社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－BIO－5 (18)
2006／6／16

Nucleotide Encoding according to Perfect Linear Code and its Application to Multiple Alignment

Yoichi TAKENAKA † Masato SAKATA †† Hideo Matsuda †
† Graduate School of Information Science and Technology, Osaka University
†† The Institute of Medical Science, The University of Tokyo

We proposed a coding method of nucleotide subsequence using perfect linear code. To manipulate nucleotide sequences on computers, bases are expressed by binary code, such as ASCII code. Our coding method uses code words of the perfect linear code to manipulate subsequences. It regards bases as elements on Galois field $GF(4)$ and subsequences as the received words of the code. The received words are error-corrected and then manipulated on the computer. Therefore, a code word represents a subsequence and its one nucleotide differences. The proposed method will be useful for the algorithm that frequently compares the nucleotide differences among subsequences. We apply the method to local multiple alignment and report the results.

完全線形符号による部分配列エンコード法の提案と多重アライメントへの応用

竹中 要一 †， 酒田 理人 ††， 松田 秀雄 †，
† 大阪大学大学院情報科学研究科 †† 東京大学医科学研究所

計算機は全ての情報を01のビット列として扱う．核酸配列を計算機で扱う場合，各塩基はそれぞれASCIIコードや2bitに圧縮された符号により表現される．本研究では，核酸配列の符号化単位を固定長の部分文字列とし，各部分文字列を4元完全線形符号の受信語として扱う．そして受信語に誤り訂正を行うことで得られた符号を部分文字列のコード表現とする．提案法では一部分文字列とその一塩基置換である部分文字列が同一のコードで表現されることになる．そのため，部分配列の文字列比較を多用するアルゴリズムに有用なコード法であると考えている．提案手法を局所マルチプルアライメント手法に適用し，その結果を報告する．

1 Introduction

To manipulate nucleotide sequences on computers, the bases need to be expressed by binary code. One of the simplest ways is ASCII code, which is used in English text files and consequently FASTA format files. The ASCII code needs eight bit to express one base, though DNA and RNA have only four types of bases respectively. Therefore some applications used two bit code to save memory space. Such a compression code can express nucleotide sequences of four, eight and 16 bases long in a word of 8-bit, 16-bit and 32-bit MPUs respectively. It enables to reduce the execution time. To manipulate the nucleotide sequences efficiently, suffix trees[10] and finite automatons are also used. The trees were used to search rapidly for repeats in a genome[11] and to find the maximal unique matching subsequences[2) 3)]. The finite automaton were used in BLAST search[1)]. They are also kinds of codes after their data structures are encoded to binary. In the paper, we propose a code of nucleotide subsequences that utilizes the feature of perfect linear code.

The perfect linear code is a special case of the linear code in the field of coding theory[4] . It satisfies both the conditions of linear code and the equality of the Hamming bound, which means that all the received word always decoded to a code word after error-correcting. The error-correcting process of linear code calculates a syndrome that is product of a parity check matrix and each received word, which is one of the most efficient ways of error-correcting.

The linear code can use not only the words on binary but also the words on Galois field. The number of elements of Galois fields are a prime number q or a power of q. As the number of types of nucleotide in DNA or RNA is four, which is square of two, we use the Galois field with four elements $GF(4)$.

One of the features of the proposed code, one code word represents plural nucleotide sequences. They are a nucleotide sequence s and all the sequences that are one nucleotide difference from s. As a code word represents a sequence and its single nucleotide different sequences, it enables the comparison of two sequences efficient. Therefore, the proposed code is suitable for the algorithm that frequently uses the sequence comparison. Concretely, they are local multiple alignment algorithms such as Multiple Expectation maximization for Motif Elicitations (MEME)[6] and Gibbs Sampling[5] and algorithms that uses hash tables such as FASTA[7] . Another feature is its computational time.

To evaluate the effectiveness of proposed code, we apply it to one of local multiple alignment algorithms. They take plural sequences as input and output an arrangement of subsequences, highlighting their similarity. Among them, we choose MEME that executes the comparison of the fixed length subsequences frequently.

