Promoter classification using genetic algorithm controlled generalized regression neural network

Kun-ho Kim and Byun-gwhan Kim[*], Kyung-nam Kim[**], Jin-Han Hong and Sang-Ho Park[***]

[*] Department of Electronic Engineering, Sejong University, 98 Kunja Dong, Kwangjin Ku, Seoul, 143-747, Korea
(Tel : 82-2-3408-3729; Fax : 82-2-3408-3329 ; E-mail : kbwhan@sejong.ac.kr)
[**]Department of Molecular Biology, Sejong University, 98, Kunja-Dong, Kwangjin-Ku,Seoul, 143-747, Korea.
[***]DNA Chip Division, Macrogen,116, Shinmun-Ro 1 Ka, Chongro-Ku, Seoul, Korea.

Abstract: A new method is presented to construct a classifier. This was accomplished by combining a generalized regression neural network (GRNN) and a genetic algorithm (GA). The classifier constructed in this way is referred to as a GA-GRNN. The GA played a role of controlling training factors simultaneously. In GA optimization, neuron spreads were represented in a chromosome. The proposed optimization method was applied to a data set, consisted of 4 different promoter sequences. The training and test data were composed of 115 and 58 sequence patterns, respectively. The range of neuron spreads was experimentally varied from 0.4 to 1.4 with an increment of 0.1. The GA-GRNN was compared to a conventional GRNN. The classifier performance was investigated in terms of the classification sensitivity and prediction accuracy. The GA-GRNN significantly improved the total classification sensitivity compared to the conventional GRNN. Also, the GA-GRNN demonstrated an improvement of about 10.1% in the total prediction accuracy. As a result, the proposed GA-GRNN illustrated improved classification sensitivity and prediction accuracy over the conventional GRNN.

Keywords: Promoter, Genetic algorithm, Generalized regression neural network, Classification, Optimization.

1. INTRODUCTION

As a biometric, artificial neural network (ANN) has been extensively applied to map and identify specific biological functions in DNA, RNA, and protein sequences [1-3]. Compared to other algorithms, ANN demonstrated superior functional mapping ability. This is mainly attributed to the ANN capability of high correlation and interpolation. Many different types of neural networks have been applied to predicting DNA sequences. These include an adaptive resonance theory-based network, backpropagation neural network, or counter-propagation network. Another paradigm that might be effectively used is a GRNN [4]. Its application to sequence analysis has been little reported. The GRNN performance depends on one training factor called ‘spread’ of the gaussian function in the pattern layer. Conventionally, the spread effect is optimized by experimentally adjusting the spread. Most critical problem is that all neurons in the pattern layer are quipped with one single, optimized spread. By adopting multi-spreads, it is expected that the GRNN predictive ability could be improved.

In this study, a method to construct a GRNN classifier of multi-valued spreads is presented. This is accomplished by applying a genetic algorithm (GA). The GA is used to search for a set of optimized spreads. For convenience, the GA-controlled GRNN is called “GA-GRNN”. The proposed GA-GRNN is applied to classify 4 promoters. The performance is evaluated in terms of the prediction accuracy and the classification sensitivity. This is conducted for all or individual set of promoter sequences. The GA-GRNN is also compared to conventional GRNN.

2. EXPERIMENTAL DATA

The DNA data evaluated consist of 4 types of promoters, including *Oriza Sativa* (OS), *Arabidopsis Thaliana* (AT), *Escherichia Coli* (EC), and *Zymomonas Mobils* (ZM). The first two promoters, OS and AT, can be classified into an eukaryotic promoter. The other EC and ZM belong to prokaryotic promoter. Promoter sequences for AT were obtained by comparing full-length cDNAs [5] with a genomic DNA [6]. Since DNA sequences upstream of the cDNAs contain the promoter activity, approximately 1-kb genomic DNA regions upstream of the translation start site (ATG codon) were selected in constructing the database. The OS promoter sequences were collected in the similar way using the rice database [7]. Meanwhile, the whole genome sequences of two bacterial species, the EC [8] and ZM were obtained from NCBI with accession number U00096 and in-house database of Macrogen, respectively. The open reading frames (ORFs) from ZM were derived from the prediction by using a program 'Glimmer V2.0' [9] and analyzed further with a BlastX [10] program with non-redundant protein database of NCBI. For the two sets of genomic data, a number of promoter sequence were collected by searching promoters, and each sequence consisted of 500 bases upstream and 100 bases downstream from the cordon start site.

