計測自動制御学会東北支部 第199回研究集会 (2001.12.15)
資料番号 199-13

データ圧縮空間とGAを用いたタンパクの分類と機能予測

Classification and Function Estimation of Protein using Data Compression and Genetic Algorithm

菅原　研、千葉　慎二†、渡辺俊典

Ken Sugawara, Shinji Chiba†, Toshinori Watanabe

電気通信大学、†仙台電波工業高等専門学校

University of Electro-Communications, †Sendai National College of Technology

キーワード： データ圧縮 (data compression)、遺伝的アルゴリズム (genetic algorithm)、アミノ酸配列 (amino acid sequence)、分類 (classification)、機能推定 (function estimation)

連絡先： 〒182-8585　東京都調布市調布ヶ丘1-5-1電気通信大学大学院情報システム学研究科
菅原　研、Tel.:(0424)43-5603, Fax.:(0424)43-5603, E-mail: sugawara@sd.is.uec.ac.jp

1. Introduction

It is well known that the structures and functions of living things are composed of many kinds of proteins. Proteins are the general class of polymers, which are simply linear molecules composed of 20 different amino acids, but they fold themselves and develop a complicated space structure and have chemical and physical functions which originate from the structures[1]. Since the sequence of amino acids are generated according to the DNA, it is necessary to investigate the relationship between the structure/function of proteins and DNA sequences/amino acid sequences in order to understand the mechanism of life. This is because this knowledge is useful from the viewpoint of biology, medical science, protein engineering and so on.

Today, the encoding rule was found and it is easy to encode DNA sequences into amino acid sequences. But unfortunately, there is no method of predicting the structure and the function of unknown proteins accurately from DNA sequences or amino acid sequences. Although much effort has been made for accurate prediction, we cannot get their structures by calculation.

Instead of accurate calculation, there are some techniques to estimate the functions approximately using known proteins data[2]. In this case, amino acid sequences are treated as character strings and are retrieved focusing on similarity. Figure 1 shows an example of such a sequence. Each amino acid is represented by one character.

The function estimation by retrieval is classified into two types. One is called "Homology retrieval", which estimates the function of the gene based on the general similarity of the arrangements. The other is called "Motif retrieval", which estimates

the function using the common pattern in functionally resembling arrangements.

As a technique for homology retrieval, there is a classical algorithm called dynamic programming (DP)[3]. Recently, the genome database has increased drastically in size, and DP is disadvantageous in the retrieval of the whole array data base because of its calculation time. As a result, high-speed retrieval algorithms such as the BLAST method[4] or the FASTA method[5] were developed in order to save the calculation time. At present the most popular method for genome data retrieval is PSI-BLAST[6].

In this study, we propose a new method for determing a similarity discriminant of amino acid sequence information based on the concept of homology retrieval using data compression.

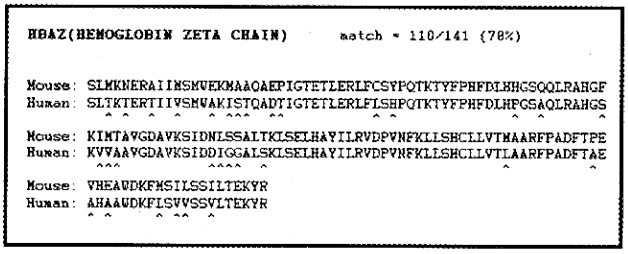

Fig. 1 An example of an amino acid sequence of a human and mouse.

2. How to classify amino acid sequences

In this section, we show the idea of how to classify sequences using the text compression method. Today there are various methods for data compression. They are divided into two classes: lossless compression and lossy compression[7]. Focusing on the lossless compression method, approaches to text compression can be divided into two classes: statistical methods and dictionary based compression methods[8]. The dictionary method was developed by Ziv and Lempel[9] and "LZ77" is one representative.

The dictionary method achieves compression by replacing groups of characters or phrases with indices in a dictionary. The dictionary is a list of characters or phrases that are expected to appear frequently. In general, the length of the indices is smaller than characters or phrases themselves, thereby we say the text is compressed.

There are two remarkable features about dictionary based compression.

- The created dictionary depends on the content of the text.
- The created dictionary is available for compressing other texts.

