An improved Genetic Algorithm for statistical potential function design and protein structure prediction

Xin Geng and Jihong Guan*

Department of Computer Science and Technology,
Tongji University,
Shanghai 201804, China
E-mail: gengxin@tongji.edu.cn
E-mail: jhguan@tongji.edu.cn
*Corresponding author

Qiwen Dong and Shuigeng Zhou

Shanghai Key Lab of Intelligent Information Processing,
Fudan University,
Shanghai 200433, China

and

School of Computer Science,
Fudan University,
Shanghai 200433, China
E-mail: qwdong@fudan.edu.cn
E-mail: sgzhou@fudan.edu.cnv

Abstract: Protein structure prediction is an important but far from being well-resolved problem in computational biology. It is generally regarded that the native structures of proteins correspond to minimum-energy states. Potential functions are useful in protein structure prediction. To obtain the optimal parameters of protein potential functions, we introduced several strategies to improve the basic Genetic Algorithm (GA). The improved GA was employed in statistical potential function design and protein structure prediction, and experimental results validate the effectiveness and efficiency of the proposed algorithm.

Keywords: GA; genetic algorithm; statistical potential function; structure prediction; amino acid; dihedral angles.

Reference to this paper should be made as follows: Geng, X., Guan, J., Dong, Q. and Zhou, S. (2012) 'An Improved Genetic Algorithm for statistical potential function design and protein structure prediction', *Int. J. Data Mining and Bioinformatics*, Vol. 6, No. 2, pp.162–177.

Biographical notes: Xin Geng is a PhD student at Tongji University, under the supervision of Prof. Jihong Guan. Her research focuses on protein structure prediction.

Jihong Guan is now a Professor of Department of Computer Science and Technology, Tongji University, Shanghai, China. She received her Bachelor degree from Huazhong Normal University in 1991, her Master degree from Wuhan Technical University of Surveying and Mapping (merged into Wuhan University since August 2000) in 1991, and her PhD from Wuhan University in 2002. Her research interests include bioinformatics, databases, data mining, distributed computing and geographic information systems. She has published more than 100 papers in domestic and international journals and conferences.

Qiwen Dong received his BE degree, ME degree and PhD all from Harbin Institute of Technology, China. Currently he is a post doctor in FuDan University. His research interests include computational investigation of sequence-structure function relationships in proteins and language model of biological sequence.

Shuigeng Zhou is a Professor of the School of Computer Science, Fudan University, Shanghai, China. He received his Bachelor degree of Electronic Engineering from Huazhong University of Science and Technology in 1988, his Master degree of Electronic Engineering from University of Electronic Science and Technology of China in 1991, and his PhD of Computer Science from Fudan University in 2000. He was a postdoctoral researcher in State Key Lab of Software Engineering, Wuhan University from 2000 to 2002. His research interests include Bioinformatics, data management, mining and learning over massive datasets. Currently he is member of IEEE, ACM-SIGMOD and IEICE.

1 Introduction

The potential function is used to estimate the energy of proteins, which is a key factor for protein structure prediction. There are two major methods to build potential functions: one is based on the fundamental laws of physics, and the other is based on experimental data of known protein structures. Due to the heavy computation resource requirement of the physics based method, knowledge-based potential functions have been successfully applied to various issues of protein structure prediction, such as folding recognition and *ab initio* prediction (Sippl, 1995). In the literature, the potential function is also referred to as 'force field' or 'scoring function'.

According to the thermo-dynamic hypothesis formulated by Anfinsen (1973), the native structure of a protein corresponds to the global minimum of its free energy under given conditions. Researchers have designed several types of knowledge-based potentials. These potentials can be roughly classified into atom level (Lu et al., 2008), residue level (Melo and Feytmans, 1997) and profile level potentials (Dong et al., 2006), or linear and non-linear potentials according to different criteria (Dong and Zhou, 2010). In this work, we discuss the empirical probability function proposed by Dong et al. (2009) on Hypothesis 5, that is, the occurrence of each conformational state is dependent on the neighbour residues, and the probability under multiple conditions can be evaluated by the weight sum of the probabilities under each individual condition.

The rest of this paper is organised as follows. In Section 2, the related work for finding the optimal parameters of potential functions is surveyed. In Section 3, we present the technical detail about how to improve the traditional GA to efficiently search the optimal parameters of potential functions, including how to derive the formula of the potential function from the primary sequence and dihedral angles. In Section 4, we give the results of experimental evaluation and some discussions. The conclusion will be drawn in Section 5.

