Journal of Bioinformatics and Computational Biology
Vol. 3, No. 2 (2005) 385–400

ICP Imperial College Press
www.icpress.co.uk

EXPLORING PROTEIN'S OPTIMAL HP CONFIGURATIONS BY SELF-ORGANIZING MAPPING

XIANG-SUN ZHANG*, YONG WANG, ZHONG-WEI ZHAN and LING-YUN WU

Institute of Applied Mathematics
Academy of Mathematics and Systems Science
CAS, Beijing 100080, China
**zxs@amt.ac.cn*

LUONAN CHEN

Osaka Sangyo University, Nakagaito 3-1-1, Daito
Osaka 574-8530, Japan

Received 16 March 2004
1st Revision 15 July 2004
2nd Revision 3 September 2004
Accepted 9 September 2004

Self-organizing map (SOM) has been used in protein folding prediction when the HP model is employed. The existing work uses a square-like shape lattice with $l = m \times n$ points to represent the optimal compact structure of a sequence of l amino acids. In this paper, a general l'-size sequence of amino acids is self-organized in a two dimensional lattice with l $(> l')$ points. The obtained minimum configuration then has a flexible shape, in contrast to the compact structure limited in the lattice. To fulfil this extension, a new self-organizing map (SOM) technique is proposed to deal with the difficulty of the unsymmetric input and output spaces. New competition rules in the training phase are introduced and a local search method is applied to overcome the multi-mapping phenomena. Several HP benchmark examples with up to 36 amino acids are tested to verify the effectiveness of the proposed approach in this paper.

Keywords: Protein folding; artificial neural networks; self-organizing map; designability; HP lattice.

1. Introduction

A protein is a sequence of amino acid residues in its rudimentary level (or, say the primary structure), which collapses into its tertiary structure (or, native state) by some folding kinetics. Since determining the tertiary structure in a three-dimensional space experimentally is difficult and time consuming,[1] and moreover,

*Corresponding author.

the folding kinetics is a complicated problem, many groups are working on computational methods for predicting the native state of a protein from its primary structure.

In presenting a model to realize the prediction process, it is believed that the interactions between various amino acids in the sequence are the dominant factors that determine the global structure of a protein. As to the details of "interactions", there are two kinds of interactions: short-range interactions (or local interactions) produced by the neighbors in the sequence, and long-range interactions (or non-local interactions) by all the amino acids in the sequence if we neglect their biological/chemical meaning.[2] Dill and his colleagues stated convincingly in their review paper that the non-local interactions play the dominant role in the encoding of proteins in both the compactness and specific architecture.[2]

Based on such assumptions, a "simple exact model" (or simple model for short), introduced by Lau and Dill in 1989,[3] has been used to predict the main properties in protein folding when a primary sequence is given. The simple models exploit a fact that the twenty different amino acids in a protein can be divided into two classes: hydrophobic/nonpolar (H) and hydrophilic/polar (P). Thus, a primary protein sequence is represented as a sequence of H points and P points, which is called the HP model[4] or a two-letter alphabet model,[5,6] where the alphabet size means the number of different kinds of amino acids in the model. In the HP model, (1) the contacts between H points are favorable, i.e., H-H contacts decide the main structure of the protein while P points are in the subsidiary positions; and (2) by the terms of "hydrophobic" and "hydrophilic", the H amino acids like to be away from the water but the P amino acids like to be near the water in the protein's environment. Then, as pointed by many researchers, the ideal structure for a protein is to maximize the number of H-H contacts such that the H points are buried in the whole globular protein and the P points are located in the surface to be with the water molecules, which implies the compactness of the protein structure. In other words, the HP model delineates a combinatorial optimization problem: given a sequence of H and P points, put it in a three-dimensional lattice with the original ordering of the H, P points unchanged and the number of H-H contacts maximized.

The simplified protein structure problem by the HP model is an NP-complete problem, even for a two-dimensional lattice.[7] So far, many approximation methods have been adopted in this area. In particular, a self-organizing network, one of the artificial neural network models, as a powerful approximation method has been appeared in several papers related with the HP model.

Self-Organizing Map (SOM)[8,9] was used in two-dimensional simple exact model to predict protein structure by Yanikoglu and Erman.[1] In their work, a primary protein chain with $l = m \times n$ amino acids was embedded in a two dimensional $m \times n$ lattice with maximum number of H-H contacts. Their algorithm is similar to the KNIES of Altinel *et al*[10]. In their paper, three amino acid sequences (model proteins) are used to numerically confirm the efficiency of the SOM. They was found that its running time is linear with the sequence length, and that the global maximum H-H contacts configurations (or the minimum energy configurations) are

found for all the test cases. However, they also pointed out that these cases are special situations since for each case the lattice size, i.e. its length and width, is predetermined and the sequence exactly fills whole lattice points. In other words, since the size of the folding space is exactly as same as the number of amino acids, their algorithm only gives a compact rectangular structure (R) and is unapplicable to the protein sequence with arbitrary length and shape, i.e. non-rectangular structure (NR).

