Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2013, 5(11):678-683

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Protein tertiary structural prediction based on a novel flexible neural tree

Hao Teng, Shuhui Liu and Yuehui Chen

School of Information Science and Engineering, University of Jinan, Shandong Provincial Key Laboratory of Network based Intelligent Computing, Jinan, China

---

ABSTRACT

The study of protein tertiary structure prediction is useful to protein function. This paper proposes a novel protein tertiary structural prediction approach based on flexible neural tree (FNT). In this approach, the approximate entropy and hydrophobicity pattern of a protein sequence are adopted to characterize the Pseudo-Amino Acid (PseAA) components as input. A novel quantum particle swarm optimization (QPSO) combined with the speed and disturbance is presented and used to optimize the parameters of FNT. The 640 protein sequence is used as the dataset. The experiment data is validated by ten- fold cross validation and the result shows the approach based on the novel quantum particle swarm optimization and FNT can improve the predictive accuracy.

Keywords: protein structure classification, Pseudo-Amino Acid composition, hydrophobicity, flexible neural tree, quantum particle swarm optimization

---

INTRODUCTION

The basic hypothesis of Protein structure prediction is that the protein tertiary structure is decided by its amino acid sequence only. The study of protein tertiary structure is useful to protein function. Tertiary structure of protein is complex and irregular, but the type of protein folding in the natural state is no more than 1000 [1]. According to the type of the folding, Levitt and Chothia [2] defines protein structure of the following four classes: all- α , all- β , $\alpha + \beta$, α / β .

In recent years, many researches have been done in this field, including: finding some new feature extract methods and new classifiers for the improvement of the protein tertiary structure. Most of the previous methods on the effective feature representation for protein are based on amino acids (AA). In order to contain more information of the sequence order, literature [3] presents the concept of pseudo amino acid composition (PseAA).

The basic principle of pseudo amino acid composition is that a protein will be expressed in $20 + \lambda$ quantity, 20 expresses the amino acid composition, λ is the number of additional features. These additional features combined with the order of sequence information that can improve the defect of amino acids. Here pseudo amino acid composition is adopted and approximate entropy and hydrophobic is selected as the additional feature.

We apply FNT as the base classifier and ensemble these based classifiers as a M-ary model. FNT's structure and parameter are optimized respectively by probabilistic incremental program evolution (PIPE) [4] and a kind of novel quantum particle swarm optimization.

For each base classifier, 27-D features are used as the input for training. Two 0-1 classifier integrates a M-ary model. According to the outputs of the two child classifier the finally results can be gotten.

The 640 protein sequence is used as the dataset. The experiment data is validated by tenfold cross validation. The experiment result shows our method can improve the predictive accuracy rate.

The rest of this paper is organized as follows. Dataset and Feature extract is introduced in Section 2. The classification model and a brief introduce about the theory of Flexible Neural Tree classifier is given in Section 3. The Flexible Neural Tree based on a novel quantum particle swarm optimization is presented in Section 4. Section 5 explains the experiments of that were carried out and others. Finally, Section 6 summarizes the conclusions of our work.

DATASET AND FEATURE EXTRACT

A. *Dataset*

Here 640 dataset is selected to make the experiment, which includes 640 protein samples：138 samples belong to all-α class, 154 samples belong to all-β class, 177 samples belong to α + β class, and 171 samples belong to α / β class. The sequence homology of this dataset is about 25%. It makes the experiment more persuasive because of the lower sequence homology.

B. *Feature extract*

According to the concept of PseAA composition, the protein sequence can be formulated as:

$$p = (p_1, p_2, ..., p_{20}, p_{21}, ..., p_{20+\lambda})^T \ (\lambda \langle \mathrm{L})$$

The first 20 components are the occurrence frequencies of 20 amino acids in sequence(as table 1). λ is the number of additional features. P_i(21 < i < 20 + λ) are the additional features. Here approximate entropy and hydrophobic are selected as the additional feature. Approximate entropy is a non-negative parameter, which represents the complexity of time sequence [5]. Be regarded as a time sequence, the sequence of amino acids can be replaced with corresponding digital, thus the complexity of different structure of protein will be different [6].

