http://dx.doi.org/10.21172/ijiet.172.01

Cyclic DNA Codes Over $M_2(F_2+uF_2)$

Baoni Dong
School of Mathematics and Statistics, Shandong University of Technology
Zibo, 255000. P. R. China

Abstract- In this paper,we consider the cyclic DNA codes over the matrix ring $M_2(F_2+uF_2)$ that is isomorphic to $F_4+uF_4+vF_4+uvF_4$,where $u^2=0$ and $v^2=0$ .We discuss the reverse constraint and reverse-complement constraint codes of the ring $M_2(F_2+uF_2)$. Finally,some cyclic DNA codes over F_4 are constructed.

Keywords – Matrix ring,Gray map,Cyclic DNA codes,Reversible cyclic codes,Reversible-complement cyclic codes

I. INTRODUCTION

Deoxyribonucleic acid (DNA) contains genetic descriptions of life structures and biological development. And DNA is a nucleic acid containing genetic instruction used as the carrier of genetic information in all living organisms. It has the information on how the biological cell runs, reproduces builds and repairs itself. The DNA strand sequence consists of four nucleotides: adenine (A) and guanine (G) are two purines, and thymine (T) and cytosine (C) are two pyrimidines. The two strands of DNA are linked with a rule that are name as Watson-Crick complement (WCC). According to (WCC) rule: every (A) is linked with a (T), and every (C) with a (G), and vice versa. That is $\overline{A}=T$, $\overline{T}=A$, $\overline{G}=C$, $\overline{C}=G$.

Recent years, since DNA computing can store more memory than silicon based computing systems. The DNA code over ring is getting more and more attention. There are some papers on cyclic DNA codes over rings, these codes caught the attention of researchers. In [1], Adleman et al. pioneered the studies on DNA computing by solving an instance of NP-complete problem over DNA molecules, explored the possibility of calculating directly with molecules. In [2], Siap et al. constructed cyclic DNA codes considering the GC content constraint over $F_2[u]/(u^2-1)$ and used the deletion distance. In [3], Yildiz et al. studied the ring with 16 elements over $F_2[u]/(u^4-1)$, first related DNA pairs with a special 16-element ring and their structure for DNA computing. In[4], Liang and Wang discussed cyclic DNA codes over the ring of four elements F_2+uF_2, and studied cyclic codes of even lengths F_2+uF_2 satisfy the reverse constraint and the reverse-complement constraint. In [5], Bayram et al. considered codes over the ring F_4+vF_4 , $v^2=v$, with 16 elements, and discussed some DNA applications. In [6], Zhu et al. studied cyclic DNA codes and also characterized the binary images of cyclic codes over the non-chain ring $F_2[u,v]/\langle u^2,v^2-v,uv-vu\rangle$ .In [7], Bennenni et al. considered the chain ring $F_2[u]/(u^6)$ with 64 elements and discussed DNA cyclic codes over this ring.

In this paper, we study cyclic DNA codes over the matrix ring $M_2(F_2+uF_2)$. The rest of this paper is organized as follows. In section 2, we do some preparatory work about ring $M_2(F_2+uF_2)$. We review some results on the matrix ring, give a Gray map from this ring to F_4. In section 3, we study the reverse constraint and reverse-complement constraint codes over $M_2(F_2+uF_2)$. A necessary and sufficient condition for cyclic DNA codes to be the reverse constraint and reverse-complement constraint codes are given. And finally, we discuss the GC weight over $M_2(F_2+uF_2)$.

II. PROPOSED ALGORITHM

A. Preliminaries-

In this paper, we denote the ring $M_2(F_2+uF_2)$ by R where $u^2=0$. R is a non-commutative ring of matrices of order 2 over the ring F_2+uF_2.

Lemma 1. [8] Let $M_2(F_2+uF_2)$ be a non-commutative ring of matrices of order 2 over the ring F_2+uF_2, and $M_2(F_2)$ be a non-commutative ring of matrices of order 2 over the finite field F_2. Then $M_2(F_2+uF_2)=M_2(F_2)+uM_2(F_2)$.

In 9, we know that codes over $M_2(F_2)$ reduce to codes over F_4+vF_4 in the following way. Let us call η an element of $M_2(F_2)$ of characteristic polynomial x^2+x+1, where

$$\eta=\begin{pmatrix}0&1\\1&1\end{pmatrix},\quad i=\begin{pmatrix}0&1\\1&0\end{pmatrix}$$

are elements of order 2 in R satisfying the relationship $i\eta=\eta^2 i$. Then $F_2[\eta]\cong F_4$ and $M_2(F_2)=F_2[\eta]+iF_2[\eta]$. Setting $v=1+i$ and identifying the subring $F_2[\eta]$ with F_4, then $M_2(F_2)=F_4+vF_4$.

From Lemma 1,we can see that

$$R=F_4+vF_4+uF_4+uvF_4$$

where $u^2=v^2=0$ and $uv=vu$.

We define a gray map from R to F_4^4 as follows

$$\varphi: R\to F_4^4$$
$$a+ub+vc+uvd\mapsto(d,c+d,b+d,a+b+c+d)$$

where $a,b,c,d\in F_4$. One can verify that the map φ is a bijection. This map can be extended to R^n. The Hamming weight w_H of $x\in F_4^n$ is defined as the number of non-zero coordinates of x.

Definition 1. A linear code C of length n over R is a left R-submodule of R^n.

The linear code C is free if C has a left R-basis. The following lemma shows that the Gray map preserves the duality.

Lemma 2. [8] If C is a linear code over R of length n, then $\varphi(C)$ is a linear code over F_4 of length $4n$.

Definition 2. Let C be a linear code over R of length n. If for any codeword $(c_0,c_1,\cdots,c_{n-1})\in C$,($(c_{n-1},c_0,c_1,\ldots,c_{n-2})$ is still a codeword in C, then C is said to be a cyclic code of length n over R.

We use $R[x]$ to represent the polynomial ring over R. Since x^n-1 is commutative, then we can make a quotient ring $R[x]/<x^n-1>$. Clearly, $R[x]/<x^n-1>$ is a left module over R .Define a map

$$\pi: R^n\to R[x]/<x^n-1>$$
$$c=(c_0,c_1,\ldots,c_{n-1})\mapsto c(x)=c_0+c_1x+\cdots+c_{n-1}x^{n-1}.$$

Clearly, π is a left R-module isomorphism. The cyclic shift of a codeword $c=(c_0,c_1,\ldots,c_{n-1})$ is $(c_{n-1},c_0,c_1,\ldots,c_{n-2})$ if and only if $x\cdot\pi(c)\in\pi(C)$, that is C is a cyclic code of length n over R if and only if $\pi(C)$ is a left ideal of $R[x]/<x^n-1>$. In this paper, we identify the cyclic code with the left ideal of the quotient ring $R[x]/<x^n-1>$.

Suppose that the length n is odd. Let $R_n = R[x]/ < x^n - 1 >$, $R = F_4$. We represent codewords by polynomials and cyclic codes are ideals in the ring R_n. The structure of cyclic codes of arbitrary length n over R was given in [10] and [17].

Lemma 3. [10] Let C be a cyclic code in $R_n = R[x]/ < x^n - 1 >$, $R = F_4 = \{0,1,w,w+1\}$ and $w^2 = 0$ mod 2.

(1) If n is odd then R_n is a principal ideal ring and

$$C = \langle g(x), up_1(x), vp_2(x), uvp_3(x) \rangle = \langle g(x) + up_1(x) + vp_2(x) + uvp_3(x) \rangle,$$

where $g(x)$, $p_1(x)$, $p_2(x)$ and $p_3(x)$ are binary polynomials with $p_3(x) \mid p_2(x) \mid p_1(x) \mid g(x) \mid (x^n - 1)$ mod 2.

(2) If n is not odd then

(i) $C = \langle g(x) + up_1(x) + vp_2(x) + uvp_3(x) \rangle$,where $g(x), p_i(x) \in F_4[x]$ with $g(x) \mid (x^n - 1)$ mod 2 and $(g(x) + up_1(x)) \mid (x^n - 1)$ in $F_4 + uF_4$,and $(g(x) + up_1(x) + vp_2(x)) \mid (x^n - 1)$ and $(g(x) + up_1(x) + vp_2(x) + uvp_3(x)) \mid (x^n - 1) \in R$.

(ii) $C = \langle g(x) + up_1(x) + vp_2(x) + uvp_3(x), uva_3(x) \rangle$, where $g(x)$, $a_3(x)$ and $p_i(x)$ are binary polynomials with $a_3(x) \mid g(x) \mid (x^n - 1)$ mod 2 with $(g(x) + up_1(x)) \mid (x^n - 1)$ in $F_4 + uF_4$, $g(x) \mid p_1(x)(\frac{x^n - 1}{g(x)})$, $a_3(x) \mid p_1(x)(\frac{x^n - 1}{g(x)})$, $a_3(x) \mid p_2(x)(\frac{x^n - 1}{g(x)})$ and $a_3(x) \mid p_3(x)(\frac{x^n - 1}{g(x)})$, $\deg a_3(x) > \deg p_3(x)$.