2 Perfect Linear Code encoding

2.1 Perfect Linear Code

When we wish to store, to search for, or to send information in the presence of noise efficiently and with the least error, we can apply various bounds to the efficiency. Sophisticated coding operations are developed in order to achieve efficiencies as close as possible to the bounds. The perfect linear code is the one that satisfies the equality of Hamming bound. In the following paragraphs, we define some notations to describe the perfect code[4] .

In the abstract sense, the information is generally understood to be a choice of an element from a finite set X. For implementation, we take a set K of q elements called alphabets. Each element of K is called a letter. We consider the direct product K^n, i.e., the set of all sequences of n letters. An injection ψ from X into K^n is called *encoding*. The sequence $\psi(x)$ for $x \in X$ is called a *code word*, and the image $\psi(X)$ (a collection of code words) is called a *code*. The inverse mapping φ, i.e., a mapping $\varphi : K^n \to X$ satisfying $\varphi \circ \psi(x) = x$ for all $x \in X$, is called *decoding* of ψ. The noise is represented by a mapping $\omega : K^n \to K^n$. Usually it is restricted to a certain subset Ω of the set of such mappings. A code ψ satisfying the property $\omega \circ \psi(x) \notin \psi(X)$ for all $x \in X$ and for all Ω is called *error-detecting* with respect to the noise Ω. If ψ has the decoding φ satisfying $\varphi \circ \omega \circ \psi(x) = x$ for all $x \in X$ and for all $\omega \in \Omega$, ψ is called *error-correcting* with respect to the noise Ω.

Let $d(x, y)$ be the Hamming distance between $x \in K^n$ and $y \in K^n$. We put $d_{min} = min\{d(x, y) | x, y \in X, x \neq y\}$. The error-correcting capability t of a code is the maximal integer satisfying $d_{min} \geq 2t + 1$. The Hamming bound is defined as

$$|X| \leq \frac{2^n}{\sum_{i=1}^{t} \binom{n}{I}}. \qquad (1)$$

A code satisfying the equality is called a perfect code[9] .

Let K^n be an $n-$dimensional vector space over K and Y be a $k-$dimensional linear subspace. Then the code $\psi : Y \to Y$ is called $(n, k)-$*linear code*. The perfect linear code that we use in the paper is a linear code that satisfies the equality

of the Hamming bound.

2.2 Encoding for nucleotide subsequences

We propose a nucleotide encoding by using perfect linear code. In the code, X is the nucleotide sequence with fixed length n and K is a set of elements of Galois field with four elements, where each element is corresponds to a base. A nucleotide sequence $x \in X$ is encoded to Y, a $k-$dimensional linear subspace of K^n, where there exists a perfect $(n,k)-$linear code ψ and whose error-correcting capability t is one.

In our encoding scheme, a nucleotide sequence x is regarded as a received word or $\omega \circ \psi(y)$, where $y \in Y$. And x is encoded to $y' = \varphi \circ \omega \circ \psi(y)$.

As ψ is a perfect code, y' is always equal to y and all the nucleotide sequence with length n are correctly encoded to a code word of ψ. And ψ is also a linear code, error-correcting φ can be equivalent to the calculation of the syndrome s of the received word $z = \omega \circ \psi(y)$. Let H be the parity check matrix of ψ, the syndrome is $s = zH^T = (z-x)H^T$. The error vector $= z - x$ is calculated easily from s.

We show an example of our encoding scheme. Let X be 5−mer DNA sequence, K be $\{0, 1, \alpha, \alpha^2\}$ that is a set of elements of Gaussian Field GF(4), Y be the linear subspace of K^n where the bases are (11100), (1α010) and (1$\alpha^2$001) and ψ is perfect $(5,3)-$linear code on Y.