From the viewpoint of applications of HP model, it is not natural to limit the native state to be a square, a rectangle or some predetermined shape. Therefore, by extending the conventional HP model, this paper aims to propose a general model with the number of lattice points larger than the number of amino acids, based on the SOM technique. Since the folding space is enlarged, a given sequence with arbitrary number of amino acids folds up in a spacious lattice to its optimal structure without restrictions, according to the minimum energy principle. In other words, we can expect to obtain a flexible (e.g. non-rectangle or non-compact) and high designable structure with a different shape to the conventional approaches, because the folding structure of a given sequence is not restricted by the predetermined lattice shape but simply determined by the energy minimization. Next, in Sec. 2, we describe the new SOM algorithm to such a general problem of the HP model.

Although a HP conformation in a lattice represents an abstract protein structure, the lattice model generally cannot exactly predict the real folding structure but rather derives the essential rules due to the simplified nature of the HP model. In particular, for a 2D lattice that is efficient to study thermodynamical process of molecules, a configuration may be far from the real protein structure. To evaluate the conformation of a lattice model, the designability of a structure is introduced and is defined as the number of sequences that have this structure as their non-degenerate ground state or as their unique lowest energy state.[5,6,11] Highly designable structures are likely to be thermodynamically stable, are likely to be stable against point mutation, and have protein-like motifs.[5,6,11] In addition, these structures represent attractive targets for protein design. Therefore, to evaluate the proposed method, besides benchmark examples, comparisons of designability between the compact rectangular structures restricted in the lattices and the non-rectangular compact structures are also provided in the numerical simulation of Sec. 3. Finally, we conclude the paper by giving several general remarks in Sec. 4.

2. A New SOM Method

The idea of applying the SOM algorithm in protein structure prediction originates from the efficient performance of SOM algorithm for the Travelling Salesman Problem (TSP). In fact we can decompose the procedure of finding a global energy minimum structure of HP model into two steps. The first is to find a feasible solution, i.e. to embed the amino acid chain into the lattice. The second step is to reduce the energy based on local search scheme by incorporating heuristic information.

It should be noted that the shortest Hamilton path formulation may be more suitable than TSP formulation due to a non-loop sequence chain of the given protein with two ends. However as for the computation method by the SOM algorithm, these two problems have similar computation procedure,[10] which is the main reason why we still choose a TSP model to describe such a problem.

When a protein chain completely fills all of the lattice to be a compact structure, i.e. the amino acids has the same number as that of the lattice points, the requirement of a feasible solution is imbedding all the amino acids to the lattice points, exactly one amino acid maps to one point. If we define the distance d_{ij} between two amino acids (x_i, y_i) and (x_j, y_j) as

$$d_{ij} = |x_i - x_j| + |y_i - y_j|, \tag{1}$$

then the procedure finding the feasible solution in lattice can be abstracted to a TSP problem. However, such a TSP model is not appropriate if the protein chain is put into a bigger lattice, which usually results in a different shape from the compact rectangular structure. Since only some of the lattice points need to be visited, a generalization of the Travelling Salesman Problem to this case is called Price Collecting Travelling Salesman Problem (PCTSP).[12]

Favata and Walker[13] modified Kohonen's SOM to solve the TSP, and the simulation results show that their algorithm is capable of rapidly computing approximate solutions. The SOM algorithm in this paper for HP model is based on the simple Favata–Walker updating rule. Specifically, the network model consists of two layers. The first layer has three neurons, two of which, x_1 and x_2, represent the coordinates of the cities (lattice points), and the third one x_3 is a normalizing variable to eliminate colinearity of the cities. Correspondingly, each city "j" as the sample of the input is represented by a vector

$$\boldsymbol{Q}_j = (x_1^j, x_2^j, x_3^j)^T. \tag{2}$$

The second layer has n neurons in a circle to indicate the visiting order to the cities. The detailed Favata–Walker SOM algorithm is described in Appendix B.

2.1. *Incorporation of HP information*

How to incorporate the HP information in the training phase is one of the most important tasks to solve the HP model using the new SOM algorithm. In Yanikoglu's SOM solver, the training phase can be independently partitioned into three parts, where the first two try to embed every amino acid into one lattice point, simuntaneously keeping the sequence of the amino acids unchanged. The third part tries to make all the H points attract each other. In this phase, one can incorporate heuristic information according to the specific conformation of the protein chain. Actually, such information is helpful for the early convergence to the given lattice.

In our new SOM algorithm, we simplify the training phase into two parts. In the first part, the classical TSP update rule of the Favata–Walker algorithm is

applied, i.e. the chosen amino acid moves closer to its teacher lattice point as well as its neighbors in the sequence. This local force keeps the bond connection of the protein sequence. Suppose that $\boldsymbol{Q}_k$ is the current input vector (lattice point) and that *winner* is the chosen amino acid to be trained according to the competition rule (see Appendix B). The weight vector of the *winner* and its neighboring amino acid i on the HP sequence are updated by following equations.

$$\boldsymbol{W}_{winner}(t+1) := \frac{\boldsymbol{W}_{winner}(t) + \alpha(\boldsymbol{Q}_k - \boldsymbol{W}_{winner}(t))}{\|\boldsymbol{W}_{winner}(t) + \alpha(\boldsymbol{Q}_k - \boldsymbol{W}_{winner}(t))\|}, \tag{3}$$

and

$$\boldsymbol{W}_i(t+1) := \frac{\boldsymbol{W}_i(t) + e^{\beta}\alpha(\boldsymbol{Q}_k - \boldsymbol{W}_i(t))}{\|\boldsymbol{W}_i(t) + e^{\beta}\alpha(\boldsymbol{Q}_k - \boldsymbol{W}_i(t))\|}, \tag{4}$$

where α and β are training parameters. The range of neighborhood is determined by a parameter d, which decreases to zero when the algorithm proceeds.