2 Related work

A number of methods and tools have been developed to optimise parameters of potential functions. By assuming a hierarchical structure of the energy landscape, Liwo et al. (2002) proposed an iterative algorithm that is comprised of seven steps to obtain the best set of parameters of the United-Residues (UNRES) energy function (Liwo et al., 1997). This algorithm can find the native structure faster than the previous optimisation methods then, especially for the potential functions of proteins with complicated topologies. For minimising Z scores and getting a relatively high correlation between the energy parameters, Fujitsuka et al. (2004) used a Monte Carlo search with simulated annealling to find a set of parameters of a physical energy function based on the energy landscape theory. Other efforts of potential function design are mainly based on the recognition of native structures of proteins and minimise the energies of the structures in native states (Qiu and Elber, 2005). Wagner et al. (2004) presented a large-scale optimisation technique to model the energy function and employed Linear Programming (LP) to identify parameters in the energy function that models proteins correctly. Their results show that the technique can improve the quality of potential functions. Tobi et al. (2000) reported the optimisation of distance-dependent, pair potentials for protein folding based on linear optimisation. The contact potential function created by using LP approach is comparable with contact potentials obtained with other approaches. Recently, several distance-dependent, knowledge-based potential functions at atom level have been developed (Qiu and Elber, 2005; Rajgaria et al., 2000, 2008). In order to determine the optimal parameters, a set of linear inequalities with the same form as equation (8) in Section 3.1 have to be solved. Moreover, they consider a number of physical constraints to guarantee a physically realistic solution. Thus, LP is the ideal technique for identifying the parameters of a potential function.

Despite of the success of LP in previous potential function design, there are some drawbacks with the LP approach in our scenario. First, the LP approach requires considerably computation resources to obtain the solution. For example, the CPLEX (a tool for solving linear optimisation problems) LP package should be executed on an Intel Pentium-4, 3.2 GHz workstation with 4 gigabytes of RAM (Rajgaria et al., 2000). Second, we use a simplified model to represent the tertiary structures of proteins without considering the physical properties of the amino acids, and only eight parameters are needed to determine, while millions or tens of millions of constraints are included and 2730 parameters are needed to determine in Tobi et al. (2000). Based on the above consideration, LP is not suitable for our work, thus we turn to a randomised algorithm – GA (Goldberg and David, 1989). Instead of solving the inequalities directly by LP approach, we use GA to find the optimal W such that most protein sequences can satisfy the inequalities.

Inspired by the principle of "the survival of the fittest" proposed by Charles Darwin, GA has been the most popular technique in evolutionary computation research (Eiben and Smith, 2003). As a type of directed random search techniques, GAs have been applied to various problems with promising results, e.g., tuning of the structure and parameters of a neural network (Leung et al., 1997), Web services selection (Su et al., 2007), and protein fold recognition (König and Dandekar, 1999). Different improvement strategies were proposed for different problems in order to obtain the local optimal solutions. And many modifications are employed to handle the intrinsic disadvantages of the standard GA for improving the performance. Reserve selection was presented in Chen et al. (2007) to prevent premature convergence and the improved algorithm can solve complex and large-scale problems more efficiently. Kumar et al. (2007) discussed various mutation operators for boosting efficiency. Etaner-Uyar and Harmanci (2002) got better performance by combining standard GA with hill climbing. Wang and Chen (2005) employed relation matrix coding scheme to enhance the convergence of GAs.

For the problem addressed in this study, there is little information about the solution and it is difficult to obtain its analytical solution. As a new application scenario, the previous strategies are not suitable. We modified the basic GA to get the optimal parameters of a potential function, and make it sufficiently accurate so that the native state of a protein has the lowest statistical potential. In the next section, we will present the detail of how to obtain the optimal weights of the potential function based on the improved GA.

3 Method

3.1 The potential function

A good potential function should have a set of free energy components that are inexpensive to compute, yet sufficiently accurate to ensure the minimum energy of the native conformation (Tobi et al., 2000). Here, we use the backbone dihedral angles ϕ and ψ, and the side-chain dihedral angle χ_1 as the structural parameters to represent local interactions. Shortle (2002) indicated that the logarithm of the propensity defined as a conditional probability of a given structural parameter, approximates the free energy of residues. He assumed that the free energy for each residue position is additive, so the score of a sequence fragment can be calculated by adding up the log of the propensity at each position. Similarly, we define the scoring function F of a protein sequence as the sum of probabilities of ϕ/ψ pairs at different positions, that is,

$$F = \sum_i \log p_i(s) \tag{1}$$

where $p_i(s)$ represents the probability of ϕ/ψ pair at the i-th position of the sequence. Contrary to Flory's isolated-pair hypothesis (Flory, 1969), the results of Pappu et al. (2000), and Ohkubo and Brooks III (2003) show that local structural interactions exist between each amino acid and its immediate neighbours. Consequently, the occurrence of current ϕ/ψ pair is dependent on the anterior and posterior residues, ϕ/ψ pairs and χ_1's, as well as the current residue and χ_1. Thus, equation (1) can be rewritten as

$$F = \sum_i \log P(s_i | a_{i-1} c_{i-1} s_{i-1} a_i c_i a_{i+1} c_{i+1} s_{i+1}) \tag{2}$$