The literature [7] shows that the couple of hydrophobic amino acids have six types:

(i,i+2), (i,i+3),(i,i+2,i+4),(i,i+5),(i,i+3,i+4),(i,i+1,i+4).Among them, (i,i+2) and (i,i+2,i+4) often appeared in the beta-folding, (i,i+3),(i,i+3,i+4),(i,i+1,i+4)and(i,i+5) often appeared in the spiral. There are 7 kinds of amino acid which was identified as hydrophobic amino acids, respectively is: Cys(C),Val(V),Leu(L),Ile(I),Met(M), Phe(F) and Trp(W) (as table 2). The formula for the frequency of two hydrophobic amino acids such as (i,i+2) can be found as below:

$$f(i,i+2) = \frac{1}{N}\sum_{i=1}^{N-2} C(i,i+2)$$

Other hydrophobic amino acids can be calculated in the same way. Hydrophobic of Alpha helix model is the sum of f (i,i+3), f (i,i+3,i+4), f (i,i+1,i+4), hydrophobic of Beta-folding is the sum of f (i,i+2) and f (i,i+2,i+4), random coil is f (i,i+5). The three hydrophobic are the other additional features of the sequence.

Table 1. The reverse encoding for Amino acids

A=GCT	G=GGT	M=ATG	S=TCA	C=TGC	H=CAC	N=AAC	T=ACT	D=GAC	I=ATT
P=CCA	V=GTG	E=GAG	K=AAG	Q=CAG	W=TGG	F=TTC	L=CTA	R=CGA	Y=TAC

Table 2. Amino acids hydration property classification

First class	Second class	Third class
R, K, E, D, Q, N	G,A, S, T, P,H,Y	C, V, L, I, M, F,W

CLASSIFICATION MODEL

A. *M-ary model*

In literature [8], a method which is called the M-ary SVM is introduced, represents each category in binary format, and to each bit of that representation is assigned a conventional SVM. This model requires only $\log_2(k)$ SVMs, where k is the number of classes. In literature [6], this M-ary model is used to make a M-ary model of FNT.

Protein tertiary structure studied in this paper includes four categories, {all-α , all-β , α + β , α / β }, respectively marked as {class 1, class 2, class 3 and class 4}, so $\log_2(4) = 2$ is the number of classifier. For the first child classifier, all of the corresponding sample data of class 2 and class 4 marked as 1, all the corresponding sample data of class 1 and class 3 marked as 0. For the second classifier, data of class 2 and class 3 marked as 1, data of class 1 and class 4 marked as 0. Then, two FNT classifiers can be trained respectively (as figure 1).

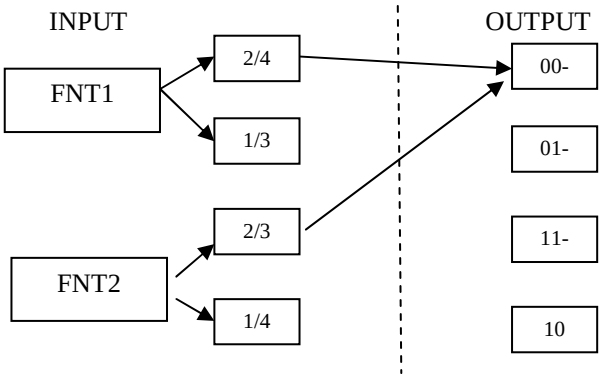

Figure 1. A M-ary model of FNT

When it is applied in testing or classification, the final output can be gotten according to results of the two child classifier, as shown in table 3.

For example, for a test sample X, the output of the first and second child classifier is 0 and 1 respectively. By the output of the first classifier, X belongs to class 1 or class 3; By the output of the second classifier, X belongs to class 2 or class 3. So, as a result, the input of X belongs to class 3.