In the second part, the remote force between every two amino acids is incorporated. We use three parameters λ, μ and ν to control the power of the forces between H-H pair, H-P pair and P-P pair respectively. Suppose that $\boldsymbol{W}_i$ is the weight vector of the ith amino acid. Then, we adjust the other amino acids' weight vectors according to the following rule:

$$\boldsymbol{W}_i(t+1) := \frac{\boldsymbol{W}_i(t) + \eta(t)(\boldsymbol{W}_i(t) - \boldsymbol{W}_{winner}(t))}{\|\boldsymbol{W}_i(t) + \eta(t)(\boldsymbol{W}_i(t) - \boldsymbol{W}_{winner}(t))\|} \tag{5}$$

where

$$\eta(t) = \begin{cases} \lambda(t), \text{ if amino acid } i \text{ and the } \textit{winner} \text{ are both H type,} \\ \mu(t), \text{ if amino acid } i \text{ and the } \textit{winner} \text{ belong to different types,} \\ \nu(t), \text{ if amino acid } i \text{ and the } \textit{winner} \text{ are both P type.} \end{cases} \tag{6}$$

Also, we introduce another parameter N_t to control the effect of the force between every two amino acids in the training phase. The forces between two different amino acids are applied at every N_t iterations, i.e. by adjusting N_t, we can decide the frequency of the HP forces to be applied.

When a protein folds in a larger space or lattice, clearly it is possible to get a configuration with a lower energy in contrast to the compact rectangular structures with specific lattice shapes. At the same time, the obtained results may be more biophysically meaningful because the algorithm can find more flexible structures with high designability. However, the application of SOM algorithm encounters algorithmic and computational difficulty. Since in this case the number of lattice points is larger than the number of amino acids, some points in the lattice will not be visited. But during the computational iterations of SOM, the amino acids (nodes of the SOM) are apt to cover the whole input space (lattice), an amino acid is often located between two or more lattice points. To overcome such a problem in this paper, we modify the training procedure by introducing the *learning sample set partition* and *learning sample set reduction* strategies and a *local search* procedure

which are applied at the end of the training phase. We will describe them in details in next section.

2.2. *Initialization*

Given a protein chain of n amino acids, of which the number of hydrophobic amino acids (H) is n_1. Hence, the number of hydrophilic amino acids (P) is $n - n_1$. Denote

$$J_H = \{j : j \in J = \{1, \ldots, n\} \text{ and the } j\text{th amino acid is hydrophobic}\}. \tag{7}$$

Set a lattice L with m points (cities) where $m > n$. Suppose that $m = p \times q$ where p, q are odd numbers as shown in Fig. 1. For each point i in L, denote its coordinates as (x_i, y_i), where $i = 1, \ldots, m$, $x_i \in \{0, \pm 1, \pm 2, \ldots, \pm\frac{p-1}{2}\}$, and $y_i \in \{0, \pm 1, \pm 2, \ldots, \pm\frac{q-1}{2}\}$. The given protein chain will be embedded in the lattice L with its n_1 H-amino acids in a smaller lattice L_1 as small as possible, as shown in Fig. 1. L_1 locates in the center of L with the number of points nearly equal to n_1. Each point in L is a city. The distance between two cities is defined as Eq. (1) where $d_{ij} = d_{ji}$.

Based on the coordinate matrix and normalizing formula, we have $\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_m$ as sample vectors for a SOM with 3 neurons in the input layer and n neurons in the output layer. The m learning samples show the dimension of the m lattice points whereas the n neurons in the output layer represent the n amino acids in the studied protein. At a final result, amino acid j occupies the lattice point i if the neuron j in the output layer wins when $\boldsymbol{Q}_i$ comes to the network as a sample. For the $3 \times n$ weight matrix $\boldsymbol{W}$, initial value of its column $\boldsymbol{W}_j$ for $j \in J_H$ is randomly given from the $\boldsymbol{Q}_i$s in L_1. On the other hand, the other

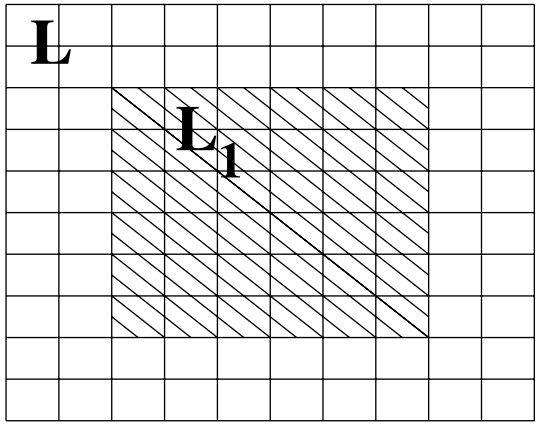

Fig. 1. A lattice model for protein folding.

initial values of $\boldsymbol{W}_j$s are randomly given from $\boldsymbol{Q}_i$s in $L \setminus L_1$, where $L \setminus L_1$ means that the remaining lattice of L by excluding L_1.