where s_j, a_j and c_j ($j = i-1,\ i,\ i+1$) represent the ϕ/ψ pairs, amino acids and χ_1's at the jth position of the sequence, respectively. In order to calculate the probabilities in equation (2) based on the statistical analysis of the data, we make some simplification. First, each backbone dihedral angles pair ϕ/ψ is mapped to a region in the Ramachandran plot as shown in Figure 1 (Shortle, 2002). In such a way, all the ϕ/ψ pairs taking numeric values are classified into 15 classes. The side chain dihedral angles χ_1's are also labelled by 3 classes corresponding to the ranges -120 to +120, 0 to +120, and 0 to -120 (Shortle, 2002). Then, the probabilities in equation (2) will be calculated for $7.29 \times 10^8 (15 \times 20 \times 3 \times 15 \times 20 \times 3 \times 20 \times 3 \times 15)$ discrete states. Second, to avoid the data sparse problem in calculating the probabilities in equation (2), we make an assumption that the probability under multiple conditions is the weighted sum of each individual probability, that is, all conditions are independent from each other. Then we have

$$\begin{aligned} &P(s_i | a_{i-1} c_{i-1} s_{i-1} a_i c_i a_{i+1} c_{i+1} s_{i+1}) \\ &\quad = w_1 p(s_i | a_{i-1}) + w_2 p(s_i | c_{i-1}) + w_3 p(s_i | s_{i-1}) + w_4 p(s_i | a_i) \\ &\quad + w_5 p(s_i | c_i) + w_6 p(s_i | a_{i+1}) + w_7 p(s_i | s_{i+1}) + w_8 p(s_i | c_{i+1}), \end{aligned} \tag{3}$$

and

$$\sum_i w_i = 1. \tag{4}$$

Hence, we have 1485 probabilities and 8 weight values to be determined. For the simplification of representation, we introduce two vectors. One is the weight vector W with the following form

$$W = (w_1,\ w_2,\ w_3,\ w_4,\ w_5,\ w_6,\ w_7,\ w_8), \tag{5}$$

and the other is the probability vector P_i with each element corresponding to an individual probability, that is,

$$\begin{aligned} P_i = (&p(s_i | a_{i-1}),\ p(s_i | c_{i-1}),\ p(s_i | s_{i-1}),\ p(s_i | a_i),\ p(s_i | c_i),\ p(s_i | a_{i+1}), \\ &p(s_i | s_{i+1}),\ p(s_i | c_{i+1})). \end{aligned} \tag{6}$$

Substituting the two vectors into equation (3), and then equation (3) into equation (2), we get the final scoring function as follows:

$$F = \sum_i \log < W, P_i^T > \tag{7}$$

where $<\cdot,\cdot>$ denotes inner product and P_i^T represents the transpose of P_i, which can be gotten by maximum likelihood estimation from the given data set. As for W, which is the focus of this work, we develop an improved GA that is robust and effective to find the optimal W. Our goal is to find the W such that there are as many as possible training proteins whose native structures have the minimum scoring function values. That is, we are to solve the following inequality:

$$F(S_n, X_i; W) - F(S_n, X_n; W) > 0. \tag{8}$$

Above, X_i is the set of decoy structures, X_n and S_n are the set of native structures and the corresponding animo acid sequence.

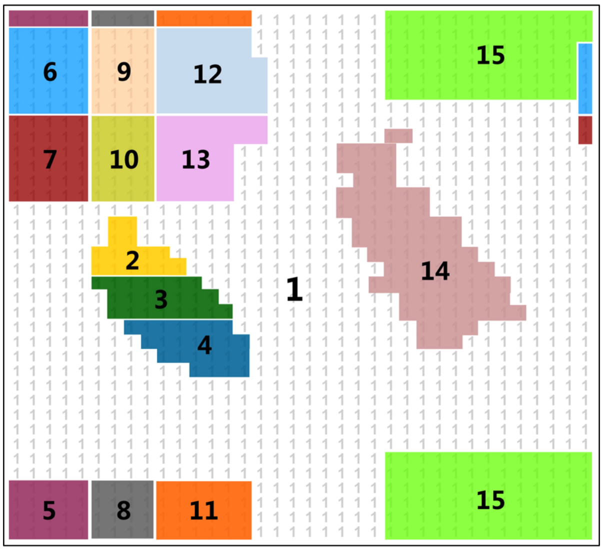
3.2 Conditional probabilities calculation

Obviously, conditional probabilities are the important ingredients of the statistical potential function in equation (7). By maximum likelihood estimation, these conditional probabilities can be estimated approximately as

$$P(s_i|B) = \frac{C(s_i, B)}{C(B)} \tag{9}$$

from dataset PDB25 (see the detail in Section 4.1). $C(B)$ represents the number of occurrences of certain condition B in PDB25, and $C(s_i, B)$ represents the number of co-occurrences of s_i and B in the protein sequences of PDB25. The backbone dihedral angles ϕ/ψ pairs and side chain dihedral angle χ_1's, ranging from $-180°$ to $180°$, are calculated by using the DSSP program (Kabsch and Sander, 1983). After the processing by DSSP, the backbone dihedral angle pair of each amino acid is

Figure 1 The 15 regions of the Ramachandran plot and the subdivisions used in this paper (see online version for colours)



marked by the region index in terms of the different conformation states in Figure 1 and χ_1 is labelled by 0, 1 or 2 according to three different ranges. equation (9) can be used to calculate all probabilities in the potential function. Take the segment "$\cdots TPREKDKLLLFT \cdots$" for example, the backbone dihedral angle pairs and side chain dihedral angles can be finally represented by "$\cdots 1133323223333 \cdots$" and "$\cdots 101002002220 \cdots$".