The training data consist of 115 sets of promoter sequences. More specifically, the data is composed of 20 OS, 25 AT, 35 ET, and 35 ZM. The test data for evaluating model appropriateness are composed of 58 sets of promoters, 13 OS, 15 AT, 15 ET, and 15 ZM. Each sequence pattern consisted of 146 base pairs.

3. GENERALIZED REGRESSION NEURAL NETWORK

A schematic of GRNN is depicted in Fig. 1. As shown in Fig. 1, the GRNN consists of four layers, including the input layer, pattern layer, summation layer, and output layer. Each input unit in the first layer corresponds to individual process parameter. The first layer is fully connected to the second, pattern layer, where each unit represents a training pattern and its output is a measure of the distance of the input from the stored patterns. Each pattern layer unit is connected to the two neurons in the summation layer: S-summation neuron and D-summation neuron. The S-summation neuron computes the sum of the weighted outputs of the pattern layer while the D-summation neuron calculates the unweighted outputs of the pattern neurons.

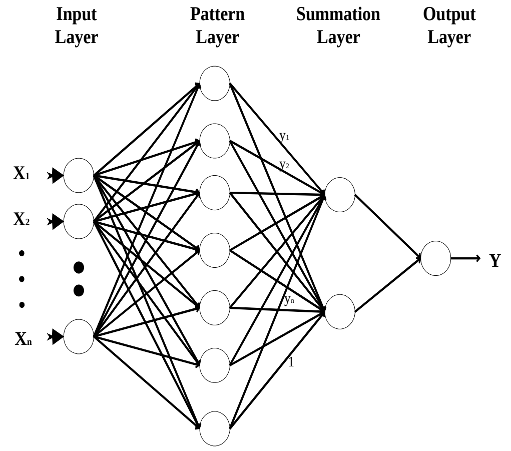

Fig. 1 Schematic of generalized regression neural network.

The connection weight between the ith neuron in the pattern layer and the S-summation neuron is y_i, the target output value corresponding to the ith input pattern. For D-summation neuron, the connection weight is unity. The output layer merely divides the output of each S-summation neuron by that of each D-summation neuron, yielding the predicted value to an unknown input vector x as

$$\hat{y}_i(x) = \frac{\sum_{i=1}^{n} y_i \exp[-D(x,x_i)]}{\sum_{i=1}^{n} \exp[-D(x,x_i)]} \qquad (1)$$

where n indicates the number of training patterns and the D function in (1) is defined as

$$D(x,x_i) = \sum_{j=1}^{p} (\frac{x_j - x_{i_j}}{\zeta})^2 \qquad (2)$$

where P indicates the number of elements of an input vector. The x_j and x_{i_j} represent the jth element of x and x_i, respectively. The ζ is generally referred to as the spread, whose optimal value is conventionally determined by adjusting it within certain experimental range.

4. RESULTS

The performance of classifier is evaluated in terms of the prediction accuracy and classification sensitivity. The prediction accuracy is measured by the root mean-squared error (RMSE) metric defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{p}\sum_{j=1}^{q}(d_{ij} - out_{ij})^2}{pq}} \qquad (3)$$

where p and q represent the number of output neurons and test patterns, respectively. The d_{ij} and out_{ij} represent the desired and calculated outputs of the ith output neuron for the jth test pattern. The prediction accuracies calculated for all data sets and individual data set are referred to as the total prediction accuracy (TPA) and individual prediction accuracy (IPA). The other classification sensitivity is defined as the total number of the test sequence patterns correctly classified into their respective classes. The classification sensitivity is evaluated as a function of the threshold expressed as

$$\left| d_{ij} - out_{ij} \right| \langle \text{ Threshold} \qquad (4)$$

Similarly as in the case of the prediction accuracy, the classification sensitivity is measured for all and individual data sets, each called TCS and ICS, respectively. Meanwhile, the threshold is set to 0.9 to examine classifier performance in the most stringent situation.

