the organizing committee and the program committee of several international conferences. He is an elected AdCom member of the IEEE Computational Intelligence Society (2006–2009) and he is the Chair of the Chicago Chapter of the IEEE Computational Intelligence Society. He was recipient of the Michael J. Birck Fellowship from the University of Notre Dame (1990), the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology (1997), and the Faculty Early Career Development (CAREER) award from the National Science Foundation (1999). He is a Fellow of the IEEE and a member of Eta Kappa Nu.

Xiaoxu Xiong (S'03) is currently a Ph.D. student in electrical and computer engineering at the University of Illinois at Chicago. He received the M.S. degree and B.S. degree in electrical engineering from Harbin Engineering University, Harbin, China, in 2000 and 1998, respectively. His current research interests include artificial intelligence, pattern recognition and bioinformatics. From 1998 to 2000, he was a software engineer in the Hitech Special Software Institute, Harbin, China. He served as a web administrator for the 2006 IEEE International Conference on Networking, Sensing and Control.

Bhaskar DasGupta (M'96-SM'01) is currently an associate professor in the Computer Science Department at University of Illinois at Chicago (UIC) and also affiliated with the Bioengineering Department at UIC. He did his PhD from University of Minnesota in 1995, was a post-doctoral fellow at DIMACS and jointly at University of Waterloo and McMaster University in Canada before he joined the computer science department of camden campus of Rutgers University; in 2001 he moved to UIC. His research specific research interests include application of combinatorial/geometric techniques to design efficient algorithms for computational problems in bioinformatics, systems biology and hybrid systems; his broader research interests include designing efficient combinatorial algorithms for computationally hard problems in diverse areas in addition to bioinformatics such as computational geometry, VLSI/CAD, parallel computing, optical networks, and combinatorial auctions. His research works have been supported by numerous NSF grants, including an NSF CAREER award.

Are you a senior member??

Huaguang Zhang (SM'04) was born in Jilin, China, in 1959. He received the B.S degree and M.S degree in control engineering from Northeastern Electric Power University of China in 1982 and 1985, respectively. He received the Ph. D degree in thermal power engineering and automation from Southeastern University of China in 1991. He entered automatic control department, Northeastern University, in Jan. 1992, as a postdoctoral fellow for two years. Since 1994, he has been a professor and Head of the Institute of Electric Automation, Northeastern University, Shenyang, China. His main research interests are fuzzy control, chaos control, neural networks based control, nonlinear control, signal processing, and their applications.

Dr. Zhang was awarded the "Excellent Youth Science Foundation Award" by the China Natural Science Foundation in 2003. He was named the "Changjiang Scholar" by China Education Ministry in 2005.