Journal of Computer-Aided Molecular Design **18:** 797–810, 2004.
DOI 10.1007/s10822-005-0578-7

Prediction of inter-residue contacts map based on genetic algorithm optimized radial basis function neural network and binary input encoding scheme

Guang-Zheng Zhang[1,2,3,*] & De-Shuang Huang[1]
[1]*Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences;*
[2]*Department of Automation, University of Science & Technology of China;* [3]*Department of Unmanned Aerial Vehicle, Academy of Artillery, People's Liberation Army, P.O. Box 1130, Hefei, Anhui 230031, P.R. China*

Received 29 April 2004; accepted in revised form 14 December 2004

Key words: contacts map, contact threshold, protein folding, radial basis function neural network

Summary

Inter-residue contacts map prediction is one of the most important intermediate steps to the protein folding problem. In this paper, we focus on the problem of protein inter-residue contacts map prediction based on neural network technique. Firstly, we use a genetic algorithm (GA) to optimize the radial basis function widths and hidden centers of a radial basis function neural network (RBFNN), then a novel binary encoding scheme is employed to train the network for the purpose of learning and predicting the inter-residue contacts patterns of protein sequences got from the protein data bank (PDB). The experimental evidence indicates the utility of our proposed encoding strategy and GA optimized RBFNN. Moreover, the simulation results demonstrate that the network got a better performance for these proteins, whose residue length falls into the area of (100, 300), and the predicted accuracy with a contact threshold of 7 Å scores higher than the other 3 values with 5, 6, and 8 Å.

Introduction

Protein three-dimensional (3D) architecture prediction from primary amino acids sequence is a fundamental and unresolved problem in molecular biology. Full molecular modelling to find the structures is at present intractable, and so intermediate steps, such as inter-residues contact prediction [1–5], and residue spatial distance prediction [6–8], have been developed rapidly recently. Our previous studies [9, 10], show that two amino acid residues which are far apart in sequence might be close together in spatial relation, and this will increase the contact probability of the residues. Therefore, the prediction of inter-residue contacts in proteins (a representation that describes the spatial neighborhood relation between secondary structure elements such as helices, beta sheets, and random coils) is an interesting problem whose solution may be useful in protein-folding recognition, secondary or tertiary structure prediction, *de novo* design and so on [3].

In this paper, the 173 non-homologous protein sequences, described in [3], are adopted in our study, and we grouped them into two parts as training and testing sets respectively. Then based on our proposed binary input encoding scheme and a RBFNN, whose hidden centers and base function widths are optimized by genetic algorithm (GA), we trained the network for the purpose of learning and predicting inter-residue contact map patterns.

The rest of the paper is organized as follows: first the detailed inter-residue contact definition is

*To whom correspondence should be addressed. E-mail: gzzhang@iim.ac.cn

given; then a genetic algorithm is used to optimize the RBFNN; and the binary input encoding strategy, data preparation and training algorithm are illustrated; the simulation results and discussion are given subsequently and finally conclusions are listed.

Inter-residue contact definition

As an important way of viewing the 3D structure of protein sequence, inter-residue contact map, in essence, is a matrix of all the pairwise distances within the molecule. The axes of the contact map are linear in the sequence of the protein chain, which makes it attractive for relating structural features to the sequence [11].

Generally, inter-residue contacts map is a particularly useful 2D representation of a protein's 3D structure. Within a protein sequence, $S = \{S_1, S_2, \ldots, S_N\}$, two amino acid residues, S_i and S_j, are considered in contact if the 3D spatial distance is less than a contact threshold t. The distance involved in the different contact definitions can be the one between the $C_\alpha - C_\alpha$ atoms of the two residues [12], between the $C_\beta - C_\beta$ [3, 13] and the minimal distance between atoms belonging to the side chain or to the backbone of the two residues [14]. But here, we use the distance between the two geometrical centers of residues, determined by the coordinates of their atoms, to gain the contact map. Specifically, we use the spatial distance described in [10], $\sigma(S_i, S_j) = |r_i - r_j|$, where r_i and r_j are the geometrical coordinates of the two residues, to determine whether the residues are in contact or not.

The contact map of a protein with N amino acid residues ia an $N \times N$ matrix, C, whose elements are defined as

$$C_{i,j} = \begin{cases} 1 & \text{if residues } S_i \text{ and } S_j \text{ are in contact} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The contact between two residues can be defined in different ways. In particular, we will use the spatial distance described in [10] to determined whether the residues are in contact or not, so the definition 1 becomes

$$C_{i,j} = \begin{cases} 1 \text{ if } |r_i - r_j| < R_c \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

where r_i and r_j are the geometrical coordinates of the two residues.

Genetic algorithm optimized radial basis function neural network

Neural network based contacts map prediction

The problem of contact map prediction can be stated as a classification problem. Given a set of proteins with known structures, contact residues and non-contact residues are separated as positive and negative instances. For each instance, various features are collected to capture useful information of the pair of residues, including amino acid content, physicochemical environment, secondary structures, evolutionary correlation, and other information that can discriminate contacts from non-contacts. Then these feature vectors of both positive and negative instances are used as the training set to neural network or other classifiers (i.e., predictors) to learn the contact information between the given two amino acid residues of a protein.

The neural network technique, as an important machine learning and pattern recognition approach, has been widely used in protein engineering, such as protein structure prediction and contacts map prediction. Previous work on protein residue contact prediction has employed neural network [13, 14], and statistical techniques based on correlated mutations [3, 4, 6]. But the studies [13, 14] have shown that neural network based prediction approach scores higher than statistical approaches. At the same time, Jingjing Hu et al. (2002) [5], have also shown that it is possible to recover the 3D structure from even corrupted contact map. So, we here use a radial basis function neural network (RBFNN) to predict the inter-residue contacts map within protein sequences.

Radial basis function neural network (RBFNN)

Because of the simple topological structure and the rapid training procedure, the RBFNN provides a powerful technique for generating multivariate

nonlinear mapping [15]. RBFNN has been successfully adopted to solve these multi-parameters and -model type of nonlinear classification problem, such as protein inter-residue contacts maps prediction.