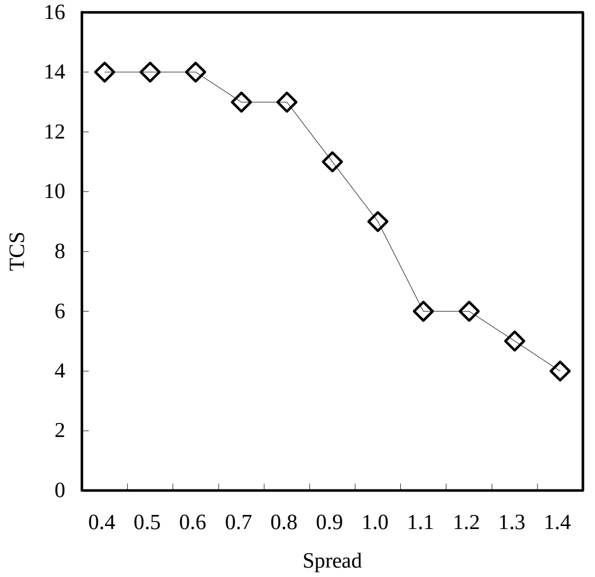

Fig. 2 Total classification sensitivity of GRNN as a function of spread.

4.1. Conventional GRNN

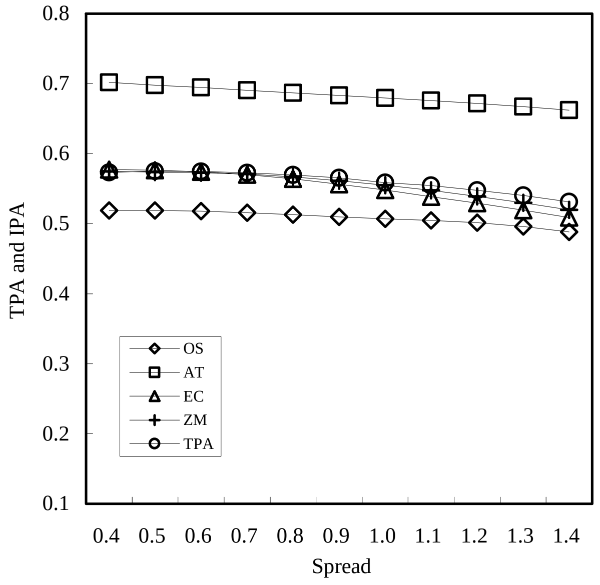

Fig. 3 Prediction accuracy of GRNN as a function of spread.

For comparison, the performance of conventional GRNN is first investigated. The spread varied incrementally from 0.4 to 1.4 by 0.1. For each spread, GRNN classifier was constructed. The TCS measured by (4) is displayed in Fig. 2 as a function of the spread. As depicted in Fig. 2, the TCS generally decreases with increasing the spread. The highest TCS of 14 is obtained at 0.4, 0.5, and 0.6. Fig. 3 shows the prediction behavior of the classifier for the same spread range. Both TPA and IPA decrease with increasing the spread. The IPA for AT is largest compared to others. As observed in Fig. 3, the smallest TPA and IPAs are obtained at 1.4. Then, the ICS of the classifier with both TCS and TPA optimized is examined. Comparing Fig. 2 and Fig. 3 reveals that both TCS and TPA seem to be optimized at 0.4, 0.5, and 0.6. To determine one classifier, the corresponding TPAs were calculated, and the smallest TPA of 0.573 was obtained at 0.4. The ICSs of the TCS for the classifier optimized at 0.4 are 6 OS, 0 AT, 4 EC, and 4 ZM. This reveals that the GRNN is incapable of classifying the AT. This is expected from the largest RMSE as noticed earlier.