B. The reverse constraint and reverse-complement constraint codes–

In this section, we study the reverse constraint and reverse-complement constraint codes over R . Here, we first consider the ring

$$R = F_4[u,v]/\langle u^2, v^2, uv - vu \rangle$$
$$= \{a_1 + a_2 u + a_3 v + a_4 uv, u^2 = 0, v^2 = 0, uv = vu\}, \text{ where } a_i \in F_4, 1 \le i \le 4.$$

Let $S_{D_4} = \{A, T, C, G\}$ represent the DNA alphabet. We define a φ correspondence between the elements of R and DNA pairs presented explicitly. The elements $0, 1, w, w+1$ of F_4 are in one-to-one correspondence with the nucleotide DNA bases A, T, C, G, respectively, such that $0 \to A$, $1 \to G$, $w \to T$ and $1 + w \to C$.

Let $x = x_0 x_1 \cdots x_{n-1} \in R^n$ be an n-tuple. We define the reverse of x as $x^r = x_{n-1} x_{n-2} \cdots x_1 x_0$, where $x_i \in \{A, T, G, C\}$. And the complement of x as $x^c = \overline{x}_0 \overline{x}_1 \cdots \overline{x}_{n-2} \overline{x}_{n-1}$, where $x_i \in \{A, T, G, C\}$. By above discussion, we have $\overline{A} = T$, $\overline{T} = A$, $\overline{G} = C$, $\overline{C} = G$. So, the reverse-complement, also called the Watson-Crick complement (WCC) , is defined as $x^{rc} = \overline{x}_{n-1} \overline{x}_{n-2} \cdots \overline{x}_1 \overline{x}_0$.

Next, we study the reverse constraint and reverse-complement constraint codes over R . Let $S_{D_4} = \{A, T, C, G\}$ represent a sequence of DNA nucleotides. We define a DNA code of length n to be a set of codewords $(x_0, x_1, \ldots, x_{n-1})$, where $x_i \in \{A, T, G, C\}$. These codewords must satisfy the four constraints given below (see [11]):

(1) Let $d_H(x,y)$ denote the Hamming distance between two words. The Hamming distance constraint is that $d_H(x,y) \geq d$ for all $x,y \in C$ with $X \neq Y$, for some prescribed minimum distance d.

(2) The reverse-complement constraint is that $d_H(x^{rc},y)$ for all $x,y \in C$, where x^{rc} is the reverse-complement of x obtained by taking the reverse x^r of x and performing the symbol interchanges $A \leftrightarrow T$, $C \leftrightarrow G(WCC)$. Note that $x = y$ is include.

(3) A further constraint is used as an intermediate step in handling the reverse-complement constraint. The reverse constraint is that $d_H(x^r,y) \geq d$ for all $x,y \in C$, where x^r is the reverse of a codeword x. As for the reverse-complement constraint, $x = y$ is included.

(4) The GC-content constraint is that each codeword $x \in C$ has the same GC-content. The GC-content of a DNA word is defined to be the number of positions in which the word has coordinate C or G.

The purpose of the first three constraints is to avoid strange hybridization between different strands. The fixed GC content ensures that all codewords have similar thermodynamic characteristics, which allows parallel operations on DNA sequences.

Definition 3. A linear code C of length n over R is said to be reversible if $x^r \in C$, $\forall x \in C$, complement if $x^c \in C$, $\forall x \in C$, and reversible- complement if $x^{rc} \in C$, $\forall x \in C$.

Definition 4. A linear code C of length n over s is called a DNA code if

(1) C is a cyclic code, i. e C is an ideal of $R[x]/x^n - 1$;

(2) For any codeword $x \in C$, $x \neq x^r$ and $x^r \in C$.

For each polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} + a_k x^k$, we define the reciprocal of $f(x)$ as

$$f^*(x) = a_0^{-1} x^k f(x^{-1}) = a_k + a_{k-1}x + \cdots + a_0 x^k .$$

We note that $\deg f^*(x) \leq \deg f(x)$, and if $a_0 \neq 0$, then $\deg f^*(x) = \deg f(x)$. $f(x)$ is called self-reciprocal if and only if $f^*(x) = f(x)$. Since the cardinality of the ring R is 256, we can construct a one-to-one correspondence between the elements of R and the 256 codons over the alphabet $\{A,T,C,G\}$, which is given in Table 3, 4, 5, 6, 7.

Lemma 4. [12] Let $f(x)$, $g(x)$ be any two polynomials in R with $\deg f(x) \leq \deg g(x)$ .Then

(1) $[f(x)g(x)]^* = f^*(x)g^*(x)$;

(2) $[f(x)+g(x)]^* = f^*(x) + x^{\deg f - \deg g} g^*(x)$.

By Lemma 4, we have the following Lemma.

Lemma 5. Let $f_i(x) \in F_4[x]$, for $i = 1,2,3,4$. Suppose that $\deg(f_1(x)) = r$, $\deg(f_2(x)) = s$, $\deg(f_3(x)) = t$, $\deg(f_4(x)) = n$,where $r > \max\{s,t,n\}$. Then

$$(f_1(x) + uf_2(x) + vf_3(x) + uvf_4(x))^* = f_1^*(x) + ux^{r-s} f_2^*(x) + vx^{r-t} f_3^*(x) + uvx^{r-n} f_4^*(x)$$

Proof. Without loss of the generality we may assume that $s \geq t$. By the Lemma 4, we deduce that

$$\begin{aligned}(f_1(x) + uf_2(x) + vf_3(x) + uvf_4(x))^* &= f_1^*(x) + x^{r-s}(uf_2(x) + vf_3(x) + uvf_4(x))^* \\ &= f_1^*(x) + x^{r-s}(uf_2^*(x) + x^{s-t}(vf_3(x) + uvf_4(x))^*) \\ &= f_1^*(x) + x^{r-s}(uf_2^*(x) + x^{s-t}(vf_3^*(x) + uvx^{t-n} f_4^*(x))) \\ &= f_1^*(x) + ux^{r-s} f_2^*(x) + vx^{r-t} f_3^*(x) + uvx^{r-n} f_4^*(x)\end{aligned}$$

Lemma 6. [13] Let $C=\langle g(x)+ua(x)\rangle$ be a cyclic code of odd length n over F_4. Then C is reversible if and only if $g(x)$ and $a(x)$ are self-reciprocal.

The reverse constraint on cyclic codes over F_4+uF_4 was studied in [14]. So, we summaries these results in a form convenient for our purpose. As a direct consequence of the above proposition, the following result follows:

Lemma 7. Let $f_i(x), g_i(x)\in F_4[x]$, for $i=1,2,3,4$ .If

$$f_1(x)+uf_2(x)+vf_3(x)+uvf_4(x)=g_1(x)+ug_2(x)+vg_3(x)+uvg_4(x),$$

then $f_i(x)=g_i(x)$, for $i=1,2,3,4$.

Proof, Let $f_1(x)+uf_2(x)+vf_3(x)+uvf_4(x)=g_1(x)+ug_2(x)+vg_3(x)+uvg_4(x)$.Multiplying uv on both sides we get $uvf_1(x)=uvg_1(x)$. So $f_1(x)=g_1(x)$. Hence $uf_2(x)+vf_3(x)+uvf_4(x)=ug_2(x)+vg_3(x)+uvg_4(x)$. Then $uvf_2(x)=uvg_2(x)$, which shows that $f_2(x)=g_2(x)$. Finally, easily we have that $f_3(x)=g_3(x)$ and $f_4(x)=g_4(x)$.

We will give one of the main conclusions below.

Theorem 1. Let $C=\langle g(x)+up_1(x)+vp_2(x)+uvp_3(x), uva_3(x)\rangle$ be a cyclic code of even length n over R. And $\deg(g(x))=r$, $\deg(p_1(x))=s$, $\deg(p_2(x))=t$, $\deg(p_3(x))=n$. Then C is reversible if and only if

(1) $g(x)$ are self-reciprocal;

(2) (a) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=p_3(x)$, or;

(b) $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=p_2(x)+p_3(x)$,or;

(c) $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=p_1(x)+p_2(x)+p_3(x)$,or;

(d) $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_2(x)+p_3(x)$,or;

(e) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_1(x)+p_3(x)$,or;

(f) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_3(x)$,or;

(g) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_3(x)$,or;

(h) $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_1(x)+p_2(x)+p_3(x)$.

Proof $(\Rightarrow)$ Suppose that C is reversible. Consider C as a left $R[x]-$module. Then, by Lemma 5. it follows that

$$\begin{aligned}(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= k(x)(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C.\end{aligned}$$

for some $k(x)\in R[x]$. Assume that $k(x)=k_0(x)+uk_1(x)+vk_2(x)+uvk_3(x)$,where $k_i(x)'s$ are polynomials in $F_4[x]$. This implies

$$\begin{aligned}g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x) &= k_0(x)g(x)+u(k_0(x)p_1(x)+k_1(x)g(x))\\ &+v(k_0(x)p_2(x)+k_2(x)g(x))+uv(k_0(x)p_3(x)+k_1(x)p_2(x)+k_2(x)p_1(x)+k_3(x)g(x)).\end{aligned}$$

Now,Lemma 7 implies that $g^*(x)=k_0(x)g(x)$, $x^{r-s}p_1^*(x)=k_0(x)p_1(x)+k_1(x)g(x)$,

$x^{r-t}p_2^*(x)=k_0(x)p_2(x)+k_2(x)g(x)$, $x^{r-n}p_3^*(x)=k_0(x)p_3(x)+k_1(x)p_2(x)+k_2(x)p_1(x)+k_3(x)g(x)$

Since $g^*(x)=k_0(x)g(x)$ and $\deg g^*(x)\le \deg g(x)$,we have that $k_0(x)=1$ and so $g(x)$ is self-reciprocal.