Table 3. Output of M-ary

FNT1	FNT2	OUTPUT	BELONG
0	0	class 1	all- α
1	1	class 2	all- β
0	1	class 3	α + β
1	0	class 4	α / β

B. *FNT Flexible neuron instructor and FNT model*

According to the characteristics of complex system, Flexible Neural Tree (FNT) [9-11] is presented by Chen, as a general sense of Neural network system, uses the tree structure and a collection of a set of operators (as figure 2). The Flexible Neural Tree has been applied to time-series prediction, cancel detection, i.e., successfully.

A FNT and its function instruction set and terminal instruction set can be represented as follow:

Function instruction set is: $F=\{+_2,+_3,\ldots,+_N\}$, terminal instruction set is: $T=\{x_1,x_2,\ldots,x_n\}$,where $+_i(i= 2, 3, \ldots ,n)$ denote the instructions of non-leaf nodes with i arguments. $x_1,x_2,\ldots,x_n$ are instructions of leaf nodes with no other arguments. The non-leaf node's output is calculated as a flexible neuron model. For a selected non-terminal instruction, i real values are randomly generated and denote the connection strength between the node $+_i$ and its connecting leaf nodes. In this paper, the activation function can be calculated as (1) in which two adjustable parameters a_i and b_i are created randomly:

$$f(a_i, b_i, x) = \exp(-(\frac{x - a_i}{b_i})^2) \tag{1}$$

The total excitation of $+n$ is as (2):

$$net_n = \sum_{j=1}^{N} w_j * x_j \tag{2}$$

where x_j (i=1,2, . . . ,n) are the inputs to node $+n$. The output of the node $+n$ can be calculated as (3):

$$out_n = f(a_i, b_i, net_n) = \exp(-(\frac{net_n - a_i}{b_i})^2) \tag{3}$$

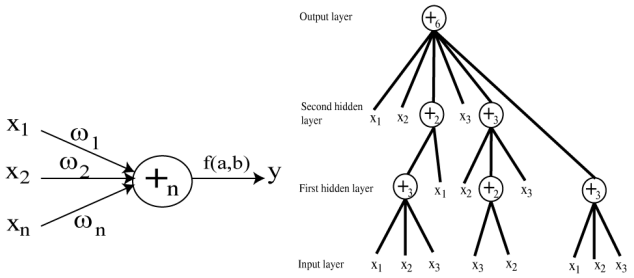

Figure 2. A FNT and its function instruction set and terminal instruction set.

FNT based on an improved quantum particle swarm optimization

A. *PSO*

The parameters of FNT usually optimized by particle swarm optimization (PSO) [12] .In this method, each particle adjusts its flying according to the flying experience of its own and its companions'. Each individual is treated as particle and also is treated as a point in a dimensional space.

The i_{th} particle is noted as: $x_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T, i = 1, 2, \cdots, m$,the best previous position of particle is denoted as $p_{ibest} = (p_{i1}, p_{i2}, ..., p_{id})^T$. g_i is the position of the best particle among all the particles in the population, called the global best (gbest) position. The velocity for particle i_{th} is denoted as $v_i = (v_{i1}, v_{i2}, \cdots, v_{id})^T$. The particles can be described according to (4) and (5):

$$v_{ik} = \omega v_{ik} + c_1 rand_1(p_{ik} - x_{ik}) + c_2 rand_2(g_k - x_{ik}) \qquad \text{where} \qquad i = 1, \cdots, m; k = 1, \cdots d \ , \tag{4}$$

$$x_{ik} = x_{ik} + v_{ik} \qquad i = 1, \cdots, m; k = 1, \cdots d \tag{5}$$

where c_1 , c_2 >0, rand_1 and rand_2 are two random numbers in the range [0,1].

B. *QPSO*

In [13], Sun et al introduce quantum theory into PSO and present a Quantum-behaved PSO algorithm (QPSO). The experiment shows that QPSO is better than PSO on several test functions. In practice applications of training RBF neural networks [14] and others, the algorithm also shows better performance [15].

