A-012

IVPITEL は郭公の翼を広げている（其の一）

―ギリシャ・ローマ数の符号理論－

和田　平司　　大庭　温人　　大庭　裕子　　大庭　ゆづき　　林　郁枝

IVPITEL include with The Wing of a Cucko
― Girico・Roman's Numeric For The Code Theory －

Heiji Wada　Haruto Ooba　Yuko Ooba　Yuzuki Ooba　Ikue Hayashi
（所属なし）

あらまし　和田、林らはギリシャ・ローマ数が Codon Code を形成することを導いた。
その符号は Codon Code に似ていて冗長のない即ち符号に検査点数部を持たない符号を構成し、一誤り訂正符号を作ることができたので報告する。

キーワード　ギリシャ・ローマ数、符号理論、Codon Code

1. はじめに

従来の符号理論ではハミング符号等を考えても冗長のある検査点数部を付加する。

そのことによって符号の効率が符号の構成によって決まって仕舞う。

本論文はギリシャ・ローマ数で符号を構成できることを導いたので報告する。

それは Codon Code に似ていて、冗長のない符号を構成することを導いた。

一誤り訂正符号を作ることができ、Codon Code と同じように制御 Codon Code が重要な役割を担う。

一誤り訂正符号を Codon Code に割り付けると代表的な 20 種類のアミノ酸基の符号を構成する。

復号化行列を H_C^T としてある法則に則って作ることができる。

此の H_C^T によりシンドロームを求めることにより一誤り訂正が可能となる。

2. 本論

2-1）符号化について

ギリシャ・ローマ数の刻印　I,V,X に（00）の状態を加えて 4 つの情報源を作る。

I＝（01）、V＝（10）、X＝（11）、Ø＝（00）とする。これは 2bit のデータとして割り付けられる。

この 4 つの情報源を Codon Code と対応づける。Codon Code は 4 つのヌクレオチドから成り立っていることが分かっている。

即ち、4 つの情報源とヌクレオチドを対応づける。I≡C≡（01）、V≡G≡（10）、X＝凵≡（11）、Ø≡A≡（00）に対応づける。

すると、3 つのヌクレオチドの組み合わせを 1 つの Codon Code と見なすことができる。

これを符号化すると 64 通りの符号ができる。

それは 64 通りのアミノ酸と等価である。

アミノ酸の種類は 20 種類であることが分かっているので、16 通りのアミノ酸と 4 種類の制御 Codon Code 即ち 4 種の制御アミノ酸に、特殊 Code（XXX≡⌊⌋⌊⌋⌊⌋≡111111）を考える。

よって、符号行列を Codon Code と同じように 3 種類のヌクレオチドとギリシャ・ローマ教と対応づけると 3 つのヌクレオチドで 1 つの情報点数列と考える。すると符号行列 G は

$$G=\begin{bmatrix}100000\\010000\\001000\\000100\\000010\\000001\end{bmatrix} \quad \cdots\cdots\cdots\cdots (1)$$

①式で与えられる。

2-2）復号化について

ここで 3 つのヌクレオチドを 1 つの Codon Code とみると、これを 1 つの情報源とする。

その時の情報点数列の符号長を n とする。本論文の冗長のない一誤り訂正符号 n は次式で与えられる。

$$n=i\cdot m=i\cdot k \quad \cdots\cdots \quad (2)$$

（k を最小情報点数列とすると k=m である。ここで i を階数とする。）

すると、復号行列 $H_C{}^T$ は③式で与えられる。

$$H_C{}^T=\begin{bmatrix}P_1\\ \cdot\\ Pi\\ \cdot\\ P_m\end{bmatrix}=\begin{bmatrix}P_1\\P_2\\P_3\end{bmatrix}=\begin{bmatrix}P_{11}\cdots\cdots P_{1m}\\P21\cdots\cdots P_{2m}\\ \cdot \quad \cdot\\P_{i1}\cdots\cdots P_{im}\\ \hline P_2\text{の並び替え}\\ \cdot\\P_{m1}\cdots\cdots P_{mm}\end{bmatrix} \quad \cdots\cdots (3)$$

ここで P_2・P_i は P_1 とその他の値での並び替え

P_1 は情報点数列の値で決まる。

そして P_{mm} は本論文の場合、i＝3 で即ち m＝2 である。符号距離 d は d ≦n－m

2-3）シンドロームの計算と一誤り訂正について

情報点数列を（VII）の 3 つのヌクレオチドからなる Codon Code とする。ただし i=3、m＝2 とする。この場合、復号行列 $H_C{}^T$ は③式より

$$H_C{}^T=\begin{bmatrix}10\\01\\11\\01\\10\\11\end{bmatrix}\begin{matrix}\rightarrow P_1\\ \\ \rightarrow P_2\\ \\ \rightarrow P_3\\ \end{matrix} \quad \cdots\cdots\cdots\cdots\cdots (4)$$