Therefore, $x^{r-s}p_1^*(x)=k_1(x)g(x)+p_1(x)$ .Whence,comparing the degrees of the two sides of this equality,

Shows that $k_1(x)=0$ or 1,On the other hand, $x^{r-t}p_2^*(x)=k_0(x)p_2(x)+k_2(x)g(x)$,
$x^{r-n}p_3^*(x)=k_3(x)g(x)+k_2(x)p_1(x)+k_1(x)p_2(x)+k_0(x)p_3(x)$,Again,by comparing the degrees of the two sides,we imply that $k_2(x)=0$ or 1, $k_3(x)=0$ or 1,and so there are eight cases:

Case 1: If $k_1(x)=k_2(x)=k_3(x)=0$,then $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=p_3(x)$.

Case 2: If $k_1(x)=1, k_2(x)=k_3(x)=0$,then $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=p_2(x)+p_3(x)$.

Case 3: If $k_1(x)=k_2(x)=1, k_3(x)=0$,then $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=p_1(x)+p_2(x)+p_3(x)$.

Case 4: If $k_1(x)=k_3(x)=1, k_2(x)=0$,then $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_2(x)+p_3(x)$.

Case 5: If $k_1(x)=0, k_2(x)=k_3(x)=1$,then $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_1(x)+p_3(x)$.

Case 6:If $k_1(x)=k_2(x)=0, k_3(x)=1$,then $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_3(x)$.

Case 7:If $k_1(x)=k_3(x)=0, k_2(x)=1$,then $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_3(x)$.

Case 8:If $k_1(x)=k_2(x)=k_3(x)=1$,then $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $x^{r-n}p_3^*(x)=g(x)+p_1(x)+p_2(x)+p_3(x)$.

($\Leftarrow$) :Assume that (1) holds.If (2)(a) holds,then

$$\begin{aligned}(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\in C.\end{aligned}$$

In the case (2)(b) holds,we have

$$\begin{aligned}(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= g(x)+u(p_1(x)+g(x))+vp_2(x)+uv(p_3(x)+p_2(x))\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)+ug(x)+uvp_2(x)\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\ &\quad +u(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C\end{aligned}$$

When (2) (c) holds, then

$$\begin{aligned}(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= g(x)+u(p_1(x)+g(x))+v(p_2(x)+g(x))\\ &\quad +uv(p_3(x)+p_2(x)+g(x))\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\ &\quad +(u+v)g(x)+uv(p_2(x)+p_1(x))\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\ &\quad +(u+v)(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C\end{aligned}$$

If we have (2) (d), then

http://dx.doi.org/10.21172/ijiet.172.01

$$
\begin{aligned}
(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\
&= g(x)+u(p_1(x)+g(x))+vp_2(x)\\
&+uv(p_3(x)+p_2(x)+g(x))\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+ug(x)+uv(p_2(x)+g(x))\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+(u+uv)(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C
\end{aligned}
$$

When (2) (*e*) holds, then

$$
\begin{aligned}
(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+vg(x)+uv(p_1(x)+g(x))\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+vg(x)+uv(p_1(x)+g(x))\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+(v+uv)(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C
\end{aligned}
$$

If we have (2) (*f*), then

$$
\begin{aligned}
(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)+uvg(x)\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+uv(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C
\end{aligned}
$$

In the case (2) (*g*) holds, we have

$$
\begin{aligned}
(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)+vg(x)+uvp_1(x)\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+v(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C
\end{aligned}
$$

When (2) (*h*) holds, then

$$
\begin{aligned}
(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+(u+v)g(x)+uv(p_2(x)+p_1(x)+g(x))\\
&= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\
&+(u+v+uv)(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\in C
\end{aligned}
$$

Therefore, C is reversible.

Theorem 2. Let $C=\langle g(x)+up_1(x)+vp_2(x)+uvp_3(x), uva_2(x)\rangle$ be a cyclic code of even length n over R with $\deg(g(x))=r$, $\deg(p_1(x))=s$, $\deg(p_2(x))=t$, $\deg(p_3(x))=n$,where $r>\max(s,t,n)$. Also, assume that $a_2(x)\mid g(x)\mid (x^n-1)$ mod 2. Then C is reversible if and only if

(1) $g(x)$ and $a_2(x)$ are self-reciprocal;

(2) (a) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_3(x)$.

(b) $x^{r-s}p_1^*(x)=g(x)+p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_1(x)+p_2(x)+p_3(x)$.

(c) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_1(x)+p_3(x)$.

(d) $x^{r-s}p_1^*(x)=p_1(x)+g(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_2(x)+p_3(x)$.

Proof. $(\Rightarrow)$:Assume that C is reversible. Notice that there are two polynomials $t(x),s(x)\in R[x]$ such that

$$\begin{aligned}(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= g(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= t(x)(g(x)+up_1(x)+vp_2(x)+uvp_3(x))+uvs(x)a_2(x).\end{aligned}$$

Let $t(x)=t_0(x)+ut_1(x)+vt_2(x)+uvt_3(x)$,where $t_i(x)'s$ are polynomials in $F_2[x]$. Clearly, we may assume that $s(x)\in F_2[x]$ .Thus

$$\begin{aligned}g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x) &= t_0(x)g(x)+u(t_0(x)p_1(x)+t_1(x)g(x))\\ &\quad +v(t_0(x)p_2(x)+t_2(x)g(x))+uv(t_0(x)p_3(x)\\ &\quad +t_1(x)p_2(x)+t_2(x)p_1(x)+t_3(x)g(x)+s(x)a_2(x))\end{aligned}$$

Similar to the proof of Theorem 1, we deduce that $g(x)$ is self-reciprocal, $t_0(x)=1, t_1(x)=0$ or $1, t_2(x)=0$ or 1.

(i) If $t_1(x)=t_2(x)=0$,then $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$, $x^{r-n}p_3^*(x)=p_3(x)+t_3(x)g(x)+s(x)a_2(x)$.

(ii) If $t_1(x)=t_2(x)=1$,then $x^{r-s}p_1^*(x)=p_1(x)+g(x)$, $x^{r-t}p_2^*(x)=p_2(x)+g(x)$ and $x^{r-n}p_3^*(x)=p_3(x)+p_2(x)+p_1(x)+t_3(x)g(x)+s(x)a_2(x)$.

(iii) If $t_1(x)=0$, $t_2(x)=1$,then $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)+g(x)$,and $x^{r-n}p_3^*(x)=p_3(x)+p_1(x)+t_3(x)g(x)+s(x)a_2(x)$.

(iv) If $t_1(x)=1$, $t_2(x)=0$,then $x^{r-s}p_1^*(x)=p_1(x)+g(x)$, $x^{r-t}p_2^*(x)=p_2(x)$, and $x^{r-n}p_3^*(x)=p_3(x)+p_2(x)+t_3(x)g(x)+s(x)a_2(x)$.

By assumption we have $a_2(x)\mid g(x)$ mod 2. So

(a) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=p_2(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_3(x)$.

(b) $x^{r-s}p_1^*(x)=p_1(x)+g(x)$, $x^{r-t}p_2^*(x)=p_2(x)+g(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_1(x)+p_2(x)+p_3(x)$

(c) $x^{r-s}p_1^*(x)=p_1(x)$, $x^{r-t}p_2^*(x)=g(x)+p_2(x)$ and $a_2(x)\mid x^{r-n}p_3^*(x)+p_1(x)+p_3(x)$.

(d) $x^{r-s}p_1^*(x)=p_1(x)+g(x)$, $x^{r-t}p_2^*(x)=p_2(x)$,and $a_2(x)\mid x^{r-n}p_3^*(x)+p_2(x)+p_3(x)$.

Moreover, $uva^*(x)\in C$. Then, there exist polynomials $\lambda_0(x),\lambda_1(x),\lambda_2(x),\lambda_3(x)\in R[x]$ and $\mu(x)\in F_4[x]$ such that

$$\begin{aligned}uva^*(x) &= \lambda_0(x)g(x)+u(\lambda_0(x)p_1(x)+\lambda_1(x)g(x))+v(\lambda_0(x)p_2(x)+\lambda_1(x)g(x))\\ &\quad +vu(\lambda_0(x)p_3(x)+\lambda_1(x)p_2(x)+\lambda_2(x)p_1(x)+\lambda_3(x)g(x)+\mu(x)a_2(x)).\end{aligned}$$

Therefore, $\lambda_0(x)=\lambda_1(x)=\lambda_2(x)=0$, $a^*(x)=\lambda_3(x)g(x)+\mu(x)a_2(x)$ .Since $a_2(x)\mid g(x)$,then $a_2(x)\mid a^*(x)$ and so $a_2(x)=a^*(x)$, i.e. $a_2(x)$ is self-reciprocal.