Now we will denote a text as T_i, a dictionary generated from T_i as D_i, and define a compression ratio $R(T_i, D_i)$ as follows.

$$R(T_i, D_i) = \frac{Compressed\ length\ of\ T_i\ by\ using\ D_i}{Original\ length\ of\ T_i}.$$

Assume we have three texts T_1, T_2 and T_x. If the content of T_x resembles T_1, the following relationship is obtained.

$$R(T_x, D_1) > R(T_x, D_2).$$

If the content of T_x resembles neither T_1 nor T_2, then

$$R(T_x, D_1) \sim R(T_x, D_2) >> 0.$$

If we create n dictionaries by compressing n proper texts, a text T_x is characterized by the dictionaries. It means that T_x can be expressed by an n-dimensional vector $\vec{K_x}$.

$$\vec{K_x} = (R(T_x, D_1), R(T_x, D_2), ..., R(T_x, D_n)).$$

The space, the axis of the space, and the vector in this space are called "Space of Data Compres-

sion(SDC)", "SDC basis", and "Feature vector", respectively[10].

Functions of unknown text data can be estimated by cluster analysis of feature vectors in SDC. Figure 2 shows a schematics of the proposed method.

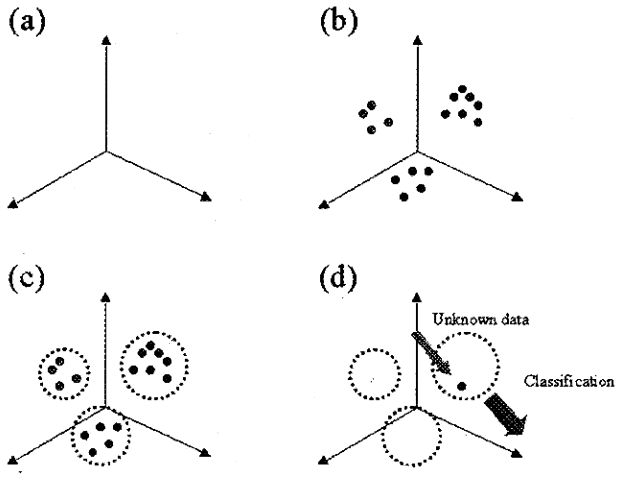

Fig. 2 Schematics of the proposed method. (a)Creation of SDC. (b) Known feature vectors are placed in SDC. (c)Clustering of the feature vectors, (d)Estimation of unknown data.

This method is simple and powerful, and we can apply it not only to protein family classification[11], but also to the recognition of audio data and image data[12, 13].

3. Dictionary Generation by Genetic Algorithm

In the previous section, we showed that the amino acid sequences are classified properly by using the text compression method. As we can see, the performance of this method depends on the quality of the dictionaries. If each dictionary has proper strings in it, the performance becomes better, but if not, a resolution becomes worse and we cannot classify proteins accurately. It is important to pick up some amino acid sequences which create good dictionaries, but we can hardly tell which protein data are appropriate for creating good dictionaries.

Fortunately, amino acid sequences are just treated as character strings in our method. It implies that artificially generated string codes can be used as ideal dictionaries.

In this section, we attempt to generate the ideal sequences for classification of amino acid sequences by introducing a genetic algorithm[15].

3.1 Algorithm for Generating Ideal Dictionaries

Now we show a concrete process for dictionary generation by GA. The aim of the operation is to generate dictionaries which classify all data into clear clusters.

< initialization >

In order to make up clusters, we introduce a new vector called "the representative vector." By considering that each feature vector belongs to the nearest representative vector, the space is segmented by them and we can treat them as clusters. It follows that the number of representative vectors corresponds to the number of clusters.

We introduce two kinds of chromosomes. One is the sequence of character strings which is used as a dictionary(denoted by $ChromoA_i$), and the other is the sequence of representative vectors (denoted by $ChromoB_j$). The length of $ChromoA_i$ is fixed, but that of $ChromoB_j$ is variable and the maximum length is given in advance. In the initial condition, $ChromoA$ are created at random. Figure 3 shows an example of $ChromoA_i$.

< repetition >

This part is composed of two phases. One is to optimize $ChromoA$ and the other is to optimize

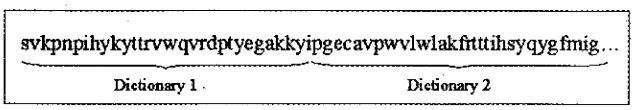

Fig. 3 An example of $ChromoA_i$.