Generally, a RBFNN consists of three layers (see the RBFNN architecture shown in Figure 1): (1) the input layer, where all the feature vectors of the training and testing samples are input; (2) the hidden layer, where at each node the input feature vector is put through a radial basis function (RBF) that is centered on a corresponding exemplar vector $\overrightarrow{\mathbf{V}}$; (3) the output layer, where all the inputs from the hidden layer are combined to indicate the class of input sample. The weights between the hidden and the output layer are adjusted during the training procedure for the purpose of minimizing the cost function. The whole architecture is therefore fixed by determining the hidden layer and the weights between the middle and output layers [17]. For an input vector $\mathbf{X} = [x_1, x_2, \ldots, x_n]^T \in R^n$, and with N_h middle layer neurons, the activation function $\varphi(\cdot)$ is described by a center $C_l \in R^n$ and a width σ_l, where $l = 1, 2, \ldots, N_h$. The general equation of an output neuron j is given by $S_j(\mathbf{X}) = \sum_{l=1}^{N-h} w_{ij}\varphi_l(\mathbf{X}) + b_j$, where $w_{ij} \in R$ is the weight between the hidden neuron l and the output neuron j, b_j is a possible bias.

Let us denote the training exemplar feature vectors set as $\mathbf{X} = \{x^q : q = 1, \ldots, Q\}$, and x^q be an exemplar vector with N components, $x^q = (x_1^q, x_2^q, \ldots, x_N^q)$. Each actual output from the jth node, z_j^q, is forced to match the target component, t_j^q, by adjusting the weights. At the same time, a bias, b_j, is added onto the sum and also adjusted to help to approximate the target value. Then the output from each of the J nodes for the input vector, x^q, has the following form:

$$z_j^q = \frac{1}{M}\sum_{m=1}^{M} w_{mj} y_m^q + b_j = \frac{1}{M}\sum_{m=1}^{M} w_{mj} exp\left(-\frac{(x^q - x^m)^2}{2\sigma^2}\right) + b_j \qquad (3)$$

where $w_{i,j}$ is the weight between the hidden and output layer, and $exp(\cdot)$ the Gaussian radial basis function.

Genetic algorithms (GAs)

Genetic algorithms are a reliable and powerful search strategy to find sub-optimal solutions in huge data spaces. The principles of evolution, namely reproduction, mutation and mating, are thereby applied to a great class of problems, which are frequently known as NP-complete ones. Like the descriptions in [18–27], GAs usually cannot find the very optimal solution, but the one that satisfies certain qualitative requirements. In all cases the principle of evolution is applied to a set of data, which serves as genotype, for example a stream of bits or arithmetic expressions. The

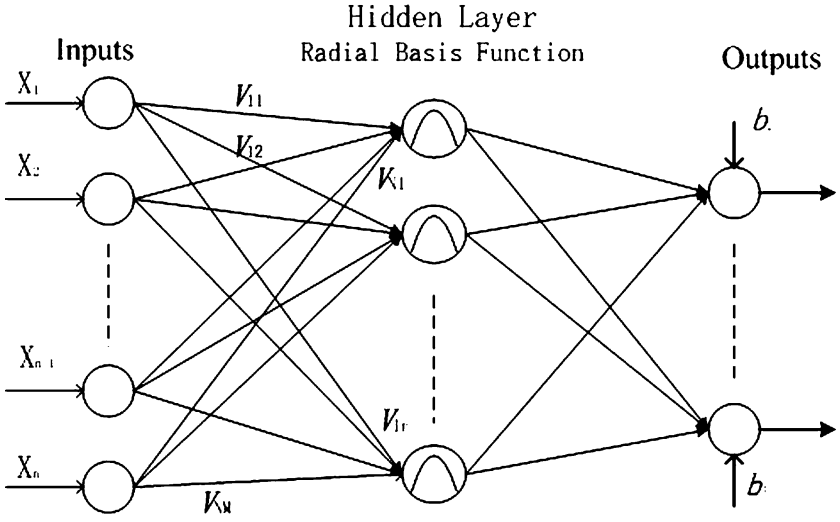

Figure 1. The architecture of RBFNN.

evolving individuals are tested for their fitness to solve a given problem. This is done by the so called fitness function, which has to be defined carefully to obtain the desired results. The complexity of the fitness function depends on the nature of the problem to be solved.

Genetic algorithms (GAs), as a form of inductive learning strategy, are adaptive search techniques which have demonstrated substantial improvement over a variety of random and local search methods [18]. This is accomplished by their ability to exploit accumulating information about an initially unknown search space in order to bias subsequent search into promising subspaces. Since GAs are basically a domain independent search technique, they are ideal for applications where domain knowledge and theory is difficult or impossible to provide.

Generally, the basic concept of GAs is that a population of the given problem solutions is maintained in the form of chromosomes, which are strings (binary encoding scheme is the most simple and typical one) encoding problem solutions [19]. Those strings are converted into problem solutions and evaluated according to fitness function. Based on the fitness evaluation, a new population of chromosomes if generated by applying genetic operators (selection, crossover and mutation). The selection process copies parent chromosomes into a new population. The crossover operator recombines genetic material of two parent chromosomes to produce offspring on the basis of crossover probability. Provided that two chromosomes were $S_i^1 = (1,1,0,0)$ and $S_i^2 = (0,0,1,1)$, one-point crossover at the third point produces two new chromosomes, $S_{i+1}^1 = (1,1,1,1)$ and $S_{i+1}^2 = (0,0,0,0)$. Finally, the mutation operator selects a random position of a random string and negates the bit value. For instance, if mutation is applied to the third bit of S_{i+1}^1, then the mutated offspring becomes $S_{i+1}^{1'} = (0,0,1,0)$. The procedure continues until an acceptable solution is got.

Optimization strategy

Since the RBFNN was proposed, quite a few learning algorithms and optimization schemes are adopted to improve the network efficiency and general performance. How to determinate the number of the units in the hidden layer and the wights matrix between the hidden and output layers are the important aspects in RBFNN optimization procedure, and learning algorithm designed for different applications may employ different optimization criteria. But, basically, there are two categories of learning algorithms proposed for RBFNN. The first category of learning algorithm simply places one radial basis function at each sample. On the other hand, the second category of learning algorithms attempts to select the most appropriate number of hidden units in the hidden layer, or the widths of radial basis function (RBF).

Here, we use a similar genetic encoding method described in [10, 20, 21], to generate the centers and widths of the radial basis function (RBF). Specifically, as shown in Figure 2, each chromosome in-

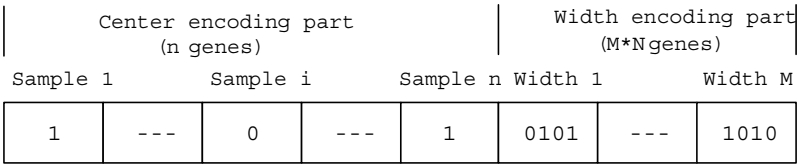

Figure 2. Encoding scheme of chromosome.