4.2 GA-GRNN

The GA was utilized to search for a particular factor setting that minimizes the prediction accuracy. In GA optimization, each training factor was coded in a real value and this resulted in a total chromosome length of 115 bits. During each computational cycle, an initial population of 100 potential solutions was created with each manipulated by the genetic operators. Next, the performance of each individual of the population is evaluated and a selection mechanism is subsequently activated to choose the best string with the highest fitness for the genetic manipulation process. The crossover operator takes two chromosomes and parts of their genetic information are swapped to produce two new chromosomes based on a specified crossover probability. Another mutation probability is given to the mutation operator, which randomly changes a fixed number of bits every generation. Here, those numerical probabilities of crossover and mutation used in this optimization are 0.9 and 0.1, respectively. A particular input setting generated by GA meets a given fitness function expressed as:

$$F = 1 + \sum_r CS \qquad (5)$$

where *CS* is the calculated classification sensitivity and r is the number of promoter attributes. Since each attribute was optimized individually, the r is equal to unity. The GA was then implemented with setting the *CS* in Eq. (5) to zero for each promoter attribute. As a termination criterion, the generation number was set to 100.

The performance of GA-GRNN is examined as a function of the spread. The experimental range of the spread is same as that employed in constructing GRNN classifier. For a given spread, the chromosomes were initially randomized within that value and one best GA-GRNN is determined as the generation is completed at 100. The TCSs of the best GA-GRNN are plotted in Fig. 4. As depicted in Fig. 4, the TCS initially decreases and then seems to remain constant at larger spreads of more than 1.1. One optimal classifier with the highest TCS is obtained at two spreads, 0.4 and 0.5. The corresponding TCS is 21. Compared to the optimized GRNN, the GA-GRNN considerably improved the TCS by 7.

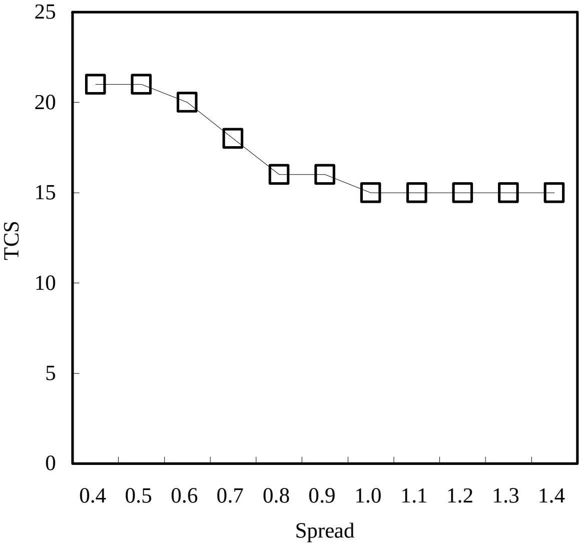

Fig. 4 Total classification sensitivity of GA-GRNN as a function of range of random spread.

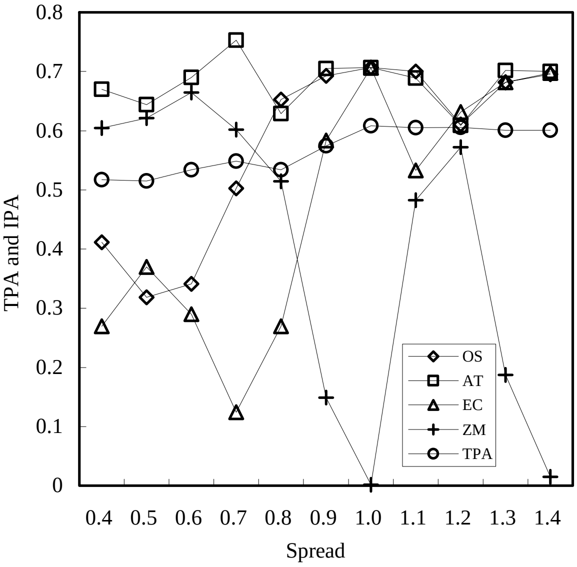

Fig. 5 Prediction accuracy of GA-GRNN as a function of spread.