ChromoB.

[Phase 1] Genetic operation for dictionary generation

In this phase, compression dictionaries are evolved by GA operation.

(STEP 1) Crossover

A pair of chromosomes are chosen at random and the crossover point is selected randomly. Two new chromosome are generated by exchanging the segmented part. As the crossover point is the same in each chromosome, the length is invariable.

(STEP 2) Mutation

Characters in each chromosome are forced to change stochastically. In order to keep the length of the chromosome, neither "Elimination" nor "Addition" is carried out.

(STEP 3) Evaluation

Evaluation and natural selection of the dictionary is executed in Phase 2.

[Phase 2] Genetic operation for representative vectors

In this phase, representative vectors are evolved by GA operation.

(STEP 0) Initialization

The feature vectors are created by compressing the data using the dictionaries in $ChromoA_i$, and m *ChromoB* are generated at random. Here m is 150.

(STEP 1) Crossover

Two chromosomes are chosen at random, and crossover is executed at a random point. If the length becomes less than 2 (we need two or more representative vectors for classification!), a new representative vector is generated randomly and attached to the short chromosome. If the length becomes larger than the maximum length defined in advance, some representative vectors in the chromosome are chosen randomly and deleted.

(STEP 2) Mutation

A representative vector is chosen at random, and one of the following operations is executed: "Elimination", "Addition", and "Change." "Elimination" is to eliminate the representative vector, "Addition" is to add a new random representative vector, and "Change" is to change the component values of the vector at random.

(STEP 3) Evaluation

Evaluation function is composed of three terms.

$$Ev = \alpha T_1 + \beta T_2 + \gamma T_3.$$

T_1 is an average of mixture ratio. It is necessary for each cluster to contain just one kind of feature vector. A mixture ratio of each cluster t_i is calculated as

$$t_i = \frac{n_m}{n_t},$$

where n_m is the number of feature vectors of a major family in the cluster, and n_t is the total number of feature vectors in the cluster. T_1 is calculated as

$$T_1 = \frac{1}{M} \sum t_i,$$

where M is the number of clusters.

T_2 is a distribution ratio in the margin region. In order to classify the data clearly, we need to prepare a margin at the boundary of the clusters, where

feature vectors should not be placed. Definition of this term is as follows:

$$T_2 = 1 - \frac{n_{ma}}{n_T},$$

where n_{ma} is the number of feature vectors in the margin region, and n_T is the total number of feature vectors.

T_3 is the ratio of representative vectors. If all of the data are composed of n kinds of families, it is preferable to divide the space into n regions. Definition of this term is given as follows:

$$T_3 = \frac{1}{|N_r - N_c| + 1},$$

where N_r is the number of representative vectors and N_c is the number of families.

(STEP 4) Natural selection

$ChromoB_j$ is ranked according to evaluation and selected for all j. $ChromoB_j$ with higher evaluations remains alive, and lower ones are deleted (the ranking selection).

(STEP 5) Termination

The above genetic operation is repeated 60 times. After that, the elite with highest evaluation is registered as $ChromoB_i$. These operations are executed for all $ChoromoA_i$, and they are ranked according to the evaluation value. The top 10% $ChromoA_i$ remains alive with $ChromoB_i$, and the remaining $ChromoA$ is alive without $ChromoB$.

Calculation terminates, if the evaluation value of the elite reaches a certain value given in advance, or the number of repetition reaches a certain value.

3.2 Experimental Results

In this experiment, we used the data of proteins shown in the previous section. As the data are composed of three families, we adopted a two dimensional SDC. Figure 4 is the case where the margin rate is 0.6. We can find all data are classified properly in the final generation. Figure 5 is the time evolution of the elite dictionaries generated in figure 4[16].

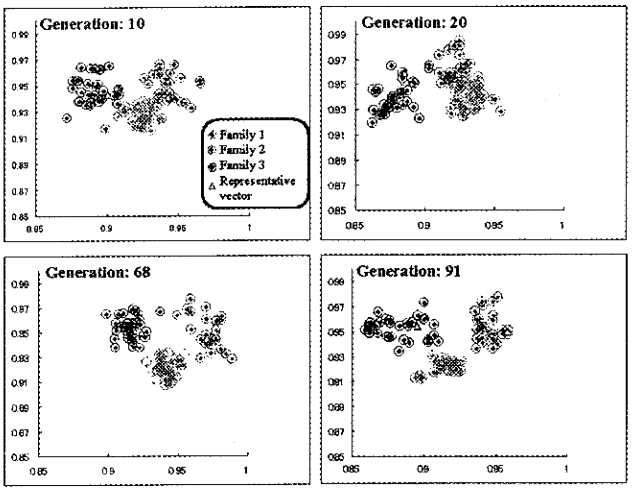

Fig. 4 Time evolution of making up the cluster. Margin rate is 0.6.