Table 1. The 4 test protein sequences.

PDB code	Residue length	*R* value	Resolution (Å)	Classification
1YSC	421	0.162	2.80	Hydrolase (carboxypeptidase)
1CNE	270	0.190	3.00	Nitrogenous acceptor
5P21	166	0.192	1.35	Oncogene protein
1SPY	89	–	–	Muscle protein

Table 2. The protein database used as training set of the neural network predictor.

PDB code	Residue length	*R* value	Resolution (Å)	Classification	PDB code	Residue length	*R* value	Resolution (Å)	Classification
				Residue length $L_s \leq 100$					
1A1i_A	90	0.191	1.60	Complex (zinc finger/DNA)	1A1T_A	55	–	–	Complex (nucleocapsid protein/RNA)
1A68	95	0.200	1.80	Potassium channels	1A7i	81	–	–	Lim domain containing proteins
1ACP	79	–	–	Fatty acid synthesis	1AH9	71	–	–	Ribosome binding
1AHO	64	0.158	0.96	Neurotoxin	1AIE	31	0.191	1.50	P53 tetramerization
1AIL	73	0.182	1.90	RNA-binding protein	1AJJ	37	0.209	1.70	Receptor
1AOO	40	–	–	Metallothionein	1AP0	73	–	–	Chromatin-binding
1ARK	60	–	–	Transferase	1AWD	94	0.146	1.40	Electron transport
1AWJ	77	–		Transferase	1AWO	60	–	–	Kinase
1BBO	57	–	–	DNA-binding protein	1BC8_C	93	0.220	1.93	Complex (DNA-binding protein/DNA)
1C5A	73	–	–	Complement factor	1CFH	47	–		Coagulation factor
1CTJ	89	0.137	1.10	Electron transport	1CYO	93	0.160	1.50	Electron transport
1FNA	91	0.180	1.80	Cell adhesion protein	1HEV	43	–		Lectin
1HRZ_A	92	–	–	DNA-binding protein/DNA	1KBS	60	–	–	Cytotoxin
1MBH	53	–	–	DNA-binding protein	1MBJ	53	-	–	DNA binding protein
1MSI	70	0.198	1.25	Antifreeze protein	1MZM	93	1.78	0.171	Lipid transport
1NXB	62	0.240	1.38	Neurotoxin (Post-synaptic)	1OCP	67	–	–	DNA-binding protein
1OPD	62	0.182	1.50	Phosphotransferase	1PCE	60	–	–	Proteinase inhibitor (Kazal type)
1PLC	99	0.150	1.33	Electron transport	1POU	71	–	–	DNA-binding protein
1PPT	36	–	1.37	Pancreatic hormone	1BRF	53	0.95	0.132	DNA-binding protein
1SCO	38	–	–	Potassium channel inhibitor	5PTI	58	0.200	1.00	Proteinase inhibitor (Trypsin)
1SRO	76	–	–	S1 RNA-binding domain	1TBN	82	–	–	Calcium-binding protein
1TIV	86	–	–	Transcription activation	1TLE	58	–	–	Glycoprotein
1TSG	98	–	–	Cell adhesion	1UBI	76	1.80	0.165	Chromosomal protein
1UXD	65	–	–	Transcription regulation	2HOA	68	–	–	DNA-binding protein
2ACY	98	0.170	1.80	Acylphosphatase	2ADX	40	–	–	Blood coagulation
2FOW	76	–	–	Ribosome	2ECH	50	–	–	Blood coagulation inhibitor
2FDN	55	0.100	0.94	Electron transport	2FN2	59	–	–	Glycoprotein
2HQI	72	–	–	Transport	1ROF	60	–	–	Electron transport
2SN3	65	0.192	1.20	Toxin	2SXL	88	–	–	RNA-binding domain
3GAT_A	98	–	–	Transcription regulation/DNA	3MEF_A	69	–	–	Gene regulation
4MT2	62	0.197	2.00	Metallothionein			–	–	

Table 2. (Continued)

PDB code	Residue length	*R* value	Resolution (Å)	Classification	PDB code	Residue length	*R* value	Resolution (Å)	Classification
				Residue length 100 < $L_s \leq 200$					
2BOP_A	102	0.200	1.70	Transcription regulation	2HFH	109	–	–	Hnf-3 homologues
1A62	130	0.216	1.55	Transcription termination	1A6G	151	0.126	1.15	Heme protein
1ACZ	108	–	–	Polysaccharide degradation	1ASX	159	0.222	2.80	Chaperonin
1AUD_A	131	–	–	Complex (ribonucleoprotein/RNA)	1AX3	162	–	–	Phosphotransferase system
1B10	142	–	–	Prion protein	1BC4	111	–	–	Hydrolase
1BD8	156	0.190	1.80	Tumor suppressor	1BEA	127	0.195	1.95	Serine protease inhibitor
1BFE_A	119	0.214	2.30	Peptide recognition	1BFG	146	0.153	1.60	Growth factor
2LEF_A	119	–	–	Gene regulation/DNA	1BKF	107	0.187	1.60	Isomerase
1BKR_A	109	0.141	1.10	Actin-binding	1BR0	120	–	–	Membrane protein
1BSN	138	–	–	Hydrolase	1BV1	159	0.198	2.00	Allergen
1BXA	105	0.169	1.30	Electron transport	1C25	161	0.227	2.30	Hydrolase
1CEW_I	105	0.198	2.00	Proteinase inhibitor (cysteine)	1CFE	135	–	–	Pathogenesis-related protein
7RSA	124	0.150	1.26	Hydrolase (phosphoric diester)	1DUN	134	0.159	1.90	Hydrolase
1ECA	136	–	1.40	Oxygen transport	1ERV	105	0.210	1.65	Oxidoreductase
1EXG	110	–	–	Cellulose degradation	1HFC	169	0.174	1.56	Metalloprotease
1IFC	132	0.169	1.19	Lipid-binding protein	1JVR	137	–	-	Matrix protein
1KPF	126	0.209	1.50	Protein kinase inhibitor	1KTE	105	0.189	2.20	Electron transport
1BGF	124	0.188	1.45	Transcription factor	1NPK	154	0.196	1.80	Phosphotransferase (Po_4 as acceptor)
1PDN_C	158	0.234	2.50	Gene regulating protein/DNA	1PKP	150	0.220	2.80	Ribosomal protein
1POA	118	0.143	1.50	Hydrolase	1PUT	106	–	–	Electron transport
1RA9	159	0.169	1.55	Oxidoreductase	1RCF	169	0.139	1.40	Electron transfer (flavoprotein)
1RIE	129	0.187	1.50	Electron transport	1SKZ	119	0.217	1.90	Serine protease inhibitor
1TAM	132	–	–	Matrix protein	1VSD	152	0.150	1.70	Endoribonuclease
1WHI	122	0.189	1.50	Ribosomal protein	2FSP	124	–	–	Response regulator
2GDM	153	0.158	1.70	Oxygen transport	2ILK	160	0.163	1.60	Cytokine
2LFB	100	–	–	DNA-binding	2PIL	159	0.187	2.60	Fiber-forming protein
2TGI	112	0.173	1.80	Growth factor	2UCZ	165	0.227	2.93	Ubiquitin conjugation
1MAK	113	–	–	Immunoglobulin	3LZT	129	0.092	0.92	Hydrolase
3NUL	130	0.180	1.60	Actin-binding protein	2FHA	183	0.186	1.90	Iron storage
1CDI	179	0.183	2.90	T-cell surface glycoprotein	1AMM	174	0.185	1.20	Crystallin
1CNV	179	0.183	2.90	Seed protein	3CHY	128	0.151	1.66	Signal transduction protein
1XNB	185	0.165	1.49	Glycosidase					