$(\Leftarrow)$:We investigate only one case. Suppose that (1) and part (*b*) of (2) hold. Then

$$\begin{aligned}(g(x)+up_1(x)+vp_2(x)+uvp_3(x))^* &= g^*(x)+ux^{r-s}p_1^*(x)+vx^{r-t}p_2^*(x)+uvx^{r-n}p_3^*(x)\\ &= g(x)+u(p_1(x)+g(x))+v(p_2(x)+g(x))+uv(p_3(x)\\ &\quad +p_2(x)+p_1(x)+\lambda(x)a_2(x))\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\ &\quad +(u+v)g(x)+uv(p_2(x)+p_1(x)+\lambda(x)a_2(x))\\ &= g(x)+up_1(x)+vp_2(x)+uvp_3(x)\\ &\quad +u(g(x)+up_1(x)+vp_2(x)+uvp_3(x))\\ &\quad +v(g(x)+up_1(x)+vp_2(x)+uvp_3(x))+uv\lambda(x)a_2(x)\in C\end{aligned}$$

where $\lambda(x)\in F_4[x]$. Consequently, C is reversible.

Cyclic codes of arbitrary length n satisfying the reverse-complement are examined. We give a important lemma firstly which can be easily proved from Table 3.

Lemma 8. For any $a,b\in R$, then we have

(i) $a+\overline{a}=uvw$;

(ii) $\overline{a+b}=\overline{a}+\overline{b}+uvw$;

Proof. The proof process follows from trying all elements in R .

Lemma 8. For any $a,b\in R$, then we have $uvw+\overline{uvwa}=uvwa$.

Proof. If $a=0$,then we have

$$uvw+\overline{uvwa}=uvwa+\overline{0}=uvw+uvw=0=uvwa\ .$$

If $a=1$,then we obtained

$$uvw+\overline{uvwa}=uvw+\overline{uvw}=uvw+0=uvwa\ .$$

Similarly, one can also check the cases of $a=w$ and $a=w+1$ respectively. Here, we omit them for the interesting of space.

Next, we will give one of our main conclusions below.

Theorem 3. Let C be a cyclic code of length n over R . Then C is reversible-complement if and only if C is reversible and $uvwH(x)\in C$, where $H(x)=\dfrac{x^n-1}{x-1}$.

Proof. Let C be reversible-complement. Since the zero codeword is in C ,so $0+0x+0x^2+\cdots+0x^{n-1}\in C$,we have that

$$\overline{0+0x+0x^2+\cdots+0x^{n-1}}=uvw(1+x+x^2+\cdots+x^{n-1})=uvwH(x)\in C$$

Next, let $f(x)\in R[x]$, $f(x)=a_0+a_1x+a_2x^2+\cdots+a_{s-1}x^{s-1}+x^s$ .Then

$$\begin{aligned}f(x)^{rc} &= \overline{0}+\overline{0}x+\overline{0}x^2+\cdots+\overline{0}x^{n-s-2}+\overline{1}x^s+\overline{a_{s-1}}x^{n-s}+\cdots+\overline{a_1}x^{n-2}+\overline{a_0}x^{n-1}\\ &= uvw+uvwx+\cdots+uvwx^{n-s-2}+(1+uvw)x^{n-s-1}+\overline{a_{s-1}}x^{n-s}+\cdots+\overline{a_1}x^{n-2}+\overline{a_0}x^{n-1}\\ &= uvw+uvwx+\cdots+uvwx^{n-s-2}+(1+uvw)x^{n-s-1}\\ &\quad +(a_{s-1}+uvw)x^{n-s}+\cdots+(a_1+uvw)x^{n-2}+(a_0+uvw)x^{n-1}\end{aligned}$$

Hence,

$$f(x)^{rc} + uvwH(x) = x^{n-s-1} + a_{s-1}x^{n-s} + \cdots + a_1x^{n-2} + a_0x^{n-1} = f(x)^r$$

So, we have C is reversible-complement if and only if C is reversible and $uvwH(x) \in C$.

Theorem 4. Let $C = \langle g(x) + up_1(x) + vp_2(x) + uvp_3(x) \rangle$ be a cyclic code of even length n over R .And $\deg g(x) = r$, $\deg p_1(x) = s$, $\deg p_2(x) = t$, $\deg p_3(x) = n$ .Then C is reversible-complement if and only if

(1) $uvwH(x) \in C$;

(2) $g(x)$ are self-reciprocal;

(3) (a) $x^{r-s}p_1^*(x) = p_1(x)$, $x^{r-t}p_2^*(x) = p_2(x)$ and $x^{r-n}p_3^*(x) = p_3(x)$, or;

(b) $x^{r-s}p_1^*(x) = g(x) + p_1(x)$, $x^{r-t}p_2^*(x) = p_2(x)$ and $x^{r-n}p_3^*(x) = p_2(x) + p_3(x)$,or;

(c) $x^{r-s}p_1^*(x) = g(x) + p_1(x)$, $x^{r-t}p_2^*(x) = g(x) + p_2(x)$ and $x^{r-n}p_3^*(x) = p_1(x) + p_2(x) + p_3(x)$,or;

(d) $x^{r-s}p_1^*(x) = g(x) + p_1(x)$, $x^{r-t}p_2^*(x) = p_2(x)$ and $x^{r-n}p_3^*(x) = g(x) + p_2(x) + p_3(x)$,or;

(e) $x^{r-s}p_1^*(x) = p_1(x)$, $x^{r-t}p_2^*(x) = g(x) + p_2(x)$ and $x^{r-n}p_3^*(x) = g(x) + p_1(x) + p_3(x)$,or;

(f) $x^{r-s}p_1^*(x) = p_1(x)$, $x^{r-t}p_2^*(x) = p_2(x)$ and $x^{r-n}p_3^*(x) = g(x) + p_3(x)$,or;

(g) $x^{r-s}p_1^*(x) = p_1(x)$, $x^{r-t}p_2^*(x) = g(x) + p_2(x)$ and $x^{r-n}p_3^*(x) = g(x) + p_3(x)$,or;

(h) $x^{r-s}p_1^*(x) = g(x) + p_1(x)$, $x^{r-t}p_2^*(x) = g(x) + p_2(x)$ and $x^{r-n}p_3^*(x) = g(x) + p_1(x) + p_2(x) + p_3(x)$.

Proof. The result follows from Theorem 1 and Theorem 4.

Theorem 5. Let $C = \langle g(x) + up_1(x) + vp_2(x) + uvp_3(x), uva_2(x) \rangle$ be a cyclic code of even length n over R with $\deg(g(x)) = r$, $\deg(p_1(x)) = s$, $\deg(p_2(x)) = t$, $\deg(p_3(x)) = n$,where $r > \max(s,t,n)$. Also, assume that $a_2(x) \mid g(x) \mid (x^n - 1)$ mod 2. Then C is reversible-complement if and only if

(1) $uvwH(x) \in C$;

(2) $g(x)$ and $a_2(x)$ are self-reciprocal;

(3) (a) $x^{r-s}p_1^*(x) = p_1(x)$, $x^{r-t}p_2^*(x) = p_2(x)$ and $a_2(x) \mid x^{r-n}p_3^*(x) + p_3(x)$.

(b) $x^{r-s}p_1^*(x) = g(x) + p_1(x)$, $x^{r-t}p_2^*(x) = g(x) + p_2(x)$ and $a_2(x) \mid x^{r-n}p_3^*(x) + p_1(x) + p_2(x) + p_3(x)$.

(c) $x^{r-s}p_1^*(x) = p_1(x)$, $x^{r-t}p_2^*(x) = g(x) + p_2(x)$ and $a_2(x) \mid x^{r-n}p_3^*(x) + p_1(x) + p_3(x)$.

(d) $x^{r-s}p_1^*(x) = p_1(x) + g(x)$, $x^{r-t}p_2^*(x) = p_2(x)$ and $a_2(x) \mid x^{r-n}p_3^*(x) + p_2(x) + p_3(x)$.

Proof. The result follows from Theorem 2 and Theorem 4.

Let C be a cyclic code of arbitrary length n over R . Then we can get the conditions that C is reversible or reversible-complement easily by mental Lemmas 3, 4, 5, 6 and Theorems 1, 2, 3, 4, 5.

C. The GC weight-

It is well known that DNA code with the same GC weight (content) in every codeword ensures that these codewords have similar thermodynamic properties (i.e. unchaining temperature and hybridization energy). In this section, we will study the GC weight over R by the Gray image.

In order to study the GC weight over R , we give the following lemma first which can be received from [16] easily.

Lemma 10. Let $C=\langle g_1(x)+up_1(x)+vp_2(x)+uvp_3(x),uvg_2(x)\rangle$ be a cyclic code over R .Then $C_{uv}=\langle uvg(x)\rangle$ is the subcode of C containing all codewords of C a multiple of uv.

Proof. Let C_{uv} be the subcode of C containing all codewords with nonzero elements uv. Then it is obvious that the code $\langle uvg_2(x)\rangle$ is a subset of C_{uv}. Now we have to show that $C_{uv}\subseteq\langle uvg_2(x)\rangle$ .Let $c\in C_{uv}$,then $c=k_0(x)(g_1(x)+up_1(x)+vp_2(x)+uvp_3(x))+uvg_2(x)k_1(x)=uvg(x)$,where $k_0(x),k_1(x),g(x)\in R[x]$. Since $g_2(x)\mid g_1(x)$, $g_2(x)\mid p_i(x)$, $0\le i\le 3$,so $g_2(x)\mid g(x)$,then $c\in uvg_2(x)$,so $C_{uv}\supseteq\langle uvg_2(x)\rangle$. Therefore, $C_{uv}=\langle uvg_2(x)\rangle$.