Generation	Dictionary 1	Dictionary 2
0	hfceqqwiirsanvgmlfhmmngflshkyk	pkpsdskscumnvhvyqmpqaetpntweke
10	eakwltewnmdprelpwclflastvfhyic	tyqqipawwapgadvqcenilydqfsmfsf
20	gvlfnmcasltqakslavvfhepfwfvcwv	mhgtyrtigfwaraidfhgtpaqrqaehke
40	wvlnpagqhcqnamglastihgpfwtvcwv	nhkvnategqpgaedvcnadpewwfaeheq
68	wfafpagwswqnatglafthipfwyvcwa	hhannwtegpptwhdvclrdrrgikrfdre
91	wfafpagwswqnatglafthipfwyvcwa	hhannwtegpptwhdvclrdrrgikrfdre

Fig. 5 An example of time evolution of the elite dictionaries.

4. Discussion and Conclusion

In this paper, we applied the concept of Space of Data Compression to classification of proteins. First, we created dictionaries from the known data and showed it is possible to classify proteins into proper families. Next, we generated ideal dictionaries by genetic algorithm and classified data into some families.

The advantage of the proposed method is the common patterns in functionally resembling arrange-

ments are obtained in the process of text compression. It implies that there is a possibility to find "Motif" automatically by analyzing the created dictionaries. We consider this is one of the important future works.

In this paper, we just showed the result of a small-scaled experiment and did not compare with the other methods. Now we are making an experiment on a large scale in order to show the validity of the proposed method.

参考文献

1) T.E.Creighton,"PROTEINS", W.H.FREEMAN AND COMPANY (1984).

2) A. Konagaya、"Genome and Computer"、kyoritsu (2000)(in Japanese).

3) S.B.Needleman and C.D.Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *J.Mol.Biol.*, 48, (1970), pp. 443-453.

4) S.F.Altschul, et.al, "Basic local alignment search tool", *J.Mol.Biol.*, 215(1990), pp.403-410.

5) D.J.Lipman and W.R.Pearson, "Rapid and sensitive protein similarity searches", *Science*, 227(1985), pp.1435-1441.

6) S.F.Altschul, T.L.Madden, A.A.Schaffer, J.Zhang, Z.Zhang, W.Miller, D.J.Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", *Nucleic Acids Res.*, 25(1997), pp.3389-3402.

7) M.Nelson and J.L.Gailly, "The Data Compression Book", M&T Books (1996).

8) T.C.Bell, J.G.Cleary and I.H.Witten, "TEXT COMPRESSION", Prentice-Hall (1990).

9) J.Ziv and A.Lempel, "A Universal Algorithm for Sequential Data Compression", *IEEE Trans. on Infor.Theory*, Vol.23, No.3, (1977), pp.337-343.

10) T.Watanabe, et.al, "A new pattern representation scheme using data compression", *in press.*

11) K. Sugawara and T. Watanabe, "Classification and Function Estimation of Protein using Data Compression", *Proc. of The Sixth Int. Symp. on Artificial Life and Robotics*, (2001), pp.246-249.

12) K. Kondou, N. Kato and T. Watanabe, "OSR: Online Sketch Recognition by Data Compression Technique", *J.IPSJ*,Vol.38, No.12, (1997) pp.2468-2478 (in Japanese).

13) T. Watanabe、"UPRDC: Universal Pattern Representation by Data Compression", *proc. JSPRS*, (2000) pp.253-258 (in Japanese).

14) http://www.ncbi.nlm.nih.gov/Genbank

15) L.Davis(ed.), "Handbook of Genetic Algorithms", Van Nostrand Reinhold (1990).

16) S. Chiba, K. Sugawara and T. Watanabe, "Classification and Function Estimation of Protein using Data Compression and Genetic Algorithm", *Proc. of 2001 IEEE Congress on Evolutionary Computation*, (2001) pp.839-844.