35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

A Hybrid Model for the Prediction of mRNA Polyadenylation Signals

Jiuqiang Han, *Member, IEEE*, Ze Liu, Dexing Zhong*, *Member, IEEE*, and Tuo Wang

***Abstract*— The mRNA polyadenylation is the cellular process that adds adenosine tails to mature mRNAs. Malfunction of polyadenylation has been implicated in several human diseases. In this paper, we proposed a novel feature extraction approach which employs the K-gram nucleotide acid pattern, the position weight matrix (PWM) and the increment of diversity (ID) to represent the original features. Then Principle Component Analysis (PCA) was applied to transform the original features into a new feature space where the low-dimensional features were used to train the real-coded genetic neural network model. In the experiments, our proposed algorithm (GA-BP) can achieve the accuracy about 82.98%, specificity 82.95% and sensitivity 83.01% in the specific dataset constructed by Kalkatawi. The results demonstrate that GA-BP is a promising algorithm for the prediction of mRNA polyadenylation signals.**

I. Introduction

Besides some histone mRNAs, all eukaryotic mRNAs possess polyadenine (poly(A)) tails at their 3' end, poly(A) tails have strong influences on the cellular metabolism of mRNA, *e.g.* mRNA stability, translation and transportation from the nucleus to the cytoplasm. The cellular process of adding poly(A) tails to mRNAs, called polyadenylation, composed of two tightly coupled steps: an endonucleolytic cleavage of pre-mRNA and followed by adding the adenosine tail to the newly formed 3' end. Generally, it is recognized that the polyadenylation signals for almost all eukaryotes seem to contain two core elements (shown in Fig. 1): the poly(A) signal (PAS), which always assumes in the form of AAUAAA hexamer or a close variant, serves as the binding site for the cleavage and polyadenylation specificity factor (CPSF) and locates between 10 and 30 *nt* upstream of the actual cleavage site. The second canonical sequence locates between 20 and 40 *nt* downstream of the cleavage site, which is usually characterized as U-rich or GU-rich region, bounded by the 64-kDa subunit of the heterotrimeric cleavage-stimulating factor (CstF) that promotes the efficiency of 3' end processing. Furthermore, a number of auxiliary elements upstream or downstream of cleavage sites also play regulatory roles in polyadenylation in both viral and cellular systems [1-3].

Prediction of poly(A) signals has drawn much attention in recent years and various statistical characteristics of sequences flanking poly(A) sites were calculated. Salamov and Solovyev designed a program POLYAH, based on Linear Discriminant Function (LDF), for predicting poly(A) motifs [5]. In order to avoid training on possibly flawed data, the program Polyadq began with a *de novo* characterization of poly(A) motifs [6]. The character was used in training two quadratic discriminant functions for the prediction of poly(A) signals. The program named Erpin, introduced by Legendre and Gautheret, utilized 2-gram position-specific nucleotide acid patterns to analyze the -300- +300 *nt* sequences flanking the candidate PAS [7]. A generalized hidden markov model (GHMM) was applied to extract sequence characteristics in C.elegans 3' end [8]. In 2006, Cheng *et al.* presented a method based on support vector machine by using 15 cis-regulatory elements, obtained by the program PROBE [9-10]. In 2009, Akhtar *et al.* developed a program called POLYAR classified the polyadenylation sites into three classes (PAS-strong, PAS-weak and PAS-less, respectively) [11]. In 2012, Kalkatawi *et al.* developed a program called Dragon PolyA Spotter based on artificial neural networks (ANNs) and random forest (RF). Dragon PolyA Spotter achieved higher sensitivity and specificity than other predictors by using thermodynamic, physico-chemical and statistical characteristics of -100 *nt* to +100 *nt* genomic sequences around the PAS [12].

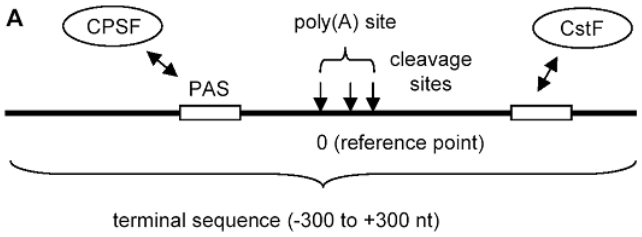

Figure 1. Schematic representation of poly(A) site and polyadenylation configuration [4].