Theorem 6. The GC weight of $C=\langle g_1(x)+up_1(x)+vp_2(x)+uvp_3(x),uvg_2(x)\rangle$ is given by the Hamming weight enumerator of the binary cyclic code $\langle g_2(x)\rangle$.

Proof.. The GC content is obtained by multiplying the codewords of C by uv, and by Lemma 10, we have $C_{uv}=\langle uvg_2(x)\rangle$. Hence, the GC content is given by the Hamming weight of the binary code generated by $g_2(x)$.

III. EXPERIMENT AND RESULT

In the following, we give the example to illustrate the main results in this paper. In the example, some cyclic DNA codes over F_4 are constructed by cyclic DNA codes over R and the Gray map.

Example 1. Consider the factorization $x^4-1=(x+1)^4=g^4$ over F_4 .Let $C=\langle g(x)+up_1(x)+vp_2(x)+uvp_3(x)\rangle$, where $g(x)=g^3$, $p_1=x^2+x$ and $p_2=x^2+x$, $p_3=x^3+1$. It is easy to find that $g(x)=x^3+x^2+x+1$ is self-reciprocal. Moreover, $x^i p_1^*(x)=p_1(x)$, $x^j p_2^*(x)=p_2(x)$ and $x^k p_3^*(x)=p_3(x)$,where $i=\deg g(x)-\deg p_1(x)$, $j=\deg g(x)-\deg p_2(x)$ and $k=\deg g(x)-\deg p_3(x)$ .The C is a cyclic DNA code. Here, the codewords of this code which are given in Table 1.

Table 1: DNA codes of length 4 obtained from C

AAAAAAAAAAAAAAAA	*ACCCACCCCCCACCCA*	*ATTTATTTTTTATTA*
AGGGAGGGGGGAGGGA	*ACCCCCCACCCAACCC*	*AAAACAACAAAACAAC*
AGGGCGGTGGGATGGC	*ATTTCTTGTTTAGTTC*	*ATTTTTTATTTAATTT*
AGGGTGGCGGGACGGT	*AAAATAATAAAATAAT*	*ACCCTCCGCCCAGCCT*
AGGGGGGAGGGAAGGG	*ATTTGTTCTTTACTTG*	*ACCCGCCT CCCATCCG*
AAAAGAAGAAAAGAAG	*CCCACCCAACCCACCC*	*CAACCAACCAACCAAC*
CGGTCGGTTGGCT GGC	*CT TGCTTGGT TCGTTC*	*CAACAAAACAACAAAA*
CCCAACCCACCCCCCA	*CTTGATTTGTTCTTTA*	*CGGTAGGGTGGCGGGA*
CGGTGGGATGGCAGGG	*CTTGGTTCGTTCCTTG*	*CCCAGCCTACCCTCCG*
CAACGAAGCAACGAAG	*CTTGTTTAGTTCATTT*	*CGGTTGGCTGGCCGGT*
CAACTAATCAACTAAT	*CCCAT CCGACCCGCCT*	*TTTATTTAATTTATTT*
TGGCTGGCCGGTCGGT	*TAATTAATTAATTAAT*	*TCCGTCCGGCCTGCCT*
TGGCGGGACGGTAGGG	*TTTAGTTCATTTCTTG*	*TCCGGCCT GCCTTCCG*
TAATGAAGTAATGAAG	*TAATAAAATAATAAAA*	*TCCGACCCGCCTCCCA*
TTTAATTTATTTTTTA	*TGGCAGGGCGGTGGGA*	*TCCGCCCAGCCTACCC*
TAATCAACTAATCAAC	*TGGCCGGTCGGTTGGC*	*TTTACTTGATTTGTTC*
GGGAGGGAAGGGAGGG	*GT TCGTTCCTTGCTTG*	*GCCTGCCTTCCGTCCG*
GAAGGAAGGAAGGAAG	*GTTCTTTACTTGATTT*	*GGGATGGCAGGGCGGT*
GAAGTAATGAAGTAAT	*GCCTTCCGTCCGGCCT*	*GCCTCCCATCCGACCC*

GAAGCAACGAAGCAAC	*GGGACGGTAGGGTGGC*	*GTTCCTTGCT TGGTTC*
GAAGAAAAGAAGAAAA	*GCCTACCCTCCGCCCA*	*GTTCATTTCTTGTTTA*
GGGAAGGGAGGGGGGA		

Example 2. Consider the factorization $x^8-1=(x+1)^8=g^8$ over F_4. Let $C=\langle g_1(x)+up_1(x)+vp_2(x)+uvp_3(x), uvg_2(x)\rangle$ where $g_1(x)=g^6$, $p_1=x^5+x$ and $p_2=x^4+x^2$, $p_3=x^3$, $g_2=g^3$. It is easy to find that $g(x)=x^3+x^2+x+1$ is self-reciprocal. Moreover, $x^i p_1^*(x)=p_1(x)$, $x^j p_2^*(x)=p_2(x)$ and $x^k p_3^*(x)=p_3(x)$, where $i=\deg g(x)-\deg p_1(x)$, $j=\deg g(x)-\deg p_2(x)$ and $k=\deg g(x)-\deg p_3(x)$. The C is a cyclic DNA code.Here, the codewords of this code with the same GC content equal to 16, which are given in Table 2. Typically the GC content is required to be in the range of 30% - 50% of the length of the code.

Table 2: DNA codes of length 8 obtained from C

GGAAAAGGGGAAGGAAAGGAGAAGAGGAAGGA	*GGCCAAGTGGAATTCACTGATCCGAGGAATGA*
GGAT AAGGCCT ACCAAT CCAGAAGACGAAGCT	*GGGGAAGAGGAAAAGAGAGAAGGGAGGAAAGA*
GGCT AAGT AAGACCCAT CAAT CCGAAGAAT AG	*GGCGAAGTCCTAAACAGACATCCGACGAATCT*
GGT AAAGCCCT AGGT AAGCACT T GACGAACCT	*GGGCAAGACCTATTGACTCAAGGGACGAAACT*
GGAACCT GT T AAT T CAAGGAGCAGAGT CAGGC	*GGCCCCTTTTAAGGAACTGATACGAGTCATGC*
GGT T CCT CT T AAAAGAT CGACGT GAGT CACGC	*GGGGCCTATTAACCTAGAGAAT GGAGTCAAGC*
GGAT CCT GAAT AAACAT CCAGCAGACT CAGCG	*GGCGCCT T AATACCAAGACAT ACGACTCATCG*
GGT ACCT CAAT AT T GAAGCACGT GACT CACCG	*GGGCCCTAAATAGGTACTCAATGGACTCAACG*
GGAAT T CGCCAACCT AAGGAGT AGAGCT AGGT	*GGCCT TCTCCAAAAGACTGAT GCGAGCTATGT*
GGT T T T CCCCAAGGAAT CGACAT GAGCT ACGT	*GGGGTTCACCAAT T CAGAGAACGGAGCTAAGT*
GGAT T T CGGGT AGGT AT CCAGT AGACCT AGCA	*GGCGT TCT GGTATTGAGACATGCGACCTATCA*
GGT AT T CCGGT ACCAAAGCACAT GACCT ACCA	*GGGCT TCAGGTAAACACTCAACGGACCTAACA*
GGAAGGAGAAAAAAGAAGGAGGAGAGAGAGGG	*GGCCGGATAAAACCTACTGAT TCGAGAGATGG*
GGT T GGACAAAAT T CAT CGACCT GAGAGACGG	*GGGGGGAAAAAAGGAAGAGAAAGGAGAGAAGG*
GGAT GGAGT T T AT T GAT CCAGGAGACAGAGCC	*GGCGGGATTTTAGGTAGACATTCGACAGATCC*
GGT AGGACT T T AAACAAGCACCT GACAGACCC	*GGGCGGAATTTACCAACTCAAAGGACAGAACC*
T T CACCGGT T CAGGAAAT GAGACT AGGCAGGA	*TTGTCCGCTTCACCTATAGACTGTAGGCACGA*
T T CCCCGGGGAAT T AACGT AGACT AT GCAGTC	*TTGGCCGCGGAAAATAGCTACTGTATGCACT C*
T T CT CCGGAAGACCAAT ACAGACT ACGCAGCT	*TTGACCGCAAGAGGTAATCACTGTACGCACCT*
T T CGCCGGCCT AAAAAGCAAGACT AAGCAGAG	*T TGCCCGCCCTAT TTACGAACTGTAAGCACAG*
T T CAGGCGAACACCT AAT GAGT CTAGCGAGGT	*T T GTGGCCAACAGGAATAGACAGTAGCGACGT*
T T CCGGCGCCAAAAT ACGT AGT CTAT CGAGT G	*T TGGGGCCCCAATTAAGCTACAGTATCGACTG*
T T CT GGCGT T GAGGT AT ACAGTCTACCGAGCA	*T TGAGGCCT TGACCAAAT CACAGTACCGACCA*
T T CGGGCGGGT ATTTAGCAAGTCTAACGAGAC	*T TGCGGCCGGTAAAAACGAACAGT AACGACAC*
CCT AT T GGCCT AGGAAACGAGATCAGGTAGGA	*CCGCTTGTCCTATTCACAGAT CGCAGGTATGA*
CCAT T T GCCCT ACCT AT GGACTACAGGTACGA	*CCCGTTGACCTAAAGAGT GAAGCCAGGT AAGA*
CCT T T T GGGGAACCAAT GCAGAT CACGTAGCT	*CCGGTTGT GGAAAACAGTCATCGCACGTATCT*
CCAAT T GCGGAAGGT AACCACT ACACGTACCT	*CCCCTTGAGGAAT T GACACAAGCCACGT AACT*
CCT AGGT GAATAT TCAACGAGCT CAGTGAGGC	*CCGCGGTTAATAGGAACAGATAGCAGTGATGC*
CCAT GGT CAATAAAGAT GGACGACAGTGACGC	*CCCGGGTAAATACCTAGT GAATCCAGTGAAGC*
CCT T GGT GTTAAAACATGCAGCTCACTGAGCG	*CCGGGGTTTTAACCAAGTCATAGCACT GATCG*
CCAAGGT CT TAATTGAACCACGACACTGACCG	*CCCCGGTATTAAGGTACACAATCCACTGAACG*
CCT AAACGGGTACCTAACGAGTTCAGCAAGGT	*CCGCAACTGGTAAAGACAGATGGCAGCAATGT*
CCAT AACCGGTAGGAAT GGACAACAGCAACGT	*CCCGAACAGGTATTCAGT GAACCCAGCAAAGT*
CCT T AACGCCAAGGTATGCAGTTCACCAAGCA	*CCGGAACT CCAAT TGAGTCATGGCACCAATCA*