2.3. *Set partition strategy of learning sample set*

In the classical SOM algorithm for TSP, every city must be a trainer once in every iteration of training. But when the number of lattice points is greater than that of the amino acids, how to choose properly some of trainers from the large learning sample set to avoid negative effect of abundant lattice points? We propose a partition method to address this problem. At the beginning of each training iteration, we partition the lattice point set into several groups. The number of groups just equals the number of amino acids. Then every amino acid can choose a lattice point as its trainer. In the trainer choosing scheme, a deterministic annealing procedure[14] is introduced. At the initial training iterations, where the temperature is high, every lattice point in the same group has the same probability to be chosen as an input sample. But as the temperature is slowly reduced, the trainer begins focusing only on one lattice point as its teacher, and thereby the whole SOM network converges to a stable state gradually. The detailed algorithm is given in Appendix C.

2.4. *Reduction strategy of learning sample set*

This strategy is to prevent over-learning the redundant samples. At the beginning of the iterations, due to larger number of the lattice points (cities) in L, the n amino acids can take configurations freely in L. Whenever the n amino acids become to locate in the center area of L gradually, the boundary cities in L as samples will make the convergence unnecessarily unstable in the learning process. However, with this step, we gradually reduce the size of learning sample set, i.e. the size of the lattice, to alleviate such a problem. Appendix D gives a detailed description of the algorithm.

2.5. *Local search procedure*

Multi-mapping problem is a main drawback of the original SOM algorithm[15] for the HP lattice model. Multi-mapping means that there are two or more cities are mapped onto the same output neuron when the training phase is finished. It also gets worse when the learning sample set is big. In the TSP framework, the reason of this phenomenon is interpreted as that the network does not care about the local ordering of these cities in the solution.[13] But in the HP model, the conformation is invalid if any two lattice points map to the same amino acid. To overcome the multi-mapping problem, in the thesis by Wu,[15] a new linear search method is proposed when applying the SOM algorithm to the TSP problem. The result shows that this method is efficient in finding a good solution locally when the network is converged. In this paper, we further modify such a method to a new environment of the HP lattice model. The detailed algorithm is given in Appendix E.

3. Numerical Results

We first use several benchmark examples to test ability of the proposed method for protein structure prediction in Sec. 3.1, and then evaluate the designability by numerical simulation for compact rectangular and non-rectangular structures in Sec. 3.2.

3.1. *Simulation of benchmark examples*

To assess the new SOM algorithm for 2-dimensional HP model, we applied the algorithm to several test sequences, including the sequences from the paper by Yanikoglu and Erman[1] and HP benchmark[16] as well as some generated sequences. The same parameters were used in all experiments. For each sequence, the algorithm run 2000 times and there were 2000 training iterations in every run. At the end of training phase, the local search method was applied and the number of H-H pairs was recorded. The detail parameters are given in Appendix F. To solve TSP by the SOM, the feasibility of a solution is not generally guaranteed by the algorithm. In the numerical results, some runs end with bad mappings in which the protein connections are broken or mis-connected. The larger the size of problem, the more the number of bad results. In numerical simulation, we simply discard those bad results. Nevertheless, in practice, the algorithm, for example in 2000 runs, usually finds some feasible (good) solutions and several of them have lowest energy configurations.

Note that if the lattice is not big enough, i.e. $m > n$ but without proper redundancy, it will limit the self-organizing process of the protein sequence. On the other hand, if the lattice is too big, the multi-mapping problem will result in unsatisfied solutions. In numerical experiments, we find that it is appropriate to make the lattice 50–100% bigger than the length of the sequence although it depends on the constitution of HP sequence, e.g. the number of H type amino acid residues.

In Figs. 2–5, the black beads denote the H's and the white beads denote the P's. The bold line represents the covalent bond between adjacent amino acids and the light line is the edge of lattice. The results of experiment demonstrate the following four main features of the proposed algorithm:

(i) **The ability to search a complete set of optimal solutions.** The first sequence with length 20 comes from the paper of Yanikoglu and Erman,[1] as shown in Fig. 2. We embedded the HP sequence of length 20, HHHHHPHHHHHHPHHHHPHH, to the 7×7 lattice. We find not only all the corresponding native structures listed in their paper, but also a brand new conformation, as shown in Fig. 2, due to the bigger lattice with free space to generate new structure.

(ii) **The ability to handle arbitrary length protein.** In order to test the ability of new algorithm to deal with arbitrary length sequence, that is, the length l is not necessarily to be the product of two integers p and q, a new HP sequence with 17 amino acids is constructed, which results in a non-rectangle structure.

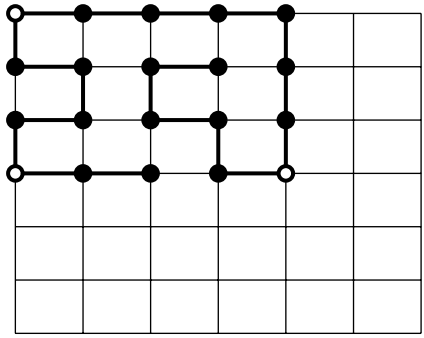

Fig. 2. A brand new minimum energy configuration of the first sequence listed in the paper by Yanikoglu and Erman.