Table 2. (Continued)

PDB code	Residue length	*R* value	Resolution (Å)	Classification	PDB code	Residue length	*R* value	Resolution (Å)	Classification
				Residue length 200 < $L_s \leq$ 300					
1CYX	205	0.196	2.30	Electron transport	1AD2	228	0.203	1.90	Ribosomal protein
1AOL	228	0.223	2.00	Viral glycoprotein	1AP8	213	–	–	RNA cap
1BF8	205	–	–	Chaperone	1BJK	295	0.166	2.30	Oxidoreductase
1BYQ_A	228	0.189	1.50	Chaperone	1C3D	294	0.188	1.80	Complement
1AKZ	223	0.189	1.57	Glycosidase	1FTS	295	0.222	2.20	Signal recognition particle receptor
1EZM	301	0.176	1.50	Hydrolase	1CSN	298	0.207	2.00	Phosphotransferase
1JUK	248	0.177	2.50	Lyase	1KID	203	0.180	1.70	Chaperone
1MML	265	0.198	1.80	Reverse transcriptase	1MRJ	247	0.173	1.60	Ribosome-inactivating protein
1NLS	237	0.127	0.94	Agglutinin	1PPN	212	0.159	1.60	Hydrolase (sulfhydryl proteinase)
1RGS	288	0.217	2.80	Kinase	1RHS	296	0.159	1.36	Transferase
1THV	207	0.165	1.75	Sweet tasting protein	1VIN	268	0.216	2.00	Binding protein
2BAA	243	0.180	1.80	Hydrolase (*O*-Glycosyl)	1YUB	245	–	–	Methyltransferase
1ZIN	217	0.173	1.60	Phosphotransferase					
				Residue length $L_s >$ 300					
1A8E	329	0.181	1.60	Iron transport	1ADS	315	0.200	1.60	Oxidoreductase
16PK	415	0.188	1.60	Kinase	1AXN	323	0.177	1.78	Calcium/phospholipid-binding protein
1B0M	753	0.193	2.50	Hydrolase	1BG2	325	0.215	1.80	Motor protein
1BGP	309	0.192	1.90	Oxidoreductase	1BXO	323	0.099	0.95	Hydrolase
1DLC	584	0.183	2.50	Toxin	1IRK	306	0.196	2.10	Transferase (phosphotransferase)
1ISO	416	0.186	1.90	Oxidoreductase	1KVU	338	0.179	1.90	Isomerase
1MOQ	368	0.185	1.57	Glutamine amidotransferase	1SVB	395	0.183	1.90	Glycoprotein
1URO_A	367	0.184	1.80	Lyase	1ARV	344	0.178	1.60	Peroxidase
2CAE	484	0.194	2.00	Oxidoreductase	2DPG	485	0.173	2.50	Oxidoreductase
2PGD	482	0.198	2.00	Oxidoreductase	3GRS	478	0.186	1.54	Oxidoreductase (flavoenzyme)

Residue length, L_s, is the protein amino acids number.

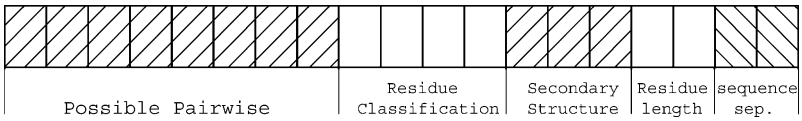

Figure 3. The binary input encoding scheme.

cludes two parts, i.e., the center encoding and the width encoding parts. The center encoding part has n genes to represent the centers and the width encoding part has $M \times N$ genes to denote the widths, where M denotes the pattern class number and N is the length of one width parameter. All the genes are encoded into a binary bit, 1's and 0's, which are used to denote whether the training samples are selected as the centers of the hidden neurons or not. The fitness function described in [10] is also used in our study, i.e.:

$$f(\varepsilon, n_c) = \frac{(b+1)n}{2^b n_c}, \qquad b = sign(\varepsilon - e) \quad (4)$$

where ε is the preset error; n is the number of the training samples; e is the practical root mean squared error; n_c represents the number of the selected hidden centers, and it can be described as follows:

$$n_c = \sum_{k}^{l} X_i(k), \qquad n_c \in (1, N) \qquad (5)$$

Here, we call X_i the ith individual of population and l the corresponding length. In addition, the general genetic operators in [10, 20, 22], are adopted to optimize the RBFNN parameters.