CCAAAACCCCAACCAAACCACAACACCAACCA	CCCCAACACCAAAACACACAACCCACCAAACA
CCT ACCAGTTTAAAGAACGAGGT CAGACAGGG	CCGCCCATTTTACCTACAGATTGCAGACAT GG
CCAT CCACTTTAT TCAT GGACCACAGACACGG	CCCGCCAATTTAGGAAGTGAAACCAGACAAGG
CCT T CCAGAAAAT T GAT GCAGGT CACACAGCC	CCGGCCAT AAAAGGT AGT CAT T GCACACAT CC
CCAACCACAAAAAACAACCACCACACACACCC	CCCCCCAAAAAACCAACACAAACCACACAACC
AAGAGGGGAAGAGGAAAAGAGAGAAGGGAGGA	AACT GGGCAAGACCT AT T GACT CAAGGGACGA
AAGCGGGGCCT AT T AACCT AGAGAAT GGAGT C	AACGGGGCCCT AAAT AGGT ACT CAAT GGACT C
AAGT GGGGT T CACCAAT T CAGAGAACGGAGCT	AACAGGGCT T CAGGT AAACACT CAACGGACCT
AAGGGGGGGGAAAAAAGGAAGAGAAAGGAGAG	AACCGGGCGGAAT T T ACCAACT CAAAGGACAG
AAGACCCGT T GACCT AAAGAGT GAAGCCAGGT	AACT CCCCT T GAGGAAT T GACACAAGCCACGT
AACGCCCCGGT AT T AAGGT ACACAAT CCACT G	AAGT CCCGAACAGGT AT T CAGT GAACCCAGCA
AACACCCCAACACCAAAACACACAACCCACCA	AAGGCCCGCCAAT T T AGGAAGT GAAACCAGAC
AACCCCCCCCAAAAAACCAACACAAACCACAC	

IV.CONCLUSION

In this paper, we study some cyclic DNA codes over the matrix ring $M_2(F_2+uF_2)$. The DNA codes over R are studied which are obtained by using a Gray map and properties of cyclic codes. We have found some properties of generator polynomials of reversible and reversible-complement cyclic codes with even length over the ring $M_2(F_2+uF_2)$ using the mathematical structure of DNA. The condition we have given about a reversible cyclic code is reversible-complement. Moreover, we also give the GC weight over the ring $M_2(F_2+uF_2)$.

REFERENCES

[1] L. Adleman: Molecular computation of solutions to combinatorial problem. Science 266, 1021-1024(1994)

[2] I.Siap, T. Abualrub, A. Ghrayeb: Cyclic DNA codes over the ring $F_2[u]/(u^2-1)$ based on the deletion distance. J. Frankl. Inst. 346, 731-740 (2009)

[3] B. Yildiz, I. Siap, Cyclic DNA codes over the ring $F_2[u]/(u^4-1)$ and applications to DNA codes, Comput. Math. Appl. 63 (2012) 1169-1176

[4] J. Liang, L. Wang, On cyclic DNA codes over F_2+uF_2, J. Appl. Math. Comput. (2015), https://doi.org/10.1007/s12190-015-0892-8.

[5] A. Bayram, E.S. Oztas, I. Siap, Codes over F_4+uF_4 and some DNA applications, Des. Codes Cryptogr. (2015), https://doi.org/10.1007/s10623-015-0100-8.

[6] S. Zhu, X. Chen, Cyclic DNA codes over $F_2+uF_2+vF_2+uvF_2$, J. Appl. Math. Comput. 1 (15) (2015).

[7] N. Bennenni, K. Guenda, S. Mesnager, New DNA cyclic codes over rings, arX iv:1505.06263v1 [cs.IT], 2015.

[8] B. Dong: Self-Dual Cyclic Codes over $M_2(F_2+uF_2)$, EJSE, VOL 2, NO 3, 2019

[9] C. Bachoc: Applications of coding theory to the construction of modular lattices, J. Combinatorial Theory A 78(1), 92-119 (1997)

[10] T. Abualrub,I. Siap: Cyclic codes over the rings Z_2+uZ_2 and $Z_2+uZ_2+u^2Z_2$. Des. Codes Cryptogr. 42, 273-287 (2007).

[11] A. Marathe, A.E. Condon, Corn R.M.: On combinatorial DNA word design. J. Comput. Biol. 8, 201-220(2001).

[12] H.Q. Dinh, T.M. Vo: Repeated-root cyclic and negacyclic codes of prime power lengths with a fifinite commutative chain ring alphabet . East-West J. Math. 13, 207-224 (2011)

[13] K. Guenda,T.A. Gulliver: Construction of cyclic codes over F_2+uF_2 for DNA computing. AAECC 24(6), 445-459 (2013)

[14] B. Srinivasulu, M. Bhaintwal : Reversible cyclic codes over F_4+uF_4 and their applications to DNA codes. IEEE Trans. Inf. Theory 55, 101-105 (2016).

[15] F. Gursoy,I. Siap,B. Yildiz: Construction of skew cyclic codes over F_q+vF_q. Adv. Math. Commun.8,313-322 (2014)

[16] P. Li, S.X. Zhu: Cyclic codes of arbitrary lengths over F_q+uF_q. J. Univ. Sci. Technol. China 38(12),1392-1396(2008)

[17] X.f Xu : On the structure of cyclic codes over $F_q+uF_q+vF_q+uvF_q$ [J]. Wuhan University Journal of Natural Sciences, 2011, 16(5):457.

In Table 3, φ - table for DNA correspondence are constructed.

Table 3: φ - table for DNA correspondence

Elements of R	Gray images	DNA codons
0	(0,0,0,0)	*AAAA*
uv	(1,1,1,1)	*GGGG*
uvw	(w,w,w,w)	*TTTT*
$(1+w)uv$	$(1+w,1+w,1+w,1+w)$	*CCCC*
v	(0,1,0,1)	*AGAG*
$v+uv$	(1,0,1,0)	*GAGA*
$v+uvw$	$(w,1+w,w,1+w)$	*TCTC*
$v+(1+w)uv$	$(1+w,w,1+w,w)$	*CTCT*
vw	$(0,w,0,w)$	*ATAT*
$vw+uv$	$(1,1+w,1,1+w)$	*GCGC*
$vw+uvw$	$(w,0,w,0)$	*TATA*
$vw+(1+w)uv$	$(1+w,1,1+w,1)$	*CGCG*
$uw+v$	$0,1+w,0,1+w)$	*ACAC*
$vw+v+uv$	$(1,w,1,w)$	*GTGT*
$vw+v+uvw$	$(w,1,w,1)$	*TGTG*
$uw+v+(1+w)uv$	$(1+w,0,1+w,0)$	*CACA*
u	(0,0,1,1)	*AAGG*
$u+uv$	(1,1,0,0)	*GGAA*
$u+uvw$	$(w,w,1+w,1+w)$	*TTCC*
$u+(1+w)uv$	$(1+w,1+w,w,w)$	*CCTT*
uw	$(0,0,w,w)$	*AATT*
$uw+uv$	$(1,1,1+w,1+w)$	*GGCC*
$uw+uvw$	$(w,w,0,0)$	*TTAA*
$uw+(1+w)uv$	$(1+w,1+w,1,1)$	*CCGG*
$uw+u$	$(0,0,1+w,1+w)$	*AACC*
$uw+u+uv$	$(1,1,w,w)$	*GGTT*
$uw+u+uvw$	$(w,w,1,1)$	*TTGG*
$uw+u+(1+w)uv$	$(1+w,1+w,0,0)$	*CCAA*
$u+v$	(0,1,1,0)	*AGGA*
$u+v+uv$	(1,0,0,1)	*GAAG*
$u+v+uvw$	$(w,1+w,1+w,w)$	*TCCT*
$u+v+(1+w)uv$	$(1+w,w,w,1+w)$	*CTTC*
$u+vw$	$(0,w,1,w)$	*ATGC*
$u+vw+uv$	$(1,1+w,0,w)$	*GCAT*
$u+vw+uvw$	$(w,0,1+w,1)$	*TACG*
$u+vw+(1+w)uv$	$(1+w,1,w,0)$	*CGTA*
$u+v+uw$	$(0,1+w,1,w)$	*ACGT*
$u+v+uw+uv$	$(1,1+w,0,1+w)$	*GTAC*
$u+v+uw+uvw$	$(w,1,1+w,0)$	*TGCA*
$u+v+uw+(1+w)uv$	$(1+w,0,w,1)$	*CATG*