Table 1 and 2 show the addition and multiplication of the $GF(4)$. The parity check matrix H of ψ is

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & \alpha & \alpha^2 \end{pmatrix}. \qquad (2)$$

Let four nucleotides (A,T,G,C) be correspond to (0,1,α,α^2) respectively. A sequence $GGGCA$ is firstly expressed as ($\alpha\alpha\alpha\alpha^2$0) that is regarded as the received word z of ψ. Then the syndrome s

Table 1 Addition of $GF(2^2)$

+	0	1	α	α^2
0	0	1	α	α^2
1	1	0	α^2	α
α	α	α^2	0	1
α^2	α^2	α	1	0

Table 2 Multiplication of $GF(2^2)$

*	0	1	α	α^2
0	0	0	0	0
1	0	1	α	α^2
α	0	α	α^2	1
α^2	0	α^2	1	α

of z is calculated.

$$s = xH^T = (\alpha\alpha\alpha\alpha^2 0) \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & \alpha \\ 1 & \alpha^2 \end{pmatrix} = (\alpha^2 \quad 1). \qquad (3)$$

As the syndrome s^T is same as the product of α^2 and the 4-th column of the parity check matrix H, subtract (000$\alpha^2$0) from z.

$$z - (000\alpha^2 0) = (\alpha\alpha\alpha\alpha^2 0) - (0000\alpha^2 0) = (\alpha\alpha\alpha 00) \qquad (4)$$

Finally, The nucleotide sequence $GGGCA$ is encoded to ($\alpha\alpha\alpha$00).

2.3 Short Code

The $(n,k)-$liner codes are composed of the information bits and the check bits. The information bits are arbitrary $n-k$ bits of the code and the other k bits are the check bits. The parity check matrix H can reproduce the check bits from the information bits. Therefore, the proposed code can shorten the length from n to $n-k$.

Let $y = (y_1, \cdots, y_n)$ is the code word of $n-$mer sequence. Without loss of generality, We regard $(y_{n-k+1}, \cdots, y_n)$ as the information bits and $(y_1, \cdots, y_k)$ as the check bits. We define the collection of the information bits of the proposed code as the short code of the proposed code. For example, a sequence (GGGCA) are encoded to (α00) by the short code.

2.4 Feature of the Code

One of the features of the proposed code is that one code word represents a nucleotide sequence and all of its single nucleotide different sequences. As the nucleotide sequences are firstly regarded as the received word of the perfect linear code and then encoded to the code word, the plural nucleotide sequences can be encoded to a code word. The perfect code with the error-correcting capability t means that it can correct all the errors with respect to the noise Ω where the sequence $x \in X$ and $\omega(x) \in \Omega$ are different only at less than t letters. Therefore, a code word y of the proposed code represents a nucleotide sequence that is equivalent to the y and the sequences that are t nucleotide difference from y. Fig. 1 shows a code word and the 5-mer sequences that are encoded to. In this case, a code word represents 16 sequences.

This feature of the code can increase the sequence identity of the two sequences. Let a and b be 21-mer sequences that are shown in Fig. 2. In the figure, we describe the codes with ATGC that substitute for $0, 1, \alpha, \alpha^2$. The two sequences have eight identity letters among 21, that is the identity ratio is $8/21 = 38\%$. After the sequences are encoded, the identity ratio becomes $9/21 = 42\%$. If the short code style is used, the identity ratio finally increases to 50%. The increase of the identity ratio will help to find the similar subsequences especially the case that the sequences are distant relatives. Therefore our encoding scheme will suit for the algorithms that search the similar subsequences by sequence comparison such as the identity ratio.

From the coding theory, variables n, k and t of the perfect $(n, k)-$linear codes are restricted. Tietäväinen proved that the largest number of t is one[9] , and the following equation on n and k need to hold.

$$n = \frac{q^k - 1}{q - 1}, \tag{5}$$

where q is the number of elements $|K|$. As the error-correcting capability t is one, the code word

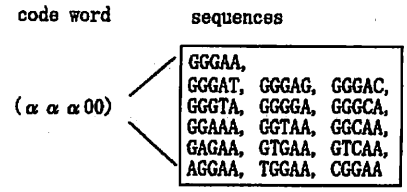

Fig. 1 A code word of our encoding method and sixteen DNA sequences that are encoded to. The sequences are GGGAA and its one nucleotide different sequences