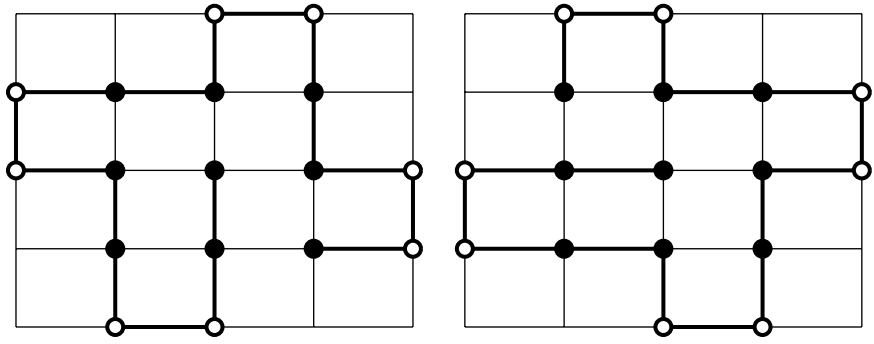

Fig. 3. The two selected minimum energy configurations of length 17, which are self-organized in 5×5 lattice.

Since this sequence cannot be put into a regular lattice, it is out of the limit of the conventional algorithms, as such one in the paper of Yanikolgu and Erman.[1] The sequence is HPPHHPPHHPPHHPPHH and it can be proved that the optimal structure is unique regardless of symmetric transformation. In the numerical test, the optimal solution is found easily, and is shown in Fig. 3.

(iii) **No limitation of protein shapes.** Some sequences of HP benchmark[16] are tested using the new algorithm. There are 14 instances for two dimensional HP model. In all of the test sequences up to 36 amino acids, including the first, the second, the third, the fourth and the ninth sequences, the algorithm was able to find the global minima of them. One of them is illustrated in Fig. 4, and is also a non-rectangle structure.

(iv) **Heredity.** For some sequences that the chain can be compactly embedded to a regular lattice, as the cases in the paper by Yanikoglu and Erman,[1] our algorithm can also be simplified to solve these problems. For example, the length of the fourth

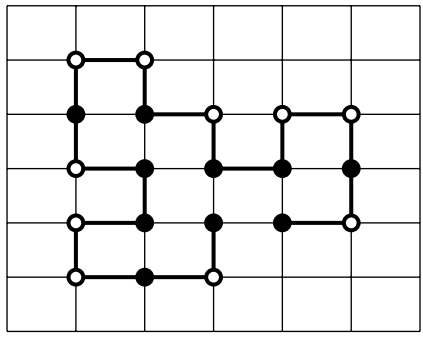

Fig. 4. The minimum energy configuration of the first sequence of HP benchmark, which is self-organized in 7×7 lattice.

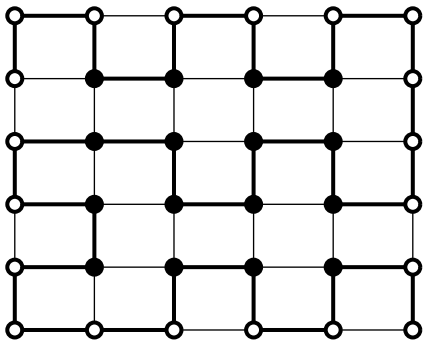

Fig. 5. The minimum energy configuration of the fourth sequence of HP benchmark. The sequence is self-organized in 6×6 lattice, and the configuration is a special case of our algorithm.

sequence in the HP benchmark is 36, which can be right-embedded into a 6×6 lattice. In the algorithm we let the learning sample set be the 6×6 lattice, as shown in Fig. 5, and the minimum energy configuration is obtained.

3.2. *Comparisons of designability*

The designability of a structure is defined as the number of sequences that have this structure as their non-degenerate ground state. A high designable structure is the unique lowest energy state of an atypically large number of sequences. It has been observed that the sequences associated with these highly designable structures are also thermodynamically more stable, fold much faster than typical sequences, possesses regular secondary structures and motifs, and in some cases, have global symmetries.[6] Therefore, in addition to the energy, the designability is also used to evaluate the conformation of the lattice model as a criterion.

Table 1. Comparisons of designable structures for sequence lengths 16–25.

Sequence Length	Number of Designable Structures		Average Designability		Maximum Designability	
	R	NR	R	NR	R	NR
16	20	436	2.5	3.4151	5	26
17	0	787	0	4.3253	0	32
18	1	1474	1	4.3066	1	48
19	0	2726	0	4.9354	0	104
20	129	5181	2.6279	4.7406	10	51
21	0	9156	0	5.6993	0	136
22	0	17881	0	5.4515	0	159
23	0	31466	0	6.3335	0	177
24	250	60836	2.172	6.2437	11	228
25	179	107157	3.3799	7.1348	10	326

Note: R denotes rectangular structure, and NR represents non-rectangular structure.