とすると、誤りがない場合のシンドロームを計算する。

$$(VII)\cdot \begin{bmatrix}H_C{}^T\end{bmatrix}=(100101)\begin{bmatrix}10\\01\\11\\01\\10\\11\end{bmatrix}$$

$$=(00)$$

ここで、情報点数列の 2bit 目に誤りがあるとすると

VII+e＝（100101）＋（010000）
＝（110101）となる。

このときのシンドロームを計算すると

$$((VII)+e)\cdot H_T{}^C = (110101)\begin{bmatrix}10\\01\\11\\01\\10\\11\end{bmatrix} = (01)$$

となる。

このことにより、一誤り検出ができる。
次に、Codon Code のアミノ酸の配列により、シンドロームの計算から情報点数列の 2bit 目が誤りであることが分かる。

2-4）Codon Code のアルゴリズムについて

一誤り訂正が行えるのは、基本の 16 通りのアミノ酸と制御 Codon Code の 4 通りで 20 通りの代表のアミノ酸に適用する。

そこで、一誤りの訂正の場合 Codon Code の Type を 4 つに分ける。それぞれの Type を区別する為に制御 Codon Code を用いる。

その Codon Code のアルゴリズムを以下に示す。

XXX……//Codon Code の始まり
III……　//Type　V
：
：　VIV、ØXØ
：　　//並び替えたもの

XXX　　//Codon Code の始まり
VVV　//Type　I
：
：　IVI、ØXØ
：　　　//並び替えたもの

XXX　//Codon Code の始まり
VØX　//Type VØX
：
：　IXI、ØXØ
：　　//並び替えたもの

XXX　//Codon Code の始まり
IØX　//Type　IØX
：
：　VXV、ØXØ
：　　//並び替えたもの

XXX
XXX　//Codon Code の終了

注）XXX
　　XXX　終止　即ち Codon Code の終了メッセージになるまで
Type　I、V、IØX、VØZ はランダムに繰り返される。

2-5）一誤り訂正の復号行列 $H_C{}^T$ の作り方

$$H_c{}^t = \begin{bmatrix} a_1\,a_2 \\ b_1 b_2 \\ P_2\ \text{並び替え} \\ P_3\ \text{並び替え} \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \cdots\cdots (4)$$

$H_C{}^T$＝を④式で与えられる 6 行 2 列の行列とする。

復号行列 $H_C{}^T$ の作り方

§1 先頭のデータ値（刻印の値）を 1 行目にもってくる。

§2　次のデータ値（刻印の値）を 2 行目にもってくる。

§3　3 行目のデータ値は 1 行目、2 行目で使われていない（刻印の値）をもってくる。4,5,6 行目は 1,2,3 行目と異なる。

§4　4 行目と 5 行目は通常 1 行目と 2 行目を入れ替える。

§5　6 行目は残りのデータ（刻印の値）である。

以上が基本の作り方であり、注意として同じデータの値（刻印の値）が続く場合は 1 行目は先頭のデータ、2 行目は“11”のデータ、3 行目は残りのデータ（刻印の値）である。4,5,6 行目は 1,2,3 行目の値の並び替え。

2-6）復号化について

Codon Code の 3 つのヌクレオチドをギリシャ・ローマ教の刻印と“00”の値の 3 つのヌクレオチドに対応づけると 64 通りの Codon Code と同じ符号が構成される。

表 2 に 64 通りの Codon Code を示す。

	U	G	C	A
U	UU※	UG※	UC※	UA※
G	GU※	GG※	GC※	GA※
C	CU※	CG※	CC※	CA※
A	AU※	AG※	AC※	AA※

表 1　符号（Codon Code）の割付け

（U≡11≡X、G≡10≡V、C≡01≡I、A≡“00”）

64 通りの Codon Code は 20 種類のアミノ酸のグループのいずれかに割り付けられる。

64 通り全部の Codon Code のを情報点数列を考えると、一誤り検出符号となる。

3.考察

3-1）復号行列 H_C^T の抽出について

Codon Code のアルゴリズムに則って、Type のどの Codon Code か分かっているので、復号行列 H_C^T 群の中から 1 つの特定の H_C^T を選び出すことができる。

また、復号行列 H_C^T と Codon Code の情報点数列の情報の値により、受信した先頭のヌクレオチドの値で復号行列 H_C^T 群の中から 1 つの特定の H_C^T を選び出す。