Although current tools have achieved moderate sensitivity and specificity, there is still room for improvements. To date, the optimization of neural network has drawn much attention in recent years and much work has been fulfilled on training neural network by using genetic algorithms. Quite a number of these hybrid models have been successfully employed to overcome the inherent limitations of the back propagation (BP) neural network [13-16]. But seldom work has been done on applying the hybrid models for the prediction of poly(A) signals. Thus, we hybridized real-coded genetic algorithm (GA) with the BP neural network. The principle component analysis (PCA) was adopted to reduce the dimension of the feature space. Compared with ANNs trained by BP algorithm, our model achieves higher sensitivity and specificity. Furthermore, our algorithm also reduces the possibility of trapping at a local optimum.

This work was supported by grants from National Natural Science Foundation of China (No. 61105021, No. 61071217), Ph.D. Program Foundation of the Ministry of Education of China (No. 20110201110010), China Postdoctoral Science Foundation (No. 2011M501442) and the Fundamental Research Funds for the Central Universities.

J. Han (email: jqhan@xjtu.edu.cn), Z. Liu (my_melody@stu.xjtu.edu.cn), D. Zhong (bell@xjtu.edu.cn) and T. Wang (tuowang@xjtu.edu.cn) are with the Ministry of Education Key Lab for Intelligent Networks and Network Security, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China.

*Corresponding author, phone: +86-029-82668665, bell@xjtu.edu.cn.

II. METHODS

A. Datasets

In this paper, we trained and tested our classification model by the data sets constructed by Kalkatawi *et al*. [12]. We selected 5,000 sequences both from the negative AATAAA and positive AATAAA folder randomly. Then we truncated all sequences flanking AATAAA motif by 100 nucleotides upstream and 100 nucleotides downstream. Furthermore, each sequence contains 200 bases as the AATAAA motifs of all those sequences were deleted. The training data set, in which the positive and negative sequences are distributed evenly, contains 5,000 sequences. And the testing data set was built the same as the training data set.

B. Feature Generation

K-gram nucleotide acid pattern is simply a pattern of *k* (*k*=1, 2, 3…) consecutive letters. For instance, CA is a 2-gram nucleotide acid pattern and CAG is a 3-gram nucleotide acid pattern. The length *L* of the scanning region must be determined at first, and then the *k*-gram pattern is scanned one by one from the first position to the *L*-*k*-1 position of the scanning region. At last, the relative frequency of each *k*-gram pattern is calculated respectively. In this paper, the nucleotide frequencies of *k*-gram (*k*=1, 2, 3) were calculated after extraction from both upstream and downstream regions. In addition to these, we also used the frequency of T nucleotide in the upstream and downstream regions, as well as G nucleotide frequency in the downstream region of poly(A) signal [12], Thus, total 87(=4+16+64+1+1+1) features were generated.

Position weight matrices (PWMs) have been widely used to describe the base conservative level near poly(A) sites [5]. To construct PWMs, the upstream and downstream regions of poly(A) sites were divided into 19 pieces respectively. Each piece has 10 nucleotides with overlap of 5 nucleotides, *i.e.* 0~+10, 5~+15, … The PWMs used 1-gram frequencies were calculated for both positive and negative samples. In this way, 76 features were extracted.

Increment of diversity (ID) can be used to measure the diversity of two data resources [17-18]. We introduced the scalar ID

$$ID(X,Y) = D(X+Y) - D(X) - D(Y) = D(N,M) - \sum_{i=1}^{s} D(n_i, m_i). \quad (1)$$

Then, it follows that

$$D(N,M) = (M+N)\log_b(M+N) - M\log_b M - N\log_b N \quad (2)$$

and

$$D(n_i,m_i) = (m_i+n_i)\log_b(m_i+n_i) - m_i\log_b m_i - n_i\log_b n_i. \quad (3)$$

Where $X:\{n_1, n_2, ..., n_s\}$ and $Y:\{m_1, m_2, ..., m_s\}$ are two discrete resources, n_i and m_i are the absolute frequency of the i^{th} state of *X*, *Y*, respectively, and *M*, *N* are shown in (4).

$$N = \sum_{i=1}^{s} n_i, M = \sum_{i=1}^{s} m_i \quad (4)$$

We set each sequence both in the training and testing data sets as the X and set the positive and negative sequences of the training data set as the Y separately. Among them, n_i , m_i were the absolute frequencies of 3-gram nucleotide acid patterns.

C. Principle Component Analysis

As the dimension of the feature space is large, the components of the feature space may be highly correlated. Thus, we employed the method of PCA to solve this problem. PCA is a quantitatively rigorous method for reducing the dimension *n* of the input space. The process of PCA is composed of four steps: In the first step, subtract the mean from each dimensions of the original feature space. In the second step, the covariance matrix is calculated by the matrix derived from the data generated by the first step. In the third step, the eigenvalue of each eigenvector of covariance matrix is calculated and saved in descending order. At last, project the original feature space to the new space calculated by covariance matrix. The maximum amount of information is transmitted by the eigenvectors of the first *m* larger principal components.