To avoid the occurrence of $C(B)$ being zero in equation (9), we set the initial values of both $C(s_i, B)$ and $C(B)$ to 1.

3.3 *An improved Genetic Algorithm*

The standard GA consists of an encoding mechanism for the parameters, a population of data points, a stochastic evolution rule and some simple operations on the objective function (Chen et al., 2007). The procedure of a standard GA is outlined in Algorithm 1. Certainly, minor changes will be made according to the requirements of applications.

Algorithm 1 Standard Genetic Algorithm

```
   Algorithm: Standard Genetic Algorithm STATE
Begin
STATE Initialize population;
FORevery individual STATE evaluate the fitness function value;
WHILEtermination condition is false STATE select individuals;
STATE perform genetic operation;
STATE generate new population;
STATE evaluate fitness function value of new generation;
ENDWHILE ENDFOR
```

In this paper, there are two constrains for W in equation (7).

1 *Continuous solution space*: The solution is an 8-dimensional vector and each element of the vector is a real number between 0 to 1.

2 *Normalisation*: The sum of all elements of the vector W is equal to 1.

Considering these two constrains, we introduce some improvements on the standard GA in order to search the optimal parameters efficiently. Next, we will present the detail of the improved algorithm.

3.3.1 *Initial population*

Let $\{p_1, p_2, \ldots, p_n\}$ (n is the size of population) denote the population, each individual $p_i = (p_{i1}, p_{i2}, p_{i3}, p_{i4}, p_{i5}, p_{i6}, p_{i7}, p_{i8})$ where $p_{ij} (j = 1, \ldots, 8)$ is a gene representation of the j-th parameter. The initial population is generated randomly. We expect that the final solution can make the scoring function recognise the native states of proteins as many as possible. Taking the above two constrains into consideration, we adopt the floating encoding mechanism for the eight components of p_i, which is more suitable than binary-encoding in this paper.

Each element in p_i is first randomly assigned a float number and then normalisation is imposed on it as follows:

$$p_{ij} = \begin{cases} \dfrac{p_{ij}}{\sum_{j=1}^{8} p_{ij}}, & \text{if } \sum_{j=1}^{8} p_{ij} \neq 0; \\ \dfrac{1}{8}, & \text{otherwise} \end{cases} \tag{10}$$

which will map p_{ij} to [0,1].

Certainly, the genetic operations of floating encoding are more complex than those of binary-encoding. In Section 3.3.4, we will describe genetic operations used in our algorithm.

3.3.2 Fitness function

The more proteins whose native structures are recognised, the better the potential function is thought to be. We use the percentage of the proteins whose native states are recognised correctly over the total proteins used in the study as the fitness function, and the expected solution is these parameters that make the fitness function get globally maximal value. For each individual of the population, we calculate the value of formula (7) to see how many proteins with native states can be correctly recognised. For each protein, we define a number X_l taking value in $\{0, 1\}$ to represent whether or not the native structure reaches minimum and let f_i denote the value of fitness function. X_l and f_i are defined as follows:

$$X_l = \begin{cases} 1 & \text{if the score of native structure is minimum} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

and

$$f_i = \frac{\sum_{l=1}^{k} X_l}{k} \tag{12}$$

where k represents the total number of proteins.

3.3.3 Selection

As a directly random search technique, to perform genetic operations, GA selects pairs of individuals probabilistically based upon their fitness values for reproduction (Goldberg and David, 1989). We use the elitist selection strategy. Just like previous work, an individual i is chosen with the probability $\tau_i = \frac{f_i}{\sum_{i=0}^{n} f_i}$ if $(\sum_{i=0}^{n} f_i) \neq 0$; otherwise $\tau_i = \frac{1}{n}$ by the method of spinning the roulette wheel (Leung et al., 1997).

3.3.4 Genetic operations

As the core integrant of GA, genetic operations include crossover and mutation operations. Through these two operations, the superior genes are preserved while the inferior ones are weeded out.

Crossover operation is performed on each selected pairs of individuals to exchange the gene information between each other. As only eight genes contained in each individual, here we employ one-point crossover operation, that is, all genes beyond that point in either individual are swapped between the two parent individuals. The parents $p_m = (p_{m1},\ p_{m2},\ p_{m3},\ p_{m4},\ p_{m5},\ p_{m6},\ p_{m7},\ p_{m8})$ and $p_n = (p_{n1},\ p_{n2},\ p_{n3},\ p_{n4},\ p_{n5},\ p_{n6},\ p_{n7},\ p_{n8})$ are replaced by the offspring produced by them, p'_m and p'_n as expressed in equations (13) and (14), where index $r \in [1, 8]$ is a randomly generated integer:

$$p'_m = (p_{m1}, \cdots,\ p_{mr},\ p_{n(r+1)}, \cdots,\ p_{n8}), \tag{13}$$

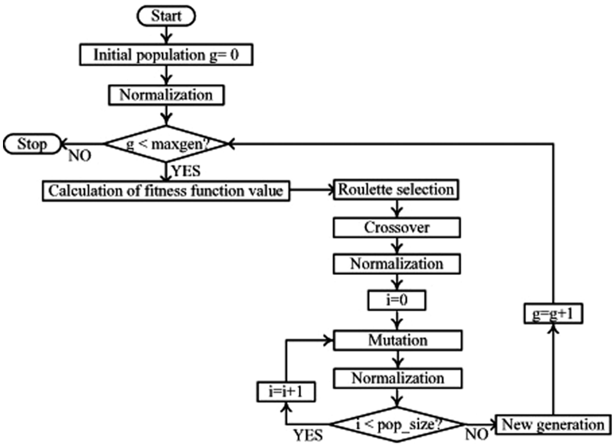
$$p'_n = (p_{n1}, \cdots,\ p_{nr},\ p_{m(r+1)}, \cdots,\ p_{m8}). \tag{14}$$

Following the crossover operation, normalisation is performed on the produced offsprings. That is, each component of the offspring must be re-calculated by equation (10).

Mutation operation is an exclusive operation that introduces new genes into the population. In the traditional GA, mutation operation is only performed on the newly produced individuals. Nevertheless, in our improved GA, the individuals that are not selected to crossover are also involved in mutation. We try to make the population as diverse as possible, so as to avoid the limitation caused by the initial population and being trapped to the local minima. Different from binary-encoding scheme, in our algorithm we can not simply exchange 0 and 1 as mutation. Instead, each individual gene is replaced by a randomly generated real number between 0 and 1 at the probability of mutation rate. For normalisation, we use the same method as in crossover operation.

The steps of the improved GA mentioned above are taken sequentially and the last three steps of fitness function calculation, selection and genetic operations are repeated iteratively. Figure 2 is the flowchart of the improved GA used in this paper.

Figure 2 The flowchart of our improved algorithm. Here, $maxgen$ is the number of generations and pop_size is the size of population



4 Experimental evaluation and discussion

4.1 Datasets

In this paper, PDB25 is used for the conditional probability calculation, where there is less than 25% sequence identity between any two proteins, and each protein has a resolution better than 2.5Å. The structures with missing atoms and broken chains are excluded. We also use the HR dataset[1] (Rajgaria et al., 2000) that comprises of 1137 subsets for experiment. HR is a large set of high resolution decoys, and the difference between native and nonnative structures is not obvious ($rmsd$ <2Å) (*rmsd* is the acronym of root mean squared deviation). Each subset contains 1 native conformation and 500 nonnative conformations of the same sequence. The length of proteins ranges from 41 to 200. The dihedral angles of all the sequences are calculated by DSSP programme and then mapped to discrete regions in the Ramachandran plot. HR dataset is randomly divided into two parts, one part used for training with 568 subsets and the other one with the rest subsets are used for testing.

All experiments were conducted on an Intel Pentium-4, 2.10 GHz PC with 2GB RAM. The improved GA was implemented in C++.

4.2 Parameter selection for the Genetic Algorithm

There are several strategy parameters in a GA, including population size, crossover rate and mutation rate. The performance of a GA is largely dependent on these parameters (Eiben et al., 1999). To find the best parameter values, we choose 100 subsets from HR randomly as the test dataset and compare the results of the GA with different parameter values. The parameters with the best performance are given in Table 1 and are used in the following experiments. We do not set the pre-defined termination criterion, instead we use the number of generations to control the iteration number of the GA. The mutation rate is 0.005, which is relatively higher than the commonly used value. Considering that mutation in this paper is operated over all individuals, the generated population is getting more and more diverse.

Table 1 The parameters used in the Genetic Algorithm

Parameter	*Value*
Population size	50
Generation size	30
Crossing rate	0.6
Mutation rate	0.005

4.3 Comparison of different strategies in Genetic Algorithm

One major drawback of GA is that its convergence can not be guaranteed. To improve the efficiency, we propose several strategies to implement the genetic operators for adapting the GA to our problem in this paper. For performance comparison, in addition to the improved GA (denoted as Algorithm I), two variations of the standard GA are also implemented, which are denoted by Algorithms II and III, respectively. In Algorithm II, mutation operation is conducted only on offspring. Before making the hypothesis that the joint conditional probability is the weighted sum of the probabilities of individual conditions, we calculate the gain of information entropy. The formula for calculation of information entropy can be seen in Dong et al. (2009). The value of the conditional entropy gets smaller comparing to the value without considering any condition, which implies each condition reduces the uncertainty and increases the probability. So all the weights(W's elements) should be greater than zero to make sure the probability is increased when it is calculated by the sum of condition probabilities under each condition. In Algorithm III, randomly generated real numbers is assigned to the vector W's elements according to the difference between the entropy without considering condition and the conditional entropy. Furthermore, an fourth algorithm was also implemented, which is denoted as Algorithm IV where the grid search is used for searching each weight (W's elements) from 0 to 1 with the step size being 0.2. We use six performance metrics to evaluate and compare these algorithms:

1 *Recognition rate* (RR for short): The ratio of targets whose native structures are the best scored structures in the decoy set.