3-2）アミノ酸の特質について

アミノ酸は Codon Code の場合、20 種類に大別される訳である。

ギリシャ・ローマ数を用いて一誤り訂正の符号を組み立てると 20 種類に大別される訳であるから

ギリシャ・ローマ教を用いて一誤り訂正の符号を組み立てると 20 種類の代表的なアミノ酸の一誤り訂正符号を作ることができる。

実際の Codon Code は誤り訂正を行っておらず、簡単な伝達方式をとっている。

そこで、Codon Code の全ての 64 通りのアミノ酸の一誤り訂正はできず。

この場合は一誤り検出は 64 通りの Codon Code に於いて行われる。Codon Code 自体には冗長部のない一誤り検出を行うことができる。

3-3）Type の切替えについて

Type の切換えができる理由の違いの表を表 2 に示す

	誤りがない場合	一誤りがある場合
III	i）I が 3 個 $H_C^T=\begin{bmatrix}01\\11\\10\\10\\11\\01\end{bmatrix}$	（i）I が 2 個で（00）か（X）が存在する
VVV	（i）V が 3 個 $H_C^T=\begin{bmatrix}10\\11\\01\\10\\11\\01\end{bmatrix}$	（i）V が 2 個で（00）か（X）が存在する
IØX	（i）I が 1 個 （ii）00 が存在 $H_C^T=\begin{bmatrix}01\\11\\10\\10\\11\\01\end{bmatrix}$	（i）I が 2 個 (ii)I が 2 個でØが 1 個存在する （iii）X と 00 が存在 （iv）00 が 2 個存在
VØX	（i）V が 1 個 （ii）00 が存在 $H_C^T=\begin{bmatrix}10\\11\\01\\10\\01\\11\end{bmatrix}$	（i）V が 2 個 （ii）V が 2 個でØが 1 個存在する （iii）X と 00 が存在 （iv）00 が 2 個存在

表 2　Type の切換えが出来る理由

表の違いによりそれぞれ復号行列 H_C^T が違う様に作れるので必ず Type の識別ができる。

3-4）復号行列 H_C^T について

（a）mが 2 bit の場合、4 つの状態を得る Codon Code に対応し最小情報点数 K＝2 となり階数 i＝3、即ち 3 つのヌクレオチドに匹敵する。

（b）mが 3bit の場合、6 つの状態を得る code に対応し最小情報点数 K＝3 となり階数 i＝2 に匹敵する。

以下同じ様にmが 4 bit の場合について考えれば良い。

例題として k ＝m＝3bit の場合について復号行列 H_C^T を求めると i=2 となり

$$H_C^T = \begin{bmatrix} P_1 \\ \cdot \\ P_i \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$$

$$= \begin{bmatrix} 011 \\ 110 \\ 101 \\ 110 \\ 101 \\ 011 \end{bmatrix} \quad \cdots\cdots\cdots (5)$$

⑤式で与えられる。

先頭の符号が（011）であるとすると、情報点数列は n＝（011110）となりシンドローム S を求めると

$$S = n \cdot H_C^T = (011110) \begin{bmatrix} 011 \\ 110 \\ 101 \\ 110 \\ 101 \\ 011 \end{bmatrix}$$

＝（000）となる。

此の場合誤りがないことを表す。

情報点数列に一誤りが発生すると一誤り検出が行える。

4.結論

ギリシャ・ローマ数を用いるとｍ－RNA の Codon Code と同じ様な仕組みの復号定理を導くことができた。

それはギリシャ・ローマ数の刻印を 2bit のデータ（最小情報点数）で割り付けることにより Codon Code と同じ 3 つのヌクレオチドで符号を構成する。

ギリシャ・ローマ数の刻印を 3 つ組み合わせることにより Codon Code と同じ動きをし 64 通りの Codon Code を導ける。

さらに 20 種類のアミノ酸の基本 code と符号長 n の情報点数列とすると、一誤り訂正符号として利用できることが分かった。

一般には一誤り検出符号である。

開始 Codon Code を XXX というギリシャ・ローマ数で割り付けると終止 Codon Code を XXX と XXX が 2 つ現れるまで Codon Code は組み立てられる。

残された課題として、一般的に Codon Code は 64 通り作られるのでその時に一誤り訂正符号が適用されるアルゴリズムの検討が残されている。

一般的には一誤り検出の場合のアルゴリズムは導ける。

参考文献

（1） 和田、三角田ら “ギリシャ・ローマ数の結合子について”
FIT2018、P349～P350（2018）

（2） 和田、大庭ら“IVPITEL は雨に乗って遣ってくる”情報処理学会全国大会
P1-149～P～150（2021）

（3） 若山正人“電子情報通信学会誌”Vol160、№7、P1280～P1284（2017）

（4） 東京教育大学数学教育研究会“少年少女と算数の教育”小峰書店（昭和 56 年）