For QPSO all particles records their pbest and compares its pbest with those of others in population to get the gbest in each iteration. Then the next point can be given by (6), in which φ_1 and φ_2 are two random numbers in the range [0,1].

$$p = (\varphi_1 pbest + \varphi_2 gbest) / （\varphi_1 + \varphi_2） \tag{6}$$

A Mean Best Position (mbest) is defined as the center of gravity of gbest position which is discribed as (7).

$$mbest = \sum_{i=1}^{M} \varphi_1 pbest_i / M \tag{7}$$

Where M is the number of population size. Therefore, QPSO can iterative according to (8):

$$x(t+1) = p \pm \beta * \left| mbest\text{-}x(t) \right| * \ln(1/u) \tag{8}$$

Where β is called Creativity Coefficient, u is a random number between 0 and 1. ± is decided by a random number between 0 and 1 in every iteration, when the number is bigger than 0.5, - is used, otherwise + is used.

As a result, QPSO has simple form and only one parameter β works on individual particle's convergence speed; therefore it is very suitable to treat most optimization problem.

C. *Improvements of QPSO*
In order to improve the performance of QPSO, scholars do many works about this algorithm.

In literature [16], although the particle's position and speed can't determined at the same time according to the "uncertainty principle" , however, the speed of the particle still have a certain influence to the speed of reaching the optimum solution. Then using velocity of the particles to guide the particles fly towards to the optimal solution, also can improve the performance of algorithm. The experimental results also prove it. Compared with the original algorithm, the improved algorithm has two major improvements:

Using the speed of the individual particles to create a number between (0, 1) ,and use it to replace the randomly number in original algorithm which is used to change the update of particle position, as equation (9) .

$$rand - q = 1/(1+\left|(V_{\max} - V_{id})(V_{id} - V_{\min})\right|) \quad (9)$$

If the position vector of the individual particles is beyond the scope of the preset, then make the particles flying to the opposite direction.

All the particles in standard PSO and QPSO algorithm will converge to a common point [17]. So the diversity of population is low and the particle rarely search further before the next iteration. In order to overcome this problem, an Improved method is adding gaussian disturbance in QPSO algorithm so as to enhance the diversity and performance of the algorithm [18].

The following three strategies can be used :
A) join the gaussian disturbance in the average of the best position.
B) join the gaussian disturbance in the global best position.
C) join the gaussian disturbance in both A and B.

Here, we join the gaussian disturbance in the average of the best position, as equation (10).

$$Mbest=mbest+e*R \quad (10)$$

Where e is a preset parameter, R is a random numbers which satisfy the gaussian disturbance. Here set e=0.005.

D. The produce process of FNT model based on the novel QPSO
FNT model produced by the process as below, in this M-ary model, two FNT must be made:
1) Create an initial population randomly (a FNT and a set of its parameters);
2) Structure optimization to find a better tree structure by the tree variation operators, to find an appropriate hierarchical architecture for a given problem, here PIPE is adopted;
3) If a better structure is found, then go to step 4), otherwise go to step 2);
4) Parameter optimization can be achieved by the parameter learning algorithm, such as genetic algorithms, particle swarm optimization, and so on, here an improved quantum particle swarm optimization which has be introduced as above is adopted in the parameter optimization of FNT;
5) If no better parameter vector appears for a enough long time, or the maximum number of local search is reached then go to step 6); otherwise go to step 4);
6) If satisfactory solution is found, the algorithm stopped; otherwise go to step 2);
7) The final output can be gotten from the results of the two child classifier.

To find an optimal FNT, the normalized mean squared error (RMSE) is adopted.

RESULTS AND DISCUSSION

Experimental results and analysis
The cross validation method is usually used to evaluate the performance of classification method in the classification problems. Here ten-fold cross validation was adopted. We calculate the accuracy of four classes and the overall success rate. The predict result of different algorithms are showed in the table 4. From table 4 we can see that the accuracy of our method is obviously higher than most of some other experiments. It performed marginally better than the model of ECOC, but this method use less number of FNT then ECOC, so it is better as a whole.