2 *Rank*: indicates the rank of native structures in the list of decoys sorted in ascending order by the value of score function.

3 *z-score*: defined as

$$z\text{-}score = \frac{<E^{\text{decoy}}> - E^{\text{native}}}{\sqrt{<(E^{\text{decoy}})^2> - <E^{\text{decoy}}>^2}} \tag{15}$$

where E^{decoy} and E^{native} represent the scoring function values of nonnative and native structures, and the operator $< \cdot >$ is to evaluate the expectation.

4 *Correlation Coefficient* (CC for short): The *Correlation Coefficient* between the scoring function value and the C_α *root mean squared deviation* (C_αRMSD) of the decoys;

5 Δ*RMSD*: The difference of C_αRMSD between the structure with the minimum scoring function value and the best decoy structure with the minimum RMSD.

6 *n% enrichment*: The relative occurrence of the most accurate (C_αRMSD) $n\%$ models among the $n\%$ best scoring conformations.

The resulting W and the average RR of the four algorithms over the training dataset are listed in Table 2. The performance comparisons of the four algorithms over the testing dataset are shown in Table 3. From Table 3, evidently, our improved algorithm outperforms the other algorithms on the test dataset in five performance metrics, including average RR, rank, z-score, and CC. Because six elements of W obtained by grid search are zero, this method is unacceptable though its 10% enrichment value and the average recognition rate are the best. And when setting the step size to 0.1, the grid search method takes too much time to get the final result.

Table 2 The optimal weight values and average *RR* on the training dataset

Algorithm	*W*	*Average RR*
I	(0.1084, 0.1804, 0.1839, 0.0258, 0.0064, 0.2278, 0.2100, 0.057)	0.7060
II	(0.0556, 0.0867, 0.0390, 0.0145, 0.0307, 06248, 0.1219, 0.0270)	0.7148
III	(0.2265, 0.2152, 0.0607, 0.0811, 0.0059, 0.1848, 0.1798, 0.0460)	0.7007
IV	(0, 0, 0.2, 0, 0, 0.8, 0, 0)	*0.7271*

Table 3 Performance comparison on the testing dataset

Algorithm	*RR*	*Rank*	*Z-score*	*CC*	*ΔRMSD*	*10% enrichment*
I	*0.6327*	*35.1793*	*−3.8925*	*0.1140*	*2.8171*	0.6288
II	0.6204	36.5220	−3.7816	0.1014	2.7264	0.6569
III	0.6116	39.6362	−3.7279	0.1063	2.7160	0.6351
IV	0.6257	35.2636	−3.8102	0.0901	2.7071	*0.6826*

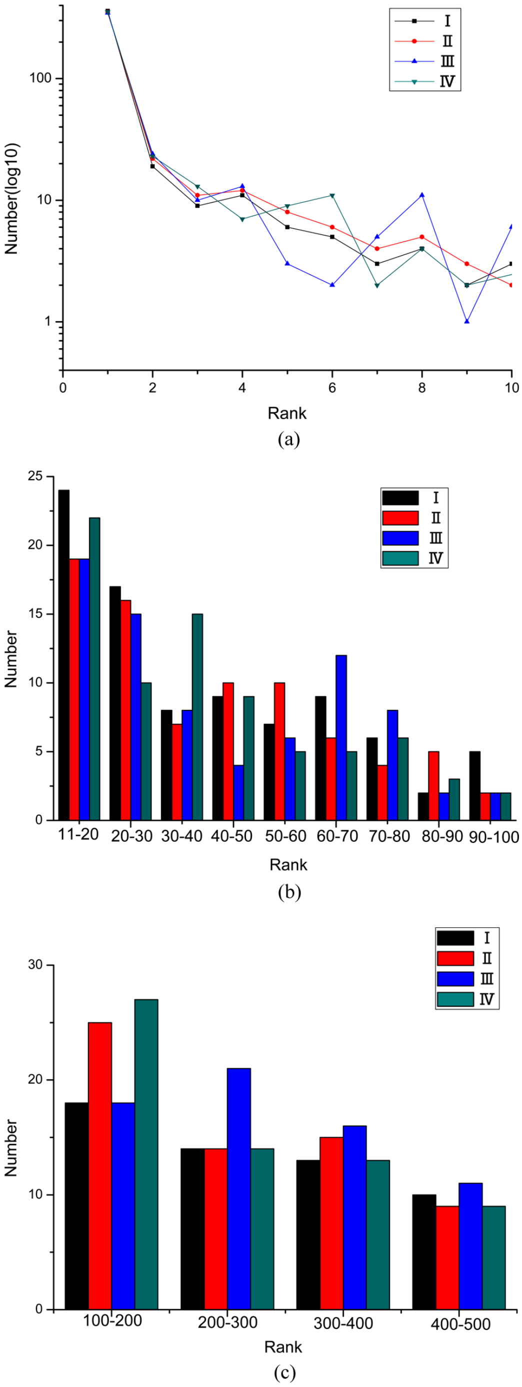
Algorithm III is designed on the assumption that the weight values may be related to the difference between the entropy and conditional entropy under one condition. The larger the difference is, the larger the value randomly generated is assigned to the corresponding weight. Therefore, the weight values are sorted and reassigned by swapping values between two elements of *W* after genetic operations every time. However, the experiment results show that it is not as good as Algorithm I and the assumption may be unreasonable. Since the recognition rate of our method is better than that of Algorithm II, we can get the conclusion that our mutation operation is good in maintaining the genetic diversity of the population and can avoid the restriction of the initial population.