Binary input encoding scheme and training algorithm

Data preparation

The 173 non-homologous proteins of known 3D structure, described in [3, 13], got from the protein data bank (PDB) are adopted to train and test the GA optimized RBFNN for the purpose of learning and predicting the inter-residue contact patterns. All the the proteins (labeled by PDB code) are extracted form the PDB [23]. Table 1 gives the 4 test protein sequences, and Table 2 illustrates the rest 169 sequences of the 173 non-homologous proteins. The lists include all proteins with percentage of sequence identity lower than 25% and whose chain was not interrupted [3]. The 173 protein sequences are grouped into 4 classes: $L_s < 100$, $100 \le L_s < 200$, $200 \le L_s < 300$ and $L_s \ge 300$, according to their residues length. The four protein sequences, namely 1SPY ($L_s = 89$), 5P21 ($L_s = 166$), 1CEN ($L_s = 270$) and 1YSC ($L_s = 421$), were selected to test the trained optimized RBFNN. What should be mentioned is that this study mainly concentrates on the utility of our proposed binary encoding scheme and GA-optimized RBFNN for contact prediction problem, so only the above 4 randomly selected sequences are

Table 3. The binary encoding scheme of the input sample space.

Possible pairwise	Binary (8 bits)	Residue classification	Binary (4 bits)	Secondary structure	Binary (3 bits)	Residue length	Binary (2 bits)	Sequence separation	Binary (2 bits)
Ala–Ala	00000000	Np–Np	0000	α–α	000	$L_s < 100$	00	(0, 4)	00
Ala–Cys	00000001	Np–P	0001	α–β	001	$100 \le L_s < 200$	01	(7, 14)	10
Ala–Asp	00000010	Np–A	0010	α–Coil	010	$200 \le L_s < 300$	10	(14, α)	11
Ala–Glu	00000011	Np–B	0011	β–β	011	$L_s \ge 300$	11		
Ala–Phe	00000100	P–P	0100	β–Coil	100				
Ala–Gly	00000101	P–A	0101	Coil–coil	101				
⋮	⋮	P–B	0110						
Ala–Tpy	00010100	A–A	0111						
⋮	⋮	A–B	1000						
Tpy–Tpy	11010010	B–B	1001						

In the residue classification column: Np → nonpolar-hydrophobic; P → polar-hydrophilic; A → acidic and B → basic. In sequence separation column, we adopt the definition of [36], grouped the residue interactions as long (whose sequence separation is greater than or equals to 14 residues), medium (6,13) and short (1,5) range contact.

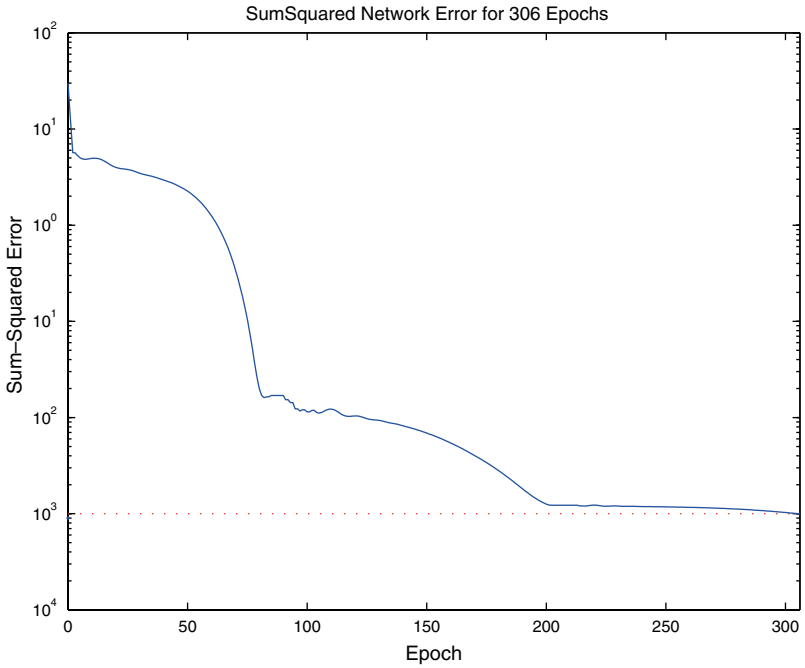

Figure 4. The sum-squared error for the RBFNN during the training procedure of inter-residues contact map prediction problem.

used to evaluate the prediction performance, and further work on large scale testing sets will be conducted in future work.

Input encoding scheme

Here, we use binary encoding scheme to denote the input vectors for every possible residue pairwise. Specifically, each possible pairwise is encoded by a $19(8+4+3+2+2)$ bit binary string as illustrated in Figure 3. The 19 bit binary string consists of 5 parts which represent possible residue pair, residue classification, secondary structure, residue length and sequence separation information, respectively.

Possible residues pairwise (8 bits for the 210 possible pairwises): As we know, the 20 different amino acid residues have 210 $(20 \times (20+1)/2 = 210)$ possible couples (considering each possible residue couple and its symmetric to be coded in the same way) [3], such as the couple of Ala–Ala, Ala–Cys, Ala–Asp, and so on. So, we can denote these 210 possible couples by a 8 bits binary, which is shown in Table 3.

Residue classification (4 bits for the 10 possible class pairs): Residues may be classified into 4 classes, i.e., nonpolar-hydrophobic, polar-hydrophilic, acidic or basic. So for a given pair of residues, there are totally 10 possible pair cases, such as both nonpolar-hydrophobic, nonpolar-hydrophobic to acidic, and so on. In this case, a given residue pair classification information can be denoted by a 4-bit binary as shown in Table 3.

Secondary structure (3 bits for the 6 possible structure pairs): Studies show that the residue spatial structures, such as secondary structure and tertiary structure, are important factors of residue contact. Usually, protein secondary structures are categorized into 8 classes [24]: H (α-helix), G (3_{10} helix), I (π-hleix), E (extended β-strand), B (isolated β-strand), T (turn), S (bend), and coil ("*"). Here we use the definition of [25] to reduce these 8 classes into 3 categories: α-helix $= \{H, G, I\}$; β-sheets $= \{E, B\}$; coil $= \{T, S, *\}$. So, a residue has three possible secondary, and we can use a 3-bit binary to denote the 6 possible secondary structure pairs (α–α, α–β, α-coil, β–β, β-coil and coil–coil). Considering the application in practice, predicted structure information should be adopted in the encoding process, but here, for the purpose of reducing computational complexity, we only use the real structure got from PDB instead of predicted information. Table 3 gives the detailed encoding elements.

Table 4. Numbers and accuracies of predicted contacts map.