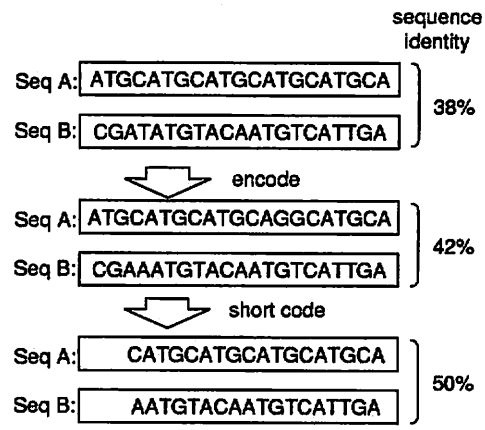

Fig. 2 Sequence identity of two sequences and their code words and the short codes. For the convenience, the codes are described with ATGC that substitute for 0,1,α and α^2

represents a nucleotide sequence and all of its single nucleotide different sequences.

3 Application to Multiple Alignment

3.1 MEME

To evaluate the proposed code, we apply to Multiple Expectation maximization for Motif Elicitations (MEME)[6] , which is one of local multiple alignment algorithms. Given are a set of sequences that are expected to have a common sequence pattern, MEME repeats two steps consecutively to find an alignment with a fixed length. The two are the expectation step and the maxi-

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & \alpha & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 \end{pmatrix}$$

Fig. 3 The parity check matrix of perfect (21,18)-linear code used in the short MEME

mization step. In advance of the two steps, an initial guess is made to find a preliminary alignment of the sequences. In the followings, the set of positions of alignment candidates for each sequence is called the site. In the first step, the column-by-column composition of the site already available is used to estimate the probability of finding the site at any position in each of the sequences. These probabilities are used in turn to provide new information as to the expected base distribution for each column in the site. The second step counts the bases for each position in the site found in the first step and substitute for the previous set. Then the first step is repeated using the new counts. The cycle is repeated until the algorithm converges on a solution.

We used the short code in our application of the proposed code to MEME. Let the length of the site in MEME be the length of the proposed code, and we use the elements of the codes $(0, 1, \alpha, \alpha^2)$ to count the frequency instead of the base (ATGC) in the two steps of MEME. We call MEME that uses the short code the short MEME. In the short MEME, the number of columns in the site is reduced from n to $n - k$.

3.2 Conditions

To compare the performance of MEME and the short MEME, we used eighteen as the length of the site in MEME and perfect $(21, 18)$−linear code for the encoding of the nucleotides into the short code. Fig. 3 shows the parity check matrix. Two types of computer generated sequences and BB30016, which is one of the benchmark sequences of BAliBASE3[8] , are used as the input. The computer generated sequences had forty 300-mer DNA sequences. Each sequence were generated as follows, i bases of a 18-mer subsequence ($CATGCATGCAGGCATGCA$) are displaced and then elongate to the 300-mer by attaching the random bases, where one type used $i = 2$, and another did $i = 6$. As the benchmark BB30016 is a set of amino acid sequences, they were reverse-translated according to the genetic code. We prepared 100 initial alignment for each input set and executed the short MEME and MEME.

3.3 Results

Fig. 4 and 5 show the results of the short MEME and MEME for the computer generated sequences with $i = 2$ and $i = 6$ respectively, and Fig. 6 shows the results for the benchmark BBS30016. The vertical axis represents the sum-of-the pair score of the alignment results of the short MEME and the vertical axis does of the MEME. A dot on the graph represents a trial. The dots above the diagonal line represent the scores of the short MEME are higher than the scores of MEME, and conversely the dots below the diagonal line represent the scores of the short MEME are lower than MEME. Numbers of dots above the diagonal line were 34, 22 and 29 for figures 4, 5 and 6 respectively. And the figures shows that the highest scores of the short MEME were lower than MEME. From these results, the short MEME exceeded MEME for one-fourth of the initial conditions, but averagely it was inferior to the MEME.