Next, instead of evaluating designability of each individual structure in the benchmark examples, we numerically show that the non-rectangular structures (NR) generally have higher designability than the rectangular structures (R) by the exhaustive enumeration. The designable structures (with positive designability) with lengths from 16 to 25 are enumerated[17] and the statistical analysis results are summarized in Table 1. Unlike most of the lattice models that consider only rectangular conformations, e.g. for a sequence with length 25 is confined to a 5×5 square, the shapes of the structures in our model have no restriction and can achieve lower energy states. As indicated in Table 1, most of the designable structures are non-rectangular, and only a small fraction (e.g. $179/107157 \simeq 0.17\%$ for structures with length 25) of the designable structures are rectangular structures, where a rectangular structure means that a sequence exactly fills a square or rectangle, e.g. the rectangular structures for the sequence with length 18 include rectangles 2×9 and 3×6 besides the trivial one-row rectangle 1×18. Moreover, the non-rectangular structures have much higher designability than the rectangular structures, thereby implying that the proposed method may find structures, which are "protein like" and can be considered as designable targets. Notice that the numbers of designable structures are zero for the sequences with sizes 17, 19 and 23 except 21 and 22 because those sequences cannot be exactly folded into non-trivial rectangular structures. On the other hand, sequences with lengths 21 and 22 can theoretically be folded into rectangles, e.g. 3×7 and 2×11, but actually have no designable rectangular structure according to the exhaustive enumeration.

4. Conclusion

A new SOM network is presented in this paper as an efficient tool for a two dimensional HP model. With the similar convergent property as that by Yanikoglu and Erman,[1] the running time of the new SOM method increases linearly with the chain length. By putting the chain into a bigger lattice to self-organize, the global minimum configuration of the amino acid chain is no longer compactly limited in a square-like or other specific-shaped lattice. In other words, the protein sequence in the proposed model may have more free space to fold into the native structure with a lower energy value, and the length of the protein chain is not limited in the number of the lattice points.

The SOM model used in this paper is different from that in the paper by Yanikoglu and Erman.[1] The three forces used by Yanikoglu and Erman are simplified to two forces. The experiments with compact lattice show that the new algorithm achieves satisfactory solution from the viewpoint of quality and convergent stability. The other contribution of this paper is that two learning strategies and a local search method are developed to overcome the difficulty brought by the bigger lattice. The adopted learning strategies, which unavoidably affect the solution of the algorithm, are the best schemes that we currently can find by intuition and experiments to our knowledge, although the algorithm for some cases suffers from convergence problem, in particular when the lattice size is sufficiently larger than sequence length, and is still time-consuming for large scale folding problems. How to improve these strategies is one of the direction for our further research.

As indicated by the recent research works, the HP model is very useful for modelling protein properties although it is simple and abstractive with several disadvantages. For example, the recent application of HP model has been applied to the investigation of aspects of ligand binding to proteins,[18] where the HP sequences having 16 monomers have been studied. Also in the work by Blackburne *et al.*, the distinct influences of function, folding, and structure on the evolution of HP model protein are studied, by investigating chains of up to 23 monomers by exhaustive enumeration of conformation and sequence space on a two-dimensional lattice, which costs four weeks of computation.[19] All those works show that it is necessary and important to develop an effective and efficient algorithm to fold the HP chain in the lattice model.

The computational experiments show that the new SOM algorithm is efficient for small scale sequences. When the input space is big, sub-optimal solutions are usually obtained, which may not be the minimum energy configurations. It is really a challenging problem to apply to large scale HP models since in that case, the conformation space grows rapidly with the increase of the chain length. This explosion can be seen from the Table II in Iwan's paper.[20] A possible method to alleviate such difficulty is to consider combination with other algorithms or carefully choose a good initial value from biological insights. On the other hand, it is also important to improve the prediction quality of the proposed method, by adjusting the lattice model or adopting other techniques of protein structure analysis.[19,23,24] In fact, a

lattice model with a large alphabet size is effective to increase the prediction accuracy, in contrast to a two-letter HP alphabet,[6] and will be further studied by the SOM algorithm in future.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China, and Center of Bioinformatics, Academy of Mathematics and Systems Science, CAS, China. The authors are grateful to the anonymous referees for comments and helping to improve the presentation of the earlier version of the paper.

Appendix A. Notation

$\boldsymbol{W}_j$,	the weight vector associated with the j-th output neuron
$\boldsymbol{Q}_i$,	the input vector of the i-th learning sample (lattice point)
$\mathcal{Q}(t)$,	the set of selected learning samples in the iteration t
L,	the set of lattice points
ξ,	a random variable with uniform distribution in $[0, 1]$
$\lfloor x \rfloor$,	the maximal integer not greater than x
$L \setminus L_1$,	the remaining lattice of L by excluding L_1

Appendix B. The Favata–Walker SOM Algorithm

Suppose that there are n normalized learning samples $\boldsymbol{Q}_i$ (the coordinates of the cities, see Eq. (2). Then, the network structure is shown in Fig. 6, and the algorithm is as follows.