D. Methods and steps combining GA with BP

Training neural network with BP algorithm has good local searching ability. However, it is easy to fall into the local optima and even more sophisticated quasi-Newton methods such as Levenberg-Marquardt offer little improvement. As it is shown in [19], 3-layer neural network can approximate any complex nonlinear function. So the neural network used in this paper has only 3 layers. The tangent function was adopted to transfer the values from the input layer to the hidden layer, whereas the values from the hidden layer to the output layer were transferred by the linear transfer function. Furthermore, the output layer had two neurons which represent the positive and negative sequence respectively. In contrast to BP algorithm which searches the weight space from one point to another, the genetic algorithm moves from one set of weights to another set. As the algorithm simultaneously searched in many directions, the search was directed toward the area of the best solution. Thus, hybridizing GA with BP for neural network training can take advantages of each algorithm: the GA (with its global search) determines a sub-optimal weight space, and BP (with its local search) seeks the best solution in the area of the weight spaces found by the GA.

The GA-BP algorithm is difficult to reach convergence due to there are too many design variables in the feature space. Thus, a real-coded genetic algorithm was applied to solve the problem. The algorithm combining real-coded GA with BP neural network mainly had three steps: firstly, a population of "individuals" was generated and the fitness of each individual was measured by calculating the value of total mean square error. The evaluation rule of individuals was "LOWS-BEST". Secondly, selection, crossover and mutation operations were applied according to the fitness of each individual. Individuals were selected on the basis of their relative fitness after ranking in the roulette wheel operator and saved as the intermediate

population. Thirdly, arithmetic crossover and non-uniform mutation were applied to the intermediate population. In this way, the population of the next generation was generated. The evolution stopped when the iteration satisfied the predefined termination criterion. At last, we set the optimal connection weights and bias to the neurons of the neural network and used BP algorithm to adjust the final weights and bias.

III. Results

In order to test the performance of our model, firstly we generated a feature space by using the data sets mentioned in section 2. Thus, 165(=87+76+2) features were created in our original feature space. As the dimension of the original feature space was too large and certain variables offered little information for the prediction, we processed the original features with PCA. After carefully experiments, compared with the original algorithm without using PCA, the parameters (SN, SP, ACC) almost remain the same. Features that contributed less than 1% to the total variation in the data set were eliminated and 24 orthogonal features were generated in the final feature space. The real-coded genetic algorithm was operated with a population size of 50. Uniform crossover and mutation probabilities were set as 0.9 and 0.1 respectively. In order to find the global optimum, the evolutional generation of the GA was set as 50 followed by a BP training procedure. The learning coefficient of BP training algorithm was set as 0.01.

Table 1 shows the comparison between the GA-BP and BP neural networks: TP, FP, TN, FN are the numbers of true positives, false positives, true negatives and false negatives, respectively. SN is the sensitivity, and SN=TP/(TP+FN). SP is the specificity, and SP=TN/(TN+FP). ACC is the accuracy, and ACC= (TP+TN)/(TP+FP+TN+FN)×100%. It could be seen that the prediction performance of GA-BP algorithm is higher than that of BP algorithm. Besides, the receiver operating characteristics (ROC) curves of these two methods are shown in Fig. 3. ROC plot of GA-BP algorithm indicates the high accuracy of this model.

TABLE I. Comparison Between GA-BP And BP Neural Networks

	BP	GA-BP
Sensitivity (SN)	82.03%	83.01%
Specificity (SP)	80.33%	82.95%
Accuracy (ACC)	81.18%	82.98%

IV. Discussion

In this paper, we has proposed a GA-BP model for the prediction of human poly(A) signals. Using PCA, the dimension of the feature space was effectively reduced at beginning. The performance of prediction by the GA-BP model is better than the one of the standard BP neural network. The experiments support the hypothesis that the GA-BP model can take both advantages of the two algorithms.

The GA strategy was only used to optimize the initial network weights and the network structure was determined by experience. Further work could focus on the optimization of neural network structure with GA strategy for the prediction of ploy(A) signals .