$uw+v$	$(0,1,w,1+w)$	*AGTC*
$uw+v+uv$	$(1,0,1+w,w)$	*GACT*
$uw+v+uvw$	$(w,1+w,0,1)$	*TCAG*
$uw+v+(1+w)uv$	$(1+w,w,1,0)$	*CTGA*
$uw+vw$	$(0,w,w,0)$	*ATTA*
$uw+vw+uv$	$(1,1+w,1+w,1)$	*GCCG*
$uw+vw+uvw$	$(w,0,0,w)$	*TAAT*
$uw+vw+(1+w)uv$	$(1+w,1,1,1+w)$	*CGGC*
$uw+vw+v$	$(0,1+w,w,1)$	*ACTG*
$uw+vw+v+uv$	$(1,w,1+w,0)$	*GTCA*
$uw+vw+v+uvw$	$(w,1,0,1+w)$	*TGAC*
$uw+vw+v+(1+w)uv$	$(1+w,0,1,w)$	*CAGT*
$uw+u+v$	$(0,1,1+w,w)$	*AGCT*
$uw+u+v+uv$	$(1,0,w,1+w)$	*GATC*
$uw+u+v+uvw$	$(w,1+w,1,0)$	*TCGA*
$uw+u+v+(1+w)uv$	$(1+w,w,0,1)$	*CTAG*
$uw+u+vw$	$(0,w,1+w,1)$	*ATCG*
$uw+u+vw+uv$	$(1,1+w,w,0)$	*GCTA*
$uw+u+vw+uvw$	$(w,0,1,1+w)$	*TAGC*
$uw+u+vw+(1+w)uv$	$(1+w,1,0,w)$	*CGAT*
$uw+u+vw+v$	$(0,1+w,1+w,0)$	*ACCA*
$uw+u+vw+v+uv$	$(1,w,w,1)$	*GTTG*
$uw+u+vw+v+uvw$	$(w,1,1,w)$	*TGGT*
$uw+u+vw+v+(1+w)uv$	$(1+w,0,0,1+w)$	*CAAC*
1	$(0,0,0,1)$	*AAAG*
$1+uv$	$(1,1,1,0)$	*GGGA*
$1+uvw$	$(w,w,w,1+w)$	*TTTC*
$1+(1+w)uv$	$(1+w,1+w,1+w,w)$	*CCCT*
$1+v$	$(0,1,0,0)$	*AGAA*
$1+v+uv$	$(1,0,1,1)$	*GAGG*
$1+v+uvw$	$(w,1+w,w,w)$	*TCTT*
$1+v+(1+w)uv$	$(1+w,w,1+w,1+w)$	*CTCC*
$1+vw$	$(0,w,0,1+w)$	*ATAC*
$1+vw+uv$	$(1,1+w,1,w)$	*GCGT*
$1+vw+uvw$	$(w,0,w,1)$	*TATG*
$1+vw+(1+w)uv$	$(1+w,w,1+w,0)$	*CGCA*
$1+vw+v$	$(0,1+w,0,1)$	*ACAG*
$1+vw+v+uv$	$(1,w,1,1+w)$	*GTGC*
$1+vw+v+uvw$	$(w,1,w,0)$	*TGTA*
$1+vw+v+(1+w)uv$	$(1+w,0,1+w,1)$	*CACG*

$1+u$	$(0,0,1,0)$	$AAGA$
$1+u+uv$	$(1,1,0,1)$	$GGAG$
$1+u+uvw$	$(w,w,1+w,w)$	$TTCT$
$1+u+(1+w)uv$	$(1+w,1+w,w,1+w)$	$CCTC$
$1+uw$	$(0,0,w,1+w)$	$AATC$
$1+uw+uv$	$(1,1,1+w,w)$	$GGCT$
$1+uw+uvw$	$(w,w,0,1)$	$TTAG$
$1+uw+(1+w)uv$	$(1+w,1+w,1,0)$	$CCTA$
$1+uw+u$	$(0,0,1+w,w)$	$AACT$
$1+uw+u+uv$	$(1,1,w,1+w)$	$GGTC$
$1+uw+u+uvw$	$(w,w,1,0)$	$TTGA$
$1+uw+u+(1+w)uv$	$(1+w,1+w,0,1)$	$CCAG$
$1+u+v$	$(0,1,1,1)$	$AGGG$
$1+u+v+uv$	$(1,0,0,0)$	$GAAA$
$1+u+v+uvw$	$(w,1+w,1+w,1+w)$	$TCCC$
$1+u+v+(1+w)uv$	$(1+w,w,w,w)$	$CTTT$
$1+u+vw$	$(0,w,1,w)$	$ATGT$
$1+u+vw+uv$	$(1,1+w,0,1+w)$	$GCAC$
$1+u+vw+uvw$	$(w,0,1+w,0)$	$TACA$
$1+u+vw+(1+w)uv$	$(1+w,1,w,1)$	$CGTG$
$1+u+vw+v$	$(0,1+w,1,1+w)$	$ACGC$
$1+u+vw+v+uv$	$(1,w,0,1)$	$GTAG$
$1+u+vw+v+uvw$	$(w,1,1+w,1)$	$TGCG$
$1+u+vw+v+(1+w)uv$	$(1+w,0,w,0)$	$CATA$
$1+uw+v$	$(0,1,w,w)$	$AGTT$
$1+uw+v+uv$	$(1,0,1+w,1+w)$	$GACC$
$1+uw+v+uvw$	$(w,1+w,0,0)$	$TCAA$
$1+uw+v+(1+w)uv$	$(1+w,w,1,1)$	$CTGG$
$1+uw+vw$	$(0,w,w,1)$	$ATTG$
$1+uw+vw+uv$	$(1,1+w,1+w,0)$	$GCCA$
$1+uw+vw+uvw$	$(w,0,0,1+w)$	$TAAC$
$1+uw+vw+(1+w)uv$	$(1+w,1,1,w)$	$CGGT$
$1+uw+vw+v$	$(0,1+w,w,0)$	$ACTA$
$1+uw+vw+v+uv$	$(1,w,1+w,1)$	$GTCG$
$1+uw+vw+v+uvw$	$(w,1,0,w)$	$TGAT$
$1+uw+vw+v+(1+w)uv$	$(1+w,0,1,1+w)$	$CAGC$
$1+uw+u+v$	$(0,1,1+w,1+w)$	$AGCC$
$1+uw+u+v+uv$	$(1,0,w,w)$	$GATT$
$1+uw+u+v+uvw$	$(w,1+w,1,1)$	$TCGG$
$1+uw+u+v+(1+w)uv$	$(1+w,w,0,0)$	$CTAA$

http://dx.doi.org/10.21172/ijiet.172.01

$1+uw+u+vw$	$(0,w,1+w,1+w)$	*AGCC*
$1+uw+u+vw+uv$	$(1,0,w,w)$	*GATT*
$1+uw+u+vw+uvw$	$(w,1+w,1,1)$	*TCGG*
$1+uw+u+vw+(1+w)uv$	$(1+w,w,0,0)$	*CTAA*
$1+uw+u+vw+v$	$(0,1+w,1+w,1)$	*ACCG*
$1+uw+u+vw+v+uv$	$(1,w,w,0)$	*GTTA*
$1+uw+u+vw+v+uvw$	$(w,1,1,1+w)$	*TGGC*
$1+uw+u+vw+v+(1+w)uv$	$(1+w,0,0,w)$	*CAAT*
w	$(0,0,0,w)$	*AAAT*
$w+uv$	$(1,1,1,1+w)$	*GGGC*
$w+uvw$	$(w,w,w,0)$	*TTTA*
$w+(1+w)uv$	$(1+w,1+w,1+w,1)$	*CCCG*
$w+v$	$(0,1,0,w+1)$	*AGAC*
$w+v+uv$	$(1,0,1,w)$	*GAGT*
$w+v+uvw$	$(w,w+1,w,1)$	*TCTG*
$w+v+(1+w)uv$	$(1+w,w,1+w,0)$	*CTCA*
$w+vw$	$(0,w,0,0)$	*ATAA*
$w+vw+vu$	$(1,1+w,1,1)$	*GCGG*
$w+vw+vuw$	$(w,0,w,w)$	*TATT*
$w+vw+(1+w)vu$	$(1+w,1,1+w,1)$	*CGCG*
$w+vw+v$	$(0,1+w,0,1)$	*ACAG*
$w+vw+v+uv$	$(1,w,1,0)$	*CTCA*
$w+vw+v+uvw$	$(w,1,w,w+1)$	*TGTC*
$w+vw+v+(1+w)uv$	$(1+w,0,1+w,w)$	*CACT*
$w+u$	$(0,0,1,1+w)$	*AAGC*
$w+u+uv$	$(1,1,0,w)$	*GGAT*
$w+u+uvw$	$(w,w,w+1,1)$	*TTCG*
$w+u+(1+w)uv$	$(1+w,1+w,w,0)$	*CCTA*
$w+wu$	$(0,0,w,0)$	*AATA*
$w+wu+uv$	$(1,1,w+1,1)$	*GGCG*
$w+wu+uvw$	$(w,w,0,w)$	*TTAT*
$w+wu+(1+w)uv$	$(1+w,1+w,1,1+w)$	*CCGC*
$w+wu+u$	$(0,0,w+1,1)$	*AATG*
$w+wu+u+uv$	$(1,1,w,0)$	*GGTA*
$w+wu+u+uvw$	$(w,w,1,w+1)$	*TTGC*
$w+wu+u+(1+w)uv$	$(1+w,1+w,0,w)$	*CCAT*
$w+u+v$	$(0,1,1,w)$	*AGGT*
$w+u+v+uv$	$(1,0,0,w+1)$	*GAAC*
$w+u+v+uvw$	$(w,w+1,w+1,0)$	*TCCA*
$w+u+v+(1+w)uv$	$(1+w,w,w,1)$	*CTTG*