Step 1. Randomly pick one $\boldsymbol{Q}_k$, and let

$$s := \arg\max \left\{ \boldsymbol{Q}_k^T \boldsymbol{W}_j, j = 1, \ldots, n \right\} \tag{B.1}$$

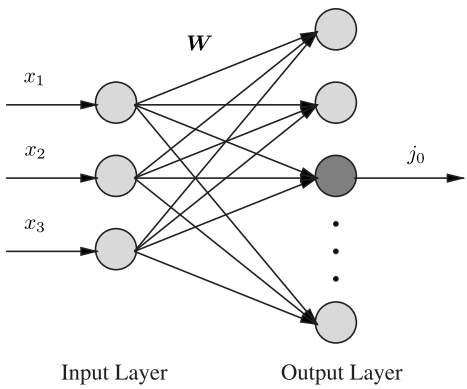

Fig. 6. Favata–Walker's SOM network to solve TSP. $\boldsymbol{W} = (w_{ij})$ is the weight matrix between the input and output layers. The initial values of w_{ij} are randomly chosen in the interval $(0, 1)$.

where $\boldsymbol{W}_j = (w_{1j}, \ldots, w_{3j})^T$, which is initially randomly chosen and corresponds to the jth output neuron, is the jth column of weight matrix $\boldsymbol{W}$ of the SOM network.

Step 2. Update the winner $\boldsymbol{W}_s$ and its neighbors $\boldsymbol{W}_{s'}$ by:

$$\boldsymbol{W}_s := \frac{\boldsymbol{W}_s + \alpha(\boldsymbol{Q}_k - \boldsymbol{W}_s)}{\|\boldsymbol{W}_s + \alpha(\boldsymbol{Q}_k - \boldsymbol{W}_s)\|} \tag{B.2}$$

$$\boldsymbol{W}_{s'} := \frac{\boldsymbol{W}_{s'} + e^{\beta}\alpha(\boldsymbol{Q}_k - \boldsymbol{W}_{s'})}{\|\boldsymbol{W}_{s'} + e^{\beta}\alpha(\boldsymbol{Q}_k - \boldsymbol{W}_{s'})\|} \tag{B.3}$$

where α and β are training parameters.

Step 3. Repeat Step 1 and Step 2 until the results of (B.1) remain unchanged.

Appendix C. Partition Strategy of Learning Sample Set

The partition strategy of the learning sample set is formally stated as follows.

Step 1. Partition the lattice L into n subsets $L_1, \ldots, L_j, \ldots, L_n$ such that

$$L = L_1 \cup L_2 \cup \cdots \cup L_n, \quad L_j \cap L_k = \emptyset, \ \ j \neq k$$

where

$$L_j = \left\{ i : \min_k \|\boldsymbol{Q}_i - \boldsymbol{W}_k\| = \|\boldsymbol{Q}_i - \boldsymbol{W}_j\| \right\}, \quad j = 1, \ldots, n \tag{C.1}$$

Denote $L_j = \{i_1^j, \ldots, i_{|L_j|}^j\}$ where the indices imply

$$\left\|\boldsymbol{Q}_{i_1^j} - \boldsymbol{W}_j\right\| \leq \|\boldsymbol{Q}_{i_2^j} - \boldsymbol{W}_j\| \leq \cdots \leq \left\|\boldsymbol{Q}_{i_{|L_j|}^j} - \boldsymbol{W}_j\right\|, \quad j = 1, \ldots, n. \tag{C.2}$$

Step 2. For $j = 1, \ldots, n$, choose

$$\boldsymbol{Q}^j := \boldsymbol{Q}_{i_{1+\lfloor \rho(|J_i|-1)\xi \rfloor}^j} \tag{C.3}$$

as the learning samples, where ξ is a random variable with uniform distribution in $[0, 1]$. ρ is a parameter decreasing from one to zero with the number of iteration, i.e. ρ corresponds to the temperature of the annealing process.

Appendix D. Reduction Strategy of Learning Sample Set

In each iteration t, $\boldsymbol{Q}^j(t)$, $j = 1, \ldots, n$, are used to train the network. Let

$$\mathcal{Q}(t) = \{\boldsymbol{Q}^1(t), \ldots, \boldsymbol{Q}^n(t)\}$$

be the sample set in the tth iteration. Usually, $\mathcal{Q}(t) \subset \{\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_m\}$. For an i_0 in the boundary (see an example of the boundary of a lattice in Fig. 7) of L, the current lattice in the computational process, if it is not in $\mathcal{Q}(t)$ for K successive iterations, we then delete i_0 from L, i.e.

$$L := L \setminus \{i_0\} \tag{D.1}$$

With this step, we gradually reduce the size of learning sample set. The integer number K will be decided in the numerical experiments.

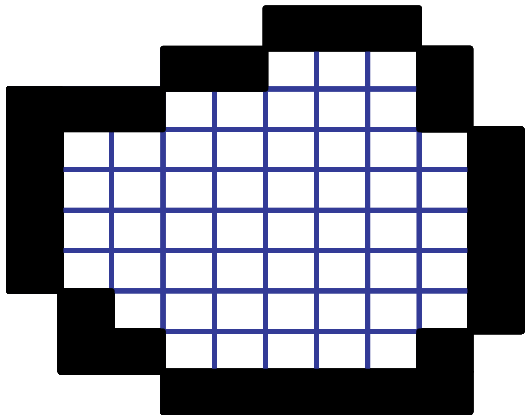

Fig. 7. An illustration for the boundary of a lattice.