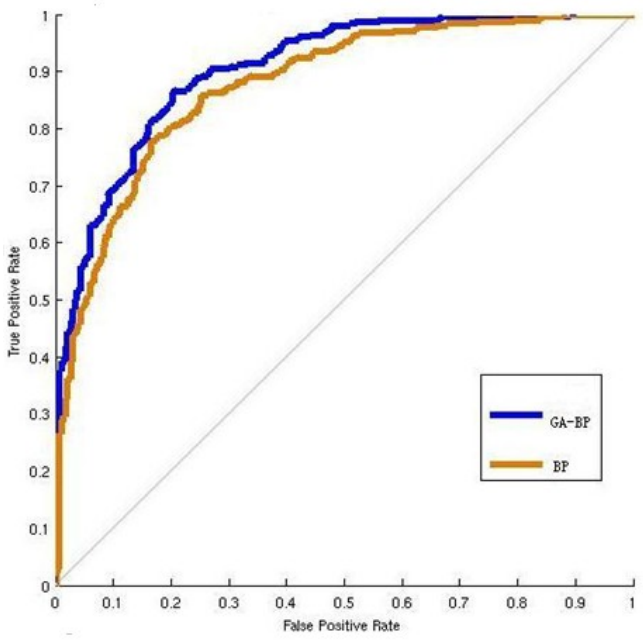

Figure 2. The receiver operating characteristics (ROC) curves of GA-BP and BP neural networks

References

[1] N.J. Proudfoot, A. Furger and M.J. Dye, "Integrating mRNA Processing with Transcription," Cell, vol. 108, pp.501-512, 2002.

[2] S. Danckwardt, M.W. Hentze and A.E. Kulozik, "3'end mRNA processing: molecular mechanisms and implications for health and disease," European Molecular Biology Organization, vol. 27, pp.482-498, 2008.

[3] G. Edwalds-Gilbert, K.L. Veraldi and C. Milcarek,"Alternative poly(A) site selection in complex transcription units: mean to an end?," Nucleic Acids Research, vol. 25, pp.2537-2561, 1997.

[4] B. Tian, J. Hu, H.B. Zhang and C.S. Lutz, "A large-scale analysis of mRNA polyadenylation of human and mouse genes," vol. 33, pp.201-212, 2005.

[5] A.A. Salamov, V.V. Solovyev, "Recognition of 3'-processing sites of human mRNA precursors," CABIOS, vol. 13, pp.23-28, 1997.

[6] J.E. Tabaska, M.Q. Zhang, "Detection of polyadenylation signals in human DNA sequences," Gene, vol. 231, pp.77-86, 1999.

[7] M. Legendre, D. Gautheret, "Sequence determinants in human polyadenylation site selection," BMC Genomics, vol. 4(7), 2003.

[8] A. Hajarnavis, I. Korf and R. Durbin, "A probabilistic model of 3' end formation in Caenorhabditis elegans," Nucleic Acid Research, vol. 32(11), pp.3392-3399, 2004.

[9] Y. Cheng, R.M. Miura, B. Tian, "Prediction of mRNA polyadenylation sites by support vector machine," Bioinformatics, vol. 22(19), pp.2320-2325, 2006.

[10] J. Hu, C.S. Lutz, J. Wilusz, B. Tian, "Bioinformatic identification of candidate cis-regulatory elements involved in human mRNA polyadenylation," Bioinformatics, vol. 11, pp.1485-1493, 2005.

[11] M.N. Akhtar, S.A. Bukhari, Z. Fazal, R. Qamar and L.A. Shahmuradov, "POLYAR, a new computer program for prediction of poly(A) sites in human sequences," BMC Genomics, vol. 11(646), 2010.

[12] M. Kalkatawi, F. Rangkuti, M. Schramm, B.R. Jankovic, A. Kamau, R. Chowdhary, J.A.C Archer and V.B. Bajic, "Dragon PolyA Spotter: predictor of poly(A) motifs within human genomic DNA sequences," Bioinformatics, vol. 28(1), pp.127-129, 2012.

[13] K. Deep, M. Thakur, "A new crossover operator for realcoded genetic algorithms," Applied Mathematics and Computation, vol. 188(1), pp.895-911, 2007.

[14] A. Sedki, D. Ouazar and E.E. Mazoudi, "Evolving neural network using real coded genetic algorithm for daily rainfall-runoff

forecasting," Expert Systems with Applications, vol. 36, pp.4523-4527, 2009.
[15] C.Y. Huang, L.H. Chen, Y.H. Chen and F.M. Chang, "Evaluating the process of a genetic algorithm to improve the back-propagation network: A Monte Carlo study," Expert Systems with Applications, vol. 36, pp.1459-1465, 2009.
[16] S.R. Sexton, R.E. Dorsey and J.D. Johnson, "A comparison of the genetic algorithm and backpropagation," Decision Support Systems, vol. 22, pp.171-185, 1998.
[17] R.R. Laxton, "The measure of diversity," J.Theor.Biol, vol. 70, pp.51-67, 1978.
[18] Y. Cui, J.Q. Han, D.X. Zhong and R.L. Liu. "A novel computational method for the identification of plant alternative splice sites," Biochemical and biophysical research communications, vol. 431(2), pp.221-224, 2013.
[19] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," Math. Control Signals Systems, vol. 2, pp.303-314, 1989.