Sequence PDB code (Length)	5 Å			6 Å			7 Å			8 Å			Average accuracy
	N_p	N_d	A (%)	N_p	N_d	A (%)	N_p	N_d	A (%)	N_p	N_d	A (%)	
1SPY (89)	187	637	29.36	313	1035	30.24	448	1457	30.75	505	1783	28.32	29.67
5P21 (166)	173	580	29.83	327	1030	31.75	484	1468	32.97	574	1766	32.50	31.45
1CNE (270)	346	1090	31.74	578	1758	**32.88**	829	2524	32.84	984	3180	30.94	**32.10**
1YSC (421)	554	1829	30.29	1305	4179	31.23	1415	4523	31.28	1657	5785	28.64	30.36
Sum & accuracy	1260	4136	30.46	2523	8002	31.53	3176	9972	32.00	3720	12514	29.73	

N_p– the predicted contacts numbers; N_d– the desired contacts numbers; accuracy: $A = N_p/N_d$.

Residue length (2 bits for the 4 cases): Previous studies shown that the protein sequence residue length is one of important determinants of predicted accuracy. The sequences we used as training and testing sets are divided into 4 groups, $L_s \leq 100$, $100 < L_s \leq 200$, $200 < L_s \leq 300$ and $L_s \geq 300$, according to their corresponding residue length which are shown in Table 2. So, we can denote these 4 residue length regions by a 2-bit binary string.

Sequence separation (2 bits for the 3 cases): Sequence separation is the distance between two amino acids in the protein sequence. The contacts between two residues that are close in space and far in the sequence are termed as long-, medium- and short-range contacts. Here, we adopt the definition of [36] and divide the contacts of two interesting amino acid residues, s_i and s_j, as:

$$\text{Contact} = \begin{cases} \text{long-range } |i-j| \geq 14 \\ \text{medium-range } 5 < |i-j| \leq 14 \\ \text{short-range } |i-j| \leq 5 \end{cases} \tag{6}$$

and a 2-bit binary string is used to denote the sequence separation information.

Training algorithm

The training procedure starts with an randomly preset weights with an value between 0 and 1. The general purpose is to adjust the weights $\{w_{mj}\}$ to minimize the sum squared error (SSE) between the actual outputs z_j^q and the target output values t_j^q, which is defined as follows:

$$E = \sum_{q=1}^{Q} \sum_{j=1}^{J} (z_j^q - t_j^q)^2 \tag{7}$$

and the weights are adjusted according to

$$w_{mj}^{i+1} = w_{mj}^{i} - \alpha(\partial E / \partial w_{mj}) \tag{8}$$

here we call α the learning rate.

Moreover, the detailed training algorithm can be described as follows:

Step 1. Input all the exemplar feature vectors $\{x^q : q = 1, 2, \ldots, Q\}$ and their corresponding desired values $\{t^q : q = 1, 2, \ldots, Q\}$, and preset the iteration number $i = 0$.

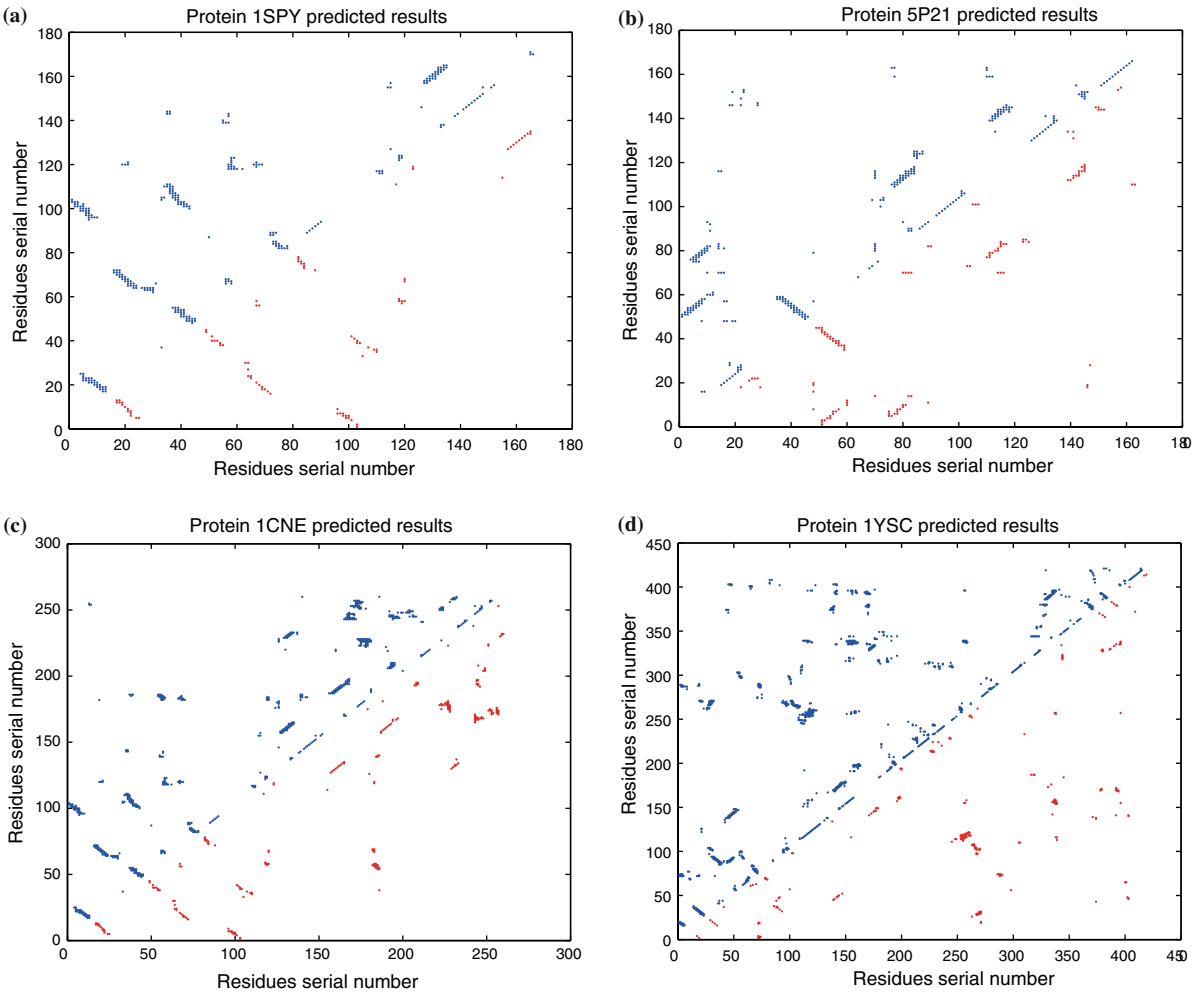

Figure 5. The predicted and desired inter-residue contacts maps (with a threshold of 7Å) for the 4 protein sequences (1SPY, 5P21, 1CNE and 1YSC). The red squares in the lower-right triangle denote the predicted map, while the blue squares of the upper-left triangle represent the desired map.