Besides recognition rate, the rank of native structure can give more information about the potential function than other four metrics. The number of proteins with native structures ranked in top 10 of the four algorithms are given in Figure 3(a). We can see that the number of native conformations ranked in top 10 of our algorithm is less than that of Algorithm II, but it is less fluctuant than Algorithms III and IV. To some degree, it can explain why the 10% enrichment value of our algorithm is the worst. The histogram of the number of native conformations ranked from 11 to 500 are shown in Figure 3(b) and (c), respectively. We divide the interval $[11, 100]$ into 9 equally-sized segments and divide $[100, 500]$ into 4 equally-sized segments. From these two figures, we can see that the number of top-ranked native conformations gotten by our method is more than that by any of the other three algorithms.

4.4 Comparison with the simulated annealling algorithm

Inspired by the physical process of metal annealling, Simulated Annealling (SA) algorithm – a Monte-Carlo global minimisation technique was proposed for complex nonlinear optimisation problems. It was shown mathematically that SA converges asymptotically to the global optimal solution with probability one (Romero et al., 1996). We also used SA to search for optimal parameters and compared its prediction result with that of the improved GA. SA is inherently sequential and hence very slow for problems with large search

Figure 3 Comparison of the numbers of proteins with native structures (ranked from 1 to 500): (a) top 10; (b) histogram from the 11th to the 100th and (c) histogram from the 100th to the 500th (see online version for colours)



(a)

(b)

(c)

space (Ram et al., 1996). In this paper, we use a considerable large dataset to determine the parameters. For each sampled weight vectors, about 3×10^5 protein sequences are traversed, which is very time-consuming. At each step of SA, some neighbouring states of the current state are generated, which are probabilistically decided whether or not to replace the current state by the metropolis criterion. With the final weights obtained by the SA algorithm, the average recognition rate is around 0.65, which is worse than that of the improved GA algorithm.

5 Conclusion

This paper presented an improved algorithm with modified genetic operations for statistical potential function design and protein structure prediction. Experimental results show that the proposed algorithm can find the optimal parameters of potential function and the designed function performs well in protein structure prediction. As for the future work, we will try to develop a hybrid GA with grid search to further improve the prediction precision and efficiency.

Acknowledgements

Xin Geng and Jihong Guan are supported by the National Natural Science Foundation of China under Grant No. 60873040, Jihong Guan is also supported by the "Shuguang Scholar" Program of Shanghai Education Development Foundation under grant No. 09SG23, and the Open Research Program of Shanghai Key Lab of Intelligent Information Processing. Qiwen Dong and Shuigeng Zhou are supported by the National Basic Research Program of China under grant No. 2010CB126604.

References

Anfinsen, C.B. (1973) 'Principles that govern the folding of protein chains', *Science*, Vol. 181, No. 96, pp.223–230.

Chen, Y., Hu, J., Hirasawa, K. and Yu, S. (2007) 'GARS: an improved genetic algorithm with reserve selection for global optimization', *2007 ACM Genetic and Evolutionary Computation Conference (GECCO 2007)*, London, England, UK, pp.1173–1178.

Dong, Q., Geng, X., Zhou, S. and Guan, J. (2009) 'Empirical probability functions derived from dihedral angles for protein structure prediction', *2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering (IEEE BIBE 2009)*, Taichung, Taiwan, pp.146–152.

Dong, Q., Wang, X. and Lin, L. (2006) 'Novel knowledge-based mean force potential at the profile level', *BMC Bioinformatics*, Vol. 7, pp.252–260.

Dong, Q. and Zhou, S. (2010), 'Novel non-linear knowledge-based mean force potentials based on machine learning', *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2011*, IEEE computer Society Digital Library, IEEE Computer Society (to be published).

Eiben, *Á*.E., Hinterding, R. and Michalewicz, Z. (1999) 'Parameter control in evolutionary algorithms', *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp.124–141.

Eiben, A.E. and Smith, J.E. (2003) *Introduction to Evolutionary Computing*, Berlin, Germany, Springer.

Etaner-Uyar, A.S. and Harmanci, A.E. (2002) 'Preserving diversity through diploidy and meiosis for improved genetic algorithm performance in dynamic environments', *Lecture Notes in Computer Science*, Vol. 2457, pp.314–323.

Flory, P.J. (1969) *Statistical Mechanics of Chain Molecules*, *Wiley*, New York.

Fujitsuka, Y., Takada, S., Luthey-Schulten, Z.A. and Wolynes, P.G. (2004) 'Optimizing physical energy functions for protein folding', *PROTEINS: Structure, Function, and Bioinformatics*, Vol. 54, pp.88–103.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York.

Kabsch, W. and Sander, C. (1983) 'Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features', *Biopolymers*, Vol. 22, pp.2577–2637.