Appendix E. Local Search Procedure

Suppose that the training phase is over and the weight vectors for output neurons are fixed, the local search method is applied to find a feasible solution or a better solution. The local search phase can be simply described as follows.

Step 1. According to the training result of the network, the output neurons (amino acids) are put into three subsets: C, if only one lattice point is mapped to the amino acid i; M, if more than one lattice point are mapped to the amino acid i; P, if no lattice point is mapped to amino acid i.

Step 2. Join the amino acids of set C by sequencing into a temporary chain and count the length.

Step 3. For each amino acid $i \in M$, choose one from all the lattice points that map to amino acid i which makes the length of new formed chain shortest.

Step 4. For each amino acid $i \in P$, choose one from all unvisited lattice points, which makes the length of new formed chain shortest.

Appendix F. Numerical Parameters

The initial value of the radius of updating neighborhood is $0.2n$, where n is the number of the lattice points, and then it reduces gradually to 1 in 2000 iterations. The initial value of α is 0.5. It decreases linearly to 0.1 in the first 50 iterations, then decreases 10% in each iteration until $\alpha < 10^{-5}$. β is initiated to 5 and decreases linearly to zero at the end of 2000 iterations. The parameters deciding the H-H, H-P, P-P forces are difficult to obtain due to many settings of different groups. In our tests, N_t is always taken as 4, and in common $\lambda = 0.5\alpha$, $\mu = 0.01\alpha$, $\nu = 0.03\alpha$. But when the length of the sequence gets longer, these parameters should be adjusted and set to be bigger.

References

1. Yanikoglu B, Erman B, Minimum energy configurations of the 2-dimensional HP-model of proteins by self-organizing networks, *J Comput Biol* **9**(4):613–620, 2002.
2. Dill KA, Bromberg S, Yue K, Fiebig KM, Yee DP, Thomas PD, Chan HS, Principles of protein folding — a perspective from simple exact models, *Protein Sci* **4**:561–602, 1995.
3. Lau KF, Dill KA, A lattice statistical mechanics model of the comformational and sequence spaces of proteins, *Macromolecules* **22**:3986–3997, 1989.
4. Li ZP, Zhang XS, Chen L, Unique optimal foldings of proteins on a triangular lattice, *App Bioinform*, in press, 2004.
5. Buchler NEG, Goldstein RA, Effect of Alphabet size and foldability requirements on protein structure designability, *Proteins* **34**:113–124, 1999.
6. Li H, Tang C, Wingreen N, Designability of protein structures: A lattice-model study using the Miyazawa-Jernigan matrix, *Proteins* **49**:403–412, 2002.
7. Crescenzi P, Goldman D, Papadimitriou CH, Piccolboni A, Yannakakis M, On the complexity of protein folding, *J Comput Biol* **5**(3):423–466, 1998.
8. Angéniol B, de la Croix G, Le Texier JY Self organizing feature maps and the travelling salesman problem, *Neural Networks* **1**:289–293, 1988.
9. Kohonen T, Self-organized formation of topologically correct feature maps, *Biol Cybern* **43**:59–69, 1982.
10. Altinel IK, Aras N, Oommen BJ, Fast, efficient and accurate solutions to the Hamiltonian path problem using neural approaches, *Comp Oper Res* **27**:461–494, 2000.
11. Li H, Helling R, Tang C, Wingreen N, Emergence of preferred structures in a simple model of protein folding, *Science* **273**(5275):666–669, 1996.
12. Egon Balas, The prize-collecting traveling salesman problem, *Networks* **19**: 621–636, 1989.
13. Favata F, Walker R, A study of the application of Kohonen-type neural networks to the travelling salesman problem, *Biol Cybern* **64**:463–468, 1991.
14. Chen L, Aihara K, Chaotic simulated annealing by a neural network model with transient chaos, *Neural Networks* **8**:915–930, 1995.
15. Wu L-Y, *Application of Neural Networks in Combinatorial Optimization and DNA Sequencing*. PhD thesis, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 2002.
16. William Hart and Sorin Istrail, http://www.cs.sandia.gov/tech/_reports/compbio/tortilla-hp-benchmarks.html.
17. Anders Irbäck, Carl Troein, Enumerating designing sequences in the HP model, *J Biol Phys* **28**(1):1–15, 2002.
18. Miller DW, Dill KA, Ligand binding to proteins: the binding landscape model, *Protein Science* **6**(10):2166–2179, 1997.
19. Benjamin P, Blackburne, Jonathan D, Hirst, Evolution of functional model proteins, *J Che Phys* **115**(4):1935–1942, July 2001.
20. Iwan Jensen, Enumerations of lattice animals and trees. Working Paper, 2003.
21. Ball KD, Erman B, Dill KA, The elastic net algorithm and protein structure prediction, *J Comput Chem* **23**(1):77–83, 2002.
22. Yue K, Dill KA, Forces of tertiary structural organization in globular proteins, *Proc Natl Acad Sci USA* **92**:146–150, 1995.
23. Chen L, Zhan T, Tang Y, Protein structure alignment by deterministic annealing, *Bioinformatics* **21**:51–62, 2005.
24. Zhan T, Chen L, Tang Y, Zhang XS, Aligning multiple protein structures, to appear in *Bioinform Comput Biol*.