Step 2. Input an integer value M for the number of exemplar vectors to be used as centers.
Step 3. Randomly select initial weights $\{w_{mj}\}$ and bias $\{b_j\}$ between 0 and 1.
Step 4. Compute all the outputs $\{z_j^q\}$ for all j and q according to Equation 3.
Step 5. Calculate the SSE value, E_0, according to Equation 4.
Step 6. Adjust the weights w_{mj} according to Equation 8
Step 7. Compute a new value for E (Equation 4)
Step 8. If $|E - E_0| < \varepsilon$ (the preset error) then stop the training procedure, otherwise, we set $E_0 = E$ and repeat steps 5–8.

In the output layer, we use a cutoff, $t = 0.5$, to determine the concerning two residues are in contact or not, i.e., if the output value greater than or equal to t, we consider the two residues are in contact, otherwise, the two residues are not in contact.

Simulation results and discussion

The predictor RBFNN is a standard feedforward network, whose input layer has 19 nodes as shown in Table 3, 34 hidden units and a single output. The expected output is 1 for contacts and 0 for non-contacts. All the experimental results are

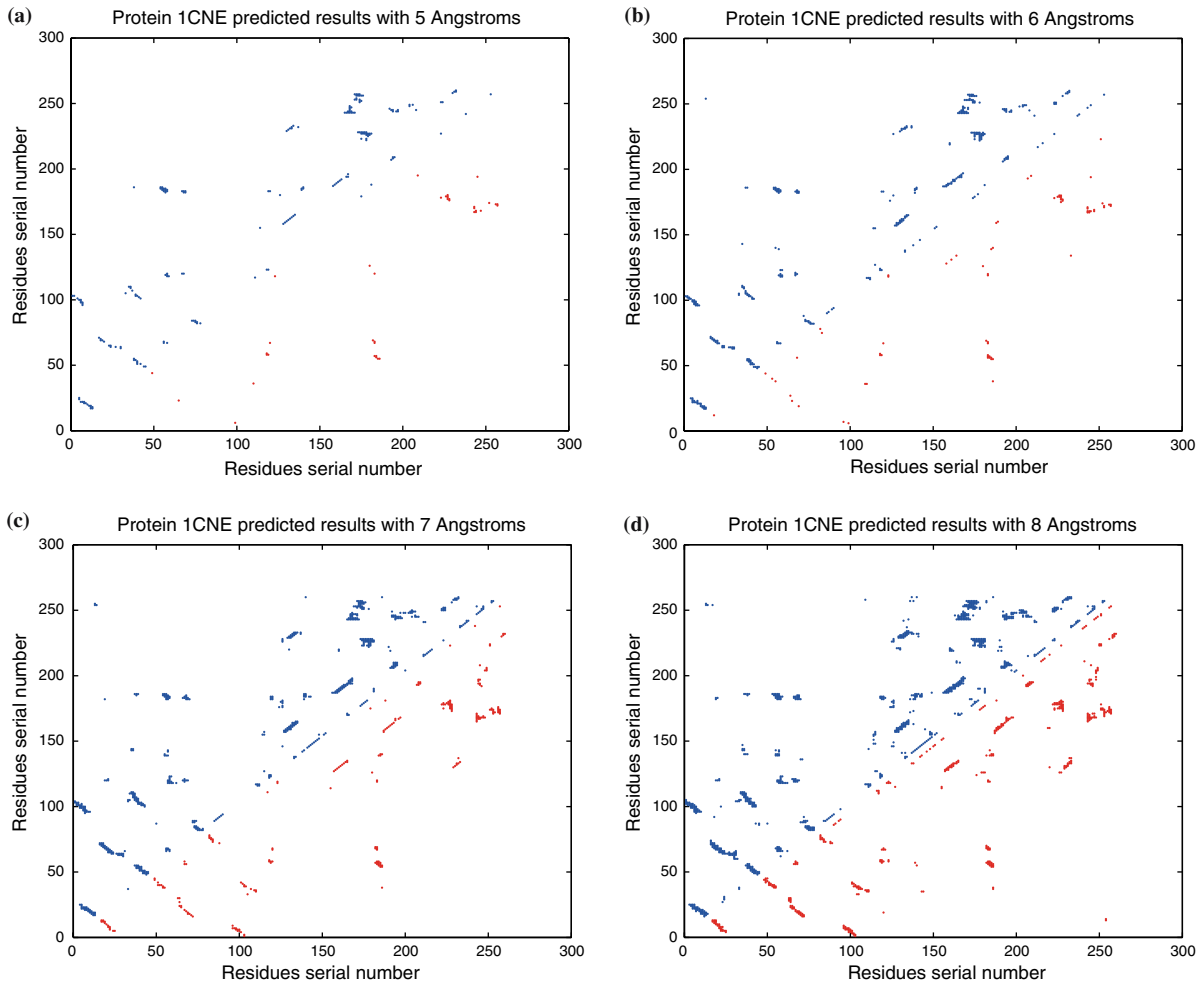

Figure 6. The 4 predicted and desired maps of the protein 1CNE with different contact thresholds of 5, 6 , 7 and 8 Å. For notations, please see Figure 5.

based on the neural network toolbox of MATLAB version 6.5.

Sum squared error vs. network iterations

Figure 4 shows the relationship of the sum squared error curve and iterations by using the proposed GA optimized RBFNN in the learning contact patterns procedure. From Figure 4, we can see that the network sum-squared error could reach the preset error, $\varepsilon = 0.001$, after about 300 iterations. This demonstrates that our proposed GA optimized RBFNN can learn the contact patterns effectively and efficiently.

Predicted and desired numbers

For a given test protein, we define the prediction accuracy A_p to be $A_p = N_p/N_d$, where N_p is the number of the correctly predicted contacts residues and N_d is the desired contacts number. The four protein sequences, named 1SPY, 5P21, 1CNE and 1YSC, which are shown in Table 1, are used to test the proposed RBFNN. The correctly predicted contacts numbers and the corresponding accuracies are given in Table 4, from which we can see that the overall performance of the network with the contact threshold of 7 Å scores higher than the other three, 5, 6 and 8 Å, and gets about the accuracy of 32%. At the same time, it can be

found that the network got the best predictig accuracy, 32.88%, when the contact threshold is equal to 6 Å and the protein residue length falls into the area of (200, 300). Meanwhile, the network average accuracy got the best (about 32.10%) to the protein 1CNE ($L_s = 270$). This demonstrates that the neural network technique is fit for the proteins whose residue length falls into the range of (200, 300), and the proposed RBFNN could get better performance with a contact threshold of 6 or 7 Å.