König, R. and Dandekar, T. (1999) 'Improving genetic algorithms for protein folding simulations by systematic crossover', *BioSystems*, Vol. 50, pp.17–25.

Kumar, P., Gospodaric, D. and Bauer, P. (2007) 'Improved genetic algorithm inspired by biological evolution', *Soft. Comput.*, Vol. 11, pp.923–941.

Leung, F.H.F., Lam, H.K., Ling, S.H. and Tam, P.K.S. (1997) 'Tuning of the structure and parameters of a neural network using an improved genetic algorithm', *IEEE Trans. Neural Networks*, Vol. 14, No. 1, pp.79–88.

Liwo, A., Arłukowicz, P., Czaplewski, C., Ołdziej, S., Pillardy, J. and Scheraga, H.A. (2002) 'A method for optimizing potential-energy functions by a hierarchical design of the potential-energy landscape: application to the UNRES force field', *PANS*, Vol. 99, No. 4, pp.1937–1974.

Liwo, A., Ołdziej, S., Pincus, M.R, Wawak, R.J., Rackovsky, S. and Scheraga, H.A. (1997) 'A united-residue force field for off-lattice protein-structure simulations. I. Functional forms and parameters of long-range side-chain interaction potentials from protein crystal data', *Journal of Computational Chemistry*, Vol. 18, No. 7, pp.849–873.

Lu, M, Dousis, A.D. and Ma, J. (2008) 'OPUS-PSP: an orientationdependent statistical all-atom potential derived from side-chain packing', *J. Mol. Biol.*, Vol. 376, No. 1, pp.288–301.

Melo, F. and Feytmans, E. (2008) 'Novel knowledge-based mean force potential at atomic level', *J. Mol. Biol.*, Vol. 267, pp.207–222.

Ohkubo, Y.Z. and Brooks III, C.L. (2003) 'Exploring flory isolated-pair hypothesis: statistical mechanics of helix-coil transitions in polyalanine and the C-peptide from RNase A', *PNAS*, Vol. 100, No. 24, pp.13916–13921.

Pappu, R.V., Srinivasan, R. and Rose, G.D. (2000) 'The Flory isolated-pair hypothesis is not valid for polypeptide chains: implications for protein folding', *PNAS*, Vol. 97, No. 23, pp.12565–12570.

Qiu, J. and Elber, R. (2005) 'Atomically detailed potentials to recognize native and approximate protein structures', *Proteins*, Vol. 61, No. 1, pp.44–55.

Rajgaria, R., McAllister, S.R. and Floudas, C.A. (2000) 'A novel high resolution Calpha–Calpha distance dependent force field based on a high quality decoy set', *PROTEINS: Structure, Function, and Bioinformatics*, Vol. 65, pp.726–741.

Rajgaria, R., McAllister, S.R. and Floudas, C.A. (2007) 'Distance dependent centroid to centroid force fields using high resolution decoys', *Proteins*, Vol. 70, pp.950–970.

Rajgaria, R., McAllister, S.R. and Floudas, C.A. (2008) 'Distance dependent centroid to centroid force fields using high resolution decoys', *PROTEINS: Structure, Function, and Bioinformatics*, Vol. 70, pp.950–970.

Ram, D.J., Sreenivas, T.H. and Subramaniam, K.G. (1996) 'Parallel simulated annealing algorithms', *Journal of Parallel and Distribtued Computing*, Vol. 37, pp.207–212.

Romero, R., Gallego, R.A. and Monticelli, A. (1996) 'Transmission system expansion planning by simulated annealing', *IEEE Transactions on Power Systems*, Vol. 11, No. 1, pp.364–369.

Shortle, D. (2002) 'Composites of local structure propensities: evidence for local encoding of long-range structure', *Protein Sci.*, Vol. 11, No. 1, pp.18–26.

Sippl, M.J. (2007) 'Knowledge-based potentials for proteins', *Curr. Opin. Struct. Biol.*, Vol. 5, No. 2, pp.229–235.

Su, S., Zhang, C. and Chen, J. (2007) 'An improved genetic algorithm for web services selection', *IEEE Trans. Neural Networks*, Vol. 14, No. 1, pp.284–295.

Tobi, D. and Elber, R. (2000) 'Distance-dependent, pair potential for protein folding: results from linear optimization', *PROTEINS: Structure, Function, and Genetics*, Vol. 41, pp.40–46.

Tobi, D., Shafran, G., Linial, N. and Elber, R. (2000) 'On the design and analysis of protein folding potentials', *Proteins, Structure, Function and Genetics*, Vol. 40, pp.71–85.

Wagner, M., Meller, J. and Elber, R. (2004) 'Large-scale linear programming techniques for the design of protein folding potentials', *Mathematical Programming Ser. B*, Vol. 101, No. 2, pp.301–318.

Wang, Y.F. and Chen, C.H. (2005) 'Improved genetic algorithm to solve preplanned backup path on WDM networks', *19th International Conference on Advanced Information Networking and Applications(AINA 2005)*, Vol. 2, pp.167–174.

Note

[1] http://titan.princeton.edu/HRDecoys/