4 Discussion and Conclusion

We proposed the new coding scheme for nucleotide sequences using perfect linear code on Galois Field $GF(4)$, where each elements were correspond to a base. Nucleotide sequences were encoded by using the parity check matrix of the perfect linear code. One code word of the pro-

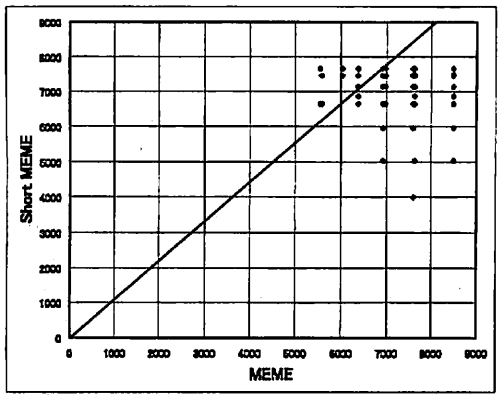

Fig. 4 The results for the computer generated data with i=2

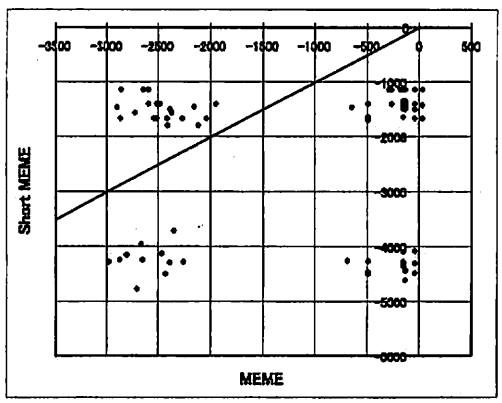

Fig. 5 The results for the computer generated data with i=6

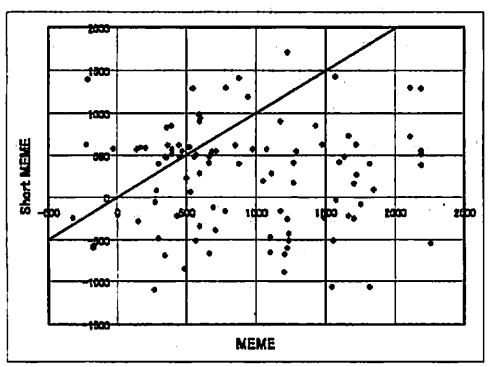

Fig. 6 The results for the benchmark BBS30016

posed code represents a sequence and all of its single nucleotide different sequences. This feature can increase the identity ratio between two sequences, especially in the short code.

To evaluate the effectiveness of the proposed code, we applied it to MEME, one of the local multiple alignment algorithms. The applied algorithm, however, could not exceed the performance of the normal MEME. To clarify the reason, we'll promote the mathematical analysis of the proposed code.

The proposed code can lessen the number of codes to represent the whole k-mer sequence. Therefore, it has potential to reduce the memory space used in the algorithms that manipulate large sequences like genome. We'd like to apply the proposed code to these algorithms in the future work.

References

1) S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–10, 1990.

2) A. L. Delcher, S. Kasif, R.D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–2376, 1999.

3) A. L. Delcher, A. Phillippy, J. Carlton, and S. L. Salzberg. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research*, 30(11):2478–2483, 2002.

4) K. Ito, editor. *Encycropedic Dictionary of Mathematics, second edition.* First MIT Press, second edition, 1996.

5) C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–14, 1993.

6) C. E. Lawrence and A. A. Reilly. An expectation maximization (em) algorithm for the

identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7:41–51, 1990.

7) W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, 85:2444–8, 1988.

8) J. D. Thompson, P. Koehl, R. Ripp, and O Poch. Balibase 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, 61:127–136, 2005.

9) A. Tietavainen. On the non-existence of perfect codes over finite fields. *SIAM J. Appl. Math.*, 24:88–96, 1973.

10) E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, pages 249–260, 1995.

11) N. Volfovsky, B. J. Haas, and S. L. Salzberg. A clustering method for repeat analysis in dna sequences. *Genome Biology*, 2:1–11, 2001.