Contacts map prediction results

Here, the four protein predicted contact maps are shown in Figures 5 and 6. Figure 5 gives the predicted and desired maps of the 4 testing proteins with a threshold of 7 Å, while Figure 6 gives the maps with different contact thresholds: 5, 6, 7 and 7 Å for the protein 1CNE. In the figures, the upper-left triangle represent the desired contacts maps, and the lower-right area is the predicted maps based on our proposed binary encoding scheme and the GA optimized RBFNN. From the figures, it can be seen that our proposed binary input strategy is effective and efficient in the problem of inter-residue contacts map prediction.

Conclusions

In this paper, we proposed a new binary input encoding scheme for the purpose of training radial basis function neural network to predict inter-residue contacts map within the protein sequences. The computational results indicate that the prediction accuracy with the contact threshold of 7 Å scores highest, and the accuracy decreases as the protein residue length increases. Future works will include protein inter-residue distance distribution analysis and tertiary structure prediction.

References

1. Park, K., Vendruscolo, M. and Domany, E., Proteins Struct. Funct. Genet., 40 (2000) 237.
2. Vendruscolo, M., Najmanovich, R. and Domany, E., Proteins Struct. Funct. Genet. 38 (2000) 134.
3. Fariselli, P., Olmea, O., Valencia, A. and Casadio, R., Protein Eng., 14(11) (2001) 835.
4. Hamilton, M., Burrage, K., Raga, M.A. and Hubor, T., Protein contact prediction using patterns of correlation, http://foo.maths.uq.edu.au/nick/Protein/Dataset/protein_ contact.pdf.
5. Hu, J., Shen, X., Shao, Y., Bystroff, C. and Zaki, M. J., Mining Protein Contact Maps, http://www.cs.rpi.edu/zaki/BIOKDD02/02-hu.pdf.
6. Singer, M.S., Vriend, G. and Bywater, R.P., Protein Eng., 15(9) (2002) 721.
7. Fairchild, S., Pachter, R. and Perrin, R. Math. J., 5(4), (1997) 64.
8. Ketterman, A.J., Prommeenate, P., Boonchauy, C., Chanama, U., Leetachewa, S., Plomtet, N., Prapanthadasa, L., Biochem. Mol. Biol., 31 (2001) 65.
9. Zhang, G.Z., Huang, D.S. and Wang, H.Q., In the Proceedings of the 2004 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montréal, Québec, Canada, May 17–21, 2004, pp. V573–576.
10. Zhang, G.-Z. and Huang, D.S., In the Proceedings of the 2004 IEEE Congress on Evolutionary Computation, Portland, Oregon, Vol. 1, June 19–23, 2004, pp. 1015–1019.
11. Sonnhammer, E.L.L. and Wootton, J.C., J. Mol. Graph. Model., 16 (1998) 1.
12. Mirny, L. and Domany, E., Proteins, 26 (1996) 319.
13. Fariselli, P., Olmea, O., Valencia, A. and Casadio, R., Proteins, (Suppl 5) (2001) 157.
14. Fariselli, P. and Casadio, R., Protein Eng., 12 (1999) 15.
15. Wang, Z.O. and Zhu, T., Neural Networks, 13 (2000) 545.
16. Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B. and Thornton, J.M., Structure, 5(8) (1997) 1093.
17. Belloir, F., Fache, A. and Billat, A. In the Proceedings of the European Symposiun on Artificial Neural Networks, Bruges (Belgium), D-Facto publicl., ISBN 2-600049-9-X, April 21–23 1999, pp. 399–404.
18. De Jong, K., Learning with Genetic Algorithms: An Overview, Machine Learning Vol. 3, Kluwer Academic Publishers, 1998.
19. Cho, S.-B., Fuzzy Sets Syst., 103 (1999) 339.
20. Guo, L., Huang D.S. and Zhao, W., IEE Electronics Lett., 39(22) (2003) 1600.
21. Guo, L., Huang, D.S. and Zhao W.-B., In the Proceedings of the International Joint Conference on Neural Networks (IJCNN2003), Portland, Oregon, July 20–24, 2003, pp. 3213–3217.
22. Zhang, G.-Z. and Huang, D.S., The 2004 International Joint Conference on Neural Networks (IJCNN2004), Budapest, Hungary, July 25–29, 2004.
23. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne P.E., Nucleic Acids Res., 28 (2000) 235.
24. Kabsch, W. and Sander, C., Biopolymers, 2577 (1983) 22.
25. Guo, J., Chen, H., Sun, Z. and Lin, Y., Proteins Struct. Funct. Bioinform., 54 (2004) 738.
26. Benedetti, G. and Morosetti, S., Biophys. Chem., 55 (1995) 253.
27. Yu, H. and Liang, W., Comput. Indus. Eng. 39 (2001) 337.
28. Huang, D.S. and Ma, S.D., J. Intel. Syst., 9(1) (1999) 1.
29. Pearl, F.M.G., Lee, D., Bray, J.E., Sillitoe, I., Todd, A.E., Harrison, A.P., Thornton, J.M. and Orengo, C.A., Nucleic Acids Res., 28(1) (2000) 277.

30. Huang, D.S., Int. J. Pattern Recog. Artif. Intel., 13(6) (1999) 945.
31. Sen, S., Biophys. Chem., 103 (2003) 35.
32. Kumarevel, T.S., Gromiba, M.M. and Ponnuswamy, M.N., Biophys. Chem., 88 (2000) 81.
33. Zhang, L.X., Li, J., Jiang, Z. and Xia, A., Polymer, 44 (2003) 1751.
34. Michael Gromiha, M. and Selvaraj, S., J. Mol. Biol. 310 (2001) 27.
35. Singer, M.S., Vriend, G. and Bywter, R.P., Protein Eng., 15 (2002) 721.
36. Park, K.-H., Suk, J.-E., Jacobsen R., Gray, W.R., Mcintosh, J.M. and Han, K.-H., J. Biol. Chem., 276(52) (2001) 49028.