Fig. 5 shows the prediction behavior of GA-GRNN as a function of the spread. Both TPA and IPAs are included. As illustrated in Fig. 5, each IPA behavior with the spread is quite complex. In contrast, the TPA appears to increase consistently with the spread. This is in contrast to what was observed for the GRNN in Fig. 2. The smallest TPA of 0.515 is obtained at 0.5. Compared to that (0.573) for GRNN, this demonstrates of about 10.1% improvements. The ICSs of the selected GA-GRNN at 0.5 are 6 OS, 1 AT, 11 EC, and 3 ZM, respectively. Compared to those for the optimized GRNN, the GA-GRNN drastically improved the ICS by 7 for EC. Consequently, the GA-GRNN demonstrated much improved TCS along with better ICS on average. In Table I, IPAs of GRNN and GA-GRNN classifiers optimized at 0.4 and 0.5 respectively are compared. As represented in Table I, the GA-GRNN improved all IPAs but the case of ZM. The improvement is considerable in the two cases of OS and EC.

As a result, the proposed GA-GRNN illustrated improved TPA and TCS. On average, this was demonstrated even in either ICS or IPA.

Table 1 Comparison of IPA of GRNN and GA-GRNN

Promoter type	GRNN	GA-GRNN	Improvement(%)
OS	0.518	0.318	38.6
AT	0.701	0.644	8.1
EC	0.577	0.370	35.8
ZM	0.574	0.620	-8.0

5. CONCLUSIONS

Using the GA, a GRNN classifier was constructed and applied to classify DNA promoter sequences. The GA was used to optimize spreads for the gaussian functions in the pattern layer. The GA-GRNN was compared to conventional GRNN in terms of the classification sensitivity and prediction accuracy. Comparisons revealed that the GA-GRNN was much better than conventional GRNN in classifying and prediction accuracy. Particularly, the improvement was significant in the TCS. The average performance of GA-GRNN was better than that for GRNN in classifying promoters individually. The proposed classifier is very simple to implement and optimize. By the demonstrated high classification capability, the GA-GRNN is expected widely used for predicting or classifying large volume of other bio-medical data.

ACKNOWLEDGEMENTS

This work was supported by Korea Health Industry Development Institute from 2000 IMT Fund.

REFERENCES

[1] M. V. Gils, H. Jansen, K. Nieminen, R. Summers, P. R. Weller, "Using artificial neural networks for classifying ICU patient states," *IEEE EMB Mag*., pp. 41-47, 1997.

[2] S. Knudsen, "Promoter 2.0: for the recognition of Pol II promoter sequences," *Bioinformatics*, vol. 15, pp. 356-361, 1999.

[3] S. Matis, Y. Xu, M. Shah, X. Guan, J. R. Einstein, R. Mural, E. Uberhacher, "Detection of RNA polymerase II promoters and polyadenylation sites in human DNA sequence." *Comp. Chem.* pp. 135-140, 1996.

[4] Specht D F, "A generalized regression neural networks." *IEEE Trans. Neural Networks* vol. 2, pp. 568-576, 1991.

[5] http://signal.salk.edu/cgi-bin/tdnaexpress.

[6] http://arabidopsis.org.

[7] http://www.ncbi.nlm.nih.gov.

[8] F. R. Blattner, G. III Plunket, C. A. Bloc, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mayhew, J. Gregor, N. W. Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau, and Y. Shao, "The complete genome sequence of Escherichia coli K-12," *Science*, vol. 277, pp. 1453-1474, 1997.

[9] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with GLIMMER," *Nucleic Acids Res*., vol. 27, pp. 4636-4641, 1999.

[10] W. Gish and D. J. States, "Identification of protein coding regions by database similarity search," *Nature Genetic*, vol. 3, pp. 266-272, 1993.