Table 4. The results of the new method and others

algorithms	accuracy rate				overall accuracy rate
	all- α	all- β	α + β	α / β	
IB1[19]	53.62	46.10	68.93	34.50	50.94
Naïve Bayes[19]	55.07	62.34	80.26	19.88	54.38
Logistic regression[19]	69.57	58.44	61.58	29.82	54.06
method in [6]	76.81	61.68	62.14	41.17	59.53
ECOC [20]	55.88	52.63	77.27	57.14	60.73
This method	61.59	73.38	69.49	40.59	60.94

CONCLUSION

This paper uses M-ary classifier as the classification model based on FNT classifier, an improved quantum particle swarm optimization (QPSO) is used to optimize the parameters of FNT. The experiment result shows the new method can improve the predictive accuracy rate and the new method is valuable for protein structure prediction.

Acknowledgment
The research work was supported by National Natural Science Foundation of China under Grant No. 61070130, the Key Project of Natural Science Foundation of Shandong Province (ZR2011FZ001).

REFERENCES

[1] C. Chothia; *Nature*, **1992**, 357(6379),543-544.
[2] M. Levitt; C. Chothia; *Nature*, **1976**,261(2),552
[3] X. Xiao; S. H. Shao; Z. D. Huang et al; *Journal of computational chemistry*, **2006**, 27(4),478.
[4] R. Salustowicz; J.Schmidhuber; *Evolutionary Computation*, **1997**, vol.14,123.
[5] J. S. Richman; J. R. Moorman; *American Journal of Physiology-Heart and Circulatory Physiology*, **2000**, 278(6),H2039.
[6] X. Huang; Y. Chen; Y. Cao; *Computer Engineering*, **2011**, 37(1),159.
[7] V. I. Lim; *Journal of Molecular Biology*, **1974**, 88(4), 873.
[8] D. J. Sebald; J. A. Bucklew; *IEEE Transactions on*, **2001**, 49(11), 2865.
[9] Y. Chen; B. Yang and J. Dong; Nonlinear System Modeling via Optimal Design of Neural Trees, International Journal of Neural Systems, **2004**, Vol.14, No.2, 125.
[10] Y. Chen; B. Yang; J. Dong and A.; Time-series Forecasting Using Flexible Neural Tree Model, Information Science, **2005**, Vol.174, Issues 3/4, 219.
[11] Y. Chen; A. Abraham; Y. Zhang; Ensemble of Flexible Neural Trees for Breast Cancer Detection, The International Journal of Information Technology and Intelligent Computing, **2006**, Vol. 1, No.1, 187-201.
[12] J. Kennedy; R.Eberhart [C]//C,Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. **1995**, 1942.
[13] J. Sun; B. Feng; W. Xu; Particle swarm optimization with particles having quantum behavior[C]//Evolutionary Computation, 2004, CEC2004. Congress on. IEEE, **2004**, 1, 325.
[14] J. Sun; W. Xu; J. Liu; Training RBF neural network via quantum-behaved particle swarm optimization[C]//Neural Information Processing, Springer Berlin Heidelberg, **2006**, 1156.
[15] M. Xi; J. Sun; W. Xu; *DCDIS Ser. B: Complex Systems and Applications, Modeling, Control and Simulations*, **2007**, 14(S2), 603
[16] J. MA; P.TANG P; *Computer Engineering and Applications*, **2007**, 43(36) : 89
[17] Y. Shi; R. Eberhart; A modified particle swarm optimizer[C] //Evolutionary Computation Proceedings, **1998**. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on. IEEE, **1998**, 69-73.
[18] X.Wang; H.LONG; J.SUN; *Application Research of Computers*, **2010**, 27(6), 2093.
[19] K. Chen; Kurgan L A; J. Ruan; *Journal of computational chemistry*, **2008**, 29(10), 1596.
[20] Y. Chen; Y. Chen; Predict the Tertiary Structure of Protein with Error-Correcting Output Coding and Flexible Neural Tree[C]//C, Proceedings of the 2nd International Conference on Computer and Information Applications, **2012**,230.