$w+u+vw$	$(0,w,1,1)$	*ATGG*
$w+u+vw+uv$	$(1,w+1,0,0)$	*GCAA*
$w+u+vw+uvw$	$(w,0,w+1,w+1)$	*TACC*
$w+u+vw+(1+w)uv$	$(1+w,1,w,w)$	*CGTT*
$w+u+vw+v$	$(0,w+1,1,0)$	*ACGA*
$w+u+vw+v+uv$	$(1,w,0,1)$	*GTAG*
$w+u+vw+v+uvw$	$(w,1,w+1,w)$	*TGCT*
$w+u+vw+v+(1+w)uv$	$(1+w,0,w,w+1)$	*CATC*
$w+uw+v$	$(0,1,w,1)$	*AGTG*
$w+uw+v+uv$	$(1,0,w+1,0)$	*GACA*
$w+uw+v+uvw$	$(w,w+1,0,w+1)$	*TCAC*
$w+uw+v+(1+w)uv$	$(1+w,w,1,w)$	*CTGT*
$w+uw+vw$	$(0,w,w,w)$	*ATTT*
$w+uw+vw+uv$	$(1,w+1,w+1,w+1)$	*GCCC*
$w+uw+vw+uvw$	$(w,0,0,0)$	*TAAA*
$w+uw+vw+(1+w)uv$	$(1+w,1,1,1)$	*CGGG*
$w+uw+vw+v$	$(0,w+1,w+1,w+1)$	*ACTC*
$w+uw+vw+v+uv$	$(1,w,w+1,w)$	*GTCT*
$w+uw+vw+v+uvw$	$(w,1,0,1)$	*TGAG*
$w+uw+vw+v+(1+w)uv$	$(1+w,0,1,0)$	*CAGA*
$w+uw+u+v$	$(0,1,w+1,0)$	*AGCA*
$w+uw+u+v+uv$	$(1,0,w,1)$	*GATG*
$w+uw+u+v+uvw$	$(w,w+1,1,w)$	*TCGT*
$w+uw+u+v+(1+w)uv$	$(w+1,w,0,w+1)$	*CTAC*
$w+uw+u+vw$	$(0,w,w+1,w+1)$	*ATCC*
$w+uw+u+vw+uv$	$(1,w+1,w,w)$	*GCTT*
$w+uw+u+vw+uvw$	$(w,0,1,1)$	*TAGG*
$w+uw+u+vw+(1+w)uv$	$(w+1,1,0,0)$	*CGAA*
$w+uw+u+vw+v$	$(0,w+1,w+1,w)$	*ACCT*
$w+uw+u+vw+v+uv$	$(1,w,w,w+1)$	*GTTC*
$w+uw+u+vw+v+uvw$	$(w,1,1,0)$	*TGGA*
$w+uw+u+vw+v+(1+w)uv$	$(w+1,0,0,1)$	*CAAG*
$w+1$	$(0,0,0,w+1)$	*AAAC*
$w+1+uv$	$(1,1,1,w)$	*GGGT*
$w+1+uvw$	$(w,w,w,1)$	*TTTG*
$w+1+(1+w)uv$	$(w+1,w+1,w+1,0)$	*CCCA*
$w+1+v$	$(0,1,0,w)$	*AGAT*
$w+1+v+uv$	$(1,0,1,w+1)$	*GAGC*
$w+1+v+uvw$	$(w,w+1,w,0)$	*TCTA*
$w+1+v+(1+w)uv$	$(w+1,w,w+1,1)$	*CTCG*

$w+1+vw$	$(0,w,0,1)$	*ATAG*
$w+1+vw+uv$	$(1,w+1,1,0)$	*GCGA*
$w+1+vw+uv+uvw$	$(w,0,w,w+1)$	*TATC*
$w+1+vw+(1+w)uv$	$(w+1,1,w+1,w)$	*CGCT*
$w+1+vw+v$	$(0,w+1,0,0)$	*ACAA*
$w+1+vw+v+uv$	$(1,w,1,1)$	*GTGG*
$w+1+vw+v+uvw$	$(w,1,w,w)$	*TGTT*
$w+1+vw+v+(1+w)uv$	$(w+1,0,w+1,w+1)$	*CACC*
$w+1+u$	$(0,0,1,w)$	*AAGT*
$w+1+u+uv$	$(1,1,0,w+1)$	*GGAC*
$w+1+u+uvw$	$(w,w,w+1,0)$	*TTCA*
$w+1+u+(1+w)uv$	$(w+1,w+1,w,1)$	*CCTG*
$w+1+uw$	$(0,0,w,1)$	*AATG*
$w+1+uw+uv$	$(1,1,w+1,0)$	*GGCA*
$w+1+uw+uvw$	$(w,w,0,w+1)$	*TTAC*
$w+1+uw+(1+w)uv$	$(w+1,w+1,1,w)$	*CCGT*
$w+1+uw+u$	$(0,0,w+1,0)$	*AACA*
$w+1+uw+u+uv$	$(1,1,w,1)$	*GGTG*
$w+1+uw+u+uvw$	$(w,w,1,w)$	*TTGT*
$w+1+uw+u+(1+w)uv$	$(w+1,w+1,0,w+1)$	*CCAC*
$w+1+u+v$	$(0,1,1,w+1)$	*AGGC*
$w+1+u+v+uv$	$(1,0,0,w)$	*GAAT*
$w+1+u+v+uvw$	$(w,w+1,w+1,1)$	*TCCC*
$w+1+u+v+(1+w)uv$	$(w+1,w,w,0)$	*CTTA*
$w+1+u+vw$	$(0,w,1,0)$	*ATGA*
$w+1+u+vw+uv$	$(1,w+1,1,0)$	*GCAG*
$w+1+u+vw+uvw$	$(w,0,w+1,w)$	*TACT*
$w+1+u+vw+(1+w)uv$	$(w+1,1,w,w+1)$	*CGTC*
$w+1+u+vw+v$	$(0,w+1,1,1)$	*ACGG*
$w+1+u+vw+v+uv$	$(1,w,0,0)$	*GTAA*
$w+1+u+vw+v+uvw$	$(w,1,w+1,w+1)$	*TGCC*
$w+1+u+vw+v+(1+w)uv$	$(w+1,0,w,w)$	*CATT*
$w+1+uw+v$	$(0,1,w,0)$	*AGTA*
$w+1+uw+v+uv$	$(1,0,w+1,1)$	*GACG*
$w+1+uw+v+uvw$	$(w,w+1,0,w)$	*TCAT*
$w+1+uw+v+(1+w)uv$	$(w+1,w,1,w+1)$	*CTGC*
$w+1+uw+vw$	$(0,w,w,w+1)$	*ATTC*
$w+1+uw+vw+uv$	$(1,w+1,w+1,w)$	*GCCT*
$w+1+uw+vw+uvw$	$(w,0,0,1)$	*TAAG*
$w+1+uw+vw+(1+w)uv$	$(w+1,1,1,0)$	*CGGA*

$w+1+uw+vw+v$	$(0,w+1,w,w)$	*ACTT*
$w+1+uw+vw+v+uv$	$(1,w,w+1,w+1)$	*GTCC*
$w+1+uw+vw+v+uvw$	$(w,1,0,0)$	*TGAA*
$w+1+uw+vw+v+(1+w)uv$	$(w+1,0,1,1)$	*CAGG*
$w+1+uw+v+u$	$(0,1,w+1,1)$	*AGCG*
$w+1+uw+v+u+uv$	$(1,0,w,0)$	*GATA*
$w+1+uw+v+u+uvw$	$(w,w+1,0,w+1)$	*TCAC*
$w+1+uw+v+u+(1+w)uv$	$(w+1,w,0,w)$	*CTAT*
$w+1+uw+vw+u$	$(0,w,w+1,w)$	*ATCT*
$w+1+uw+vw+u+uv$	$(1,w+1,w,w+1)$	*GCTC*
$w+1+uw+vw+u+uvw$	$(w,0,1,0)$	*TAGA*
$w+1+uw+vw+u+(1+w)uv$	$(w+1,1,0,1)$	*CGAG*
$w+1+uw+u+vw+v$	$(0,w+1,w+1,w+1)$	*ACCC*
$w+1+uw+u+vw+v+uv$	$(1,w,w,w)$	*GTTT*
$w+1+uw+u+vw+v+uvw$	$(w,1,1,1)$	*TGGG*
$w+1+uw+u+vw+v+(1+w)uv$	$(w+1,0,0,0)$	*CAAA*