The relation between k-circularity and circularity of codes

Elena Fimmel, Christian J Michel, François Pirot, Jean-Sébastien Sereni, Martin Starman, Lutz Strüngmann

The relation between k-circularity and circularity of codes

ELENA FIMMEL[1], CHRISTIAN J. MICHEL[2,*], FRANÇOIS PIROT[2,3], JEAN-SÉBASTIEN SERENI[2], MARTIN STARMAN[1,2] AND LUTZ STRÜNGMANN[1]

[1] *Institute of Mathematical Biology*
Faculty for Computer Sciences
Mannheim University of Applied Sciences
68163 Mannheim, Germany

[2] *Theoretical Bioinformatics, ICube,*
C.N.R.S., University of Strasbourg,
300 Boulevard Sébastien Brant
67400 Illkirch, France
[*] *Corresponding author*

[3] *LORIA (Orpailleur)*
C.N.R.S., University of Lorraine, INRIA
Campus scientifique
54506 Vandœuvre-lès-Nancy Cedex, France

Abstract. A code X is k-circular if any concatenation of at most k words from X, when read on a circle, admits exactly one partition into words from X. It is circular if it is k-circular for every integer k. While it is not *a priori* clear from the definition, there exists, for every pair (n, ℓ), an integer k such that every k-circular ℓ-letter code over an alphabet of cardinality n is circular, and we determine the least such integer k for all values of n and ℓ. The k-circular codes may represent an important evolutionary step between the circular codes, such as the comma-free codes, and the genetic code.

1. Introduction

The discovery of the DNA structure by Watson and Crick in 1953 [**37**] spurred a new branch of mathematical biology which overlaps the theory of block codes, *i.e.* codes consisting of words of a fixed length over some finite alphabet. A relevant concept in this biomathematical context is that of comma-freeness, meaning that code words can be separated without using extra symbols ("commas"). This concept was later weakened to that of a circular code, meaning that the reading frame can always be retrieved in any word written on a circle.

The DNA structure is a sequence of nucleotides on the 4-letter alphabet $\{A, C, G, T\}$, where A stands for adenine, C for cytosine, G for guanine and T for thymine, organized in an antiparallel and

E-mail address: e.fimmel@hs-mannheim.de, l.struengmann@hs-mannheim.de, m.starman@live.com, c.michel@unistra.fr, sereni@kam.mff.cuni.cz, francois.pirot@loria.fr.

Date: June 12, 2020.

Key words and phrases. circular code; k-circular code; genetic code; code evolution.

complementary double helix. A (protein coding) gene is a DNA sequence which is read during the translation process by words of 3 letters also called trinucleotides or codons. The genetic code is a map between the 64 possible codons and the 20 amino acids constituting the proteins and the 3 stop codons. Soon after this discovery, scientists believed that the redundancy in the codon amino acid assignment must be some kind of code used by nature for error detection. Crick, Griffith and Orgel in 1957 [**4**] proposed that in such a code, no codon can be obtained by concatenating a non-empty suffix and a non-empty prefix of codons in the code. It follows that a frameshift during translation would be detected immediately. This is the definition of comma-free codes.

One year later, in 1958, Golomb and Welch, with Gordon [**17**] and with Delbrück [**18**], introduced a mathematical definition of comma-free codes as block-codes. One of their discoveries showed that a comma-free code on a 4-letter alphabet with words of 3 letters is maximal if the code contains exactly 20 of the 64 trinucleotides. In other words, there is no comma-free code with 21 and more trinucleotides. The interesting fact that the maximal number of codons in a comma-free code is equal to the number of amino acids motivated theoretical researchers all over the world in the late 1950s to solve the gene coding.

Only a short period later, at the beginning of the 1960s, it was finally proven that the genetic code cannot be a comma-free code as the trinucleotide TTT, an excluded trinucleotide in a comma-free code, codes an amino acid: the phenylalanine [**34**]. Within the following years, comma-free codes gained almost no attention in biology, and were rather studied from the point of view of coding and information theory; the work by Lam in 2003 [**21**], entitled: "Completing comma-free codes" seemed to uncover the last relevant properties of comma-free codes.

Before Lam's work, Arquès and Michel in 1996 [**1**] discovered after a statistical computation of the 64 trinucleotides in each of the three frames of genes of bacteria and eukaryotes ($2 \cdot 3 \cdot 64 = 384$ trinucleotides analysed), that a simple inspection identifies 20 codons which occur preferentially in reading frames (compared to the two shifted frames). Furthermore, this set X of 20 codons forms a circular code:

$$X = \{AAC, AAT, ACC, ATC, ATT, CAG, CTC, CTG, GAA, GAC, GAG, GAT, GCC, GGC, GGT, GTA, GTC, GTT, TAC, TTC\}. \tag{1.1}$$

Circular codes come from the same family as comma-free codes, but are slightly less restrictive [**3**, p. 233]. They enable to discover frameshifts not immediately, but after the reading of a finite (bounded) number of codons (called letter window of the circular code). Thus, they may have an important function of frame retrieval during the translational process in the ribosome. Figure 1 illustrates that in any sequence of trinucleotides from X a frameshift of one or two positions will be detected. The set X identified in genes has additional strong properties: it is also self-complementary and C^3 [**1**]. A code is

Frame 0: A T G ... G G T A A T T A C G A G T A C A C C ... T A A
Frame 1: A T G ... G G T A A T T A C G A G T A C A C C ... T A A
Frame 2: A T G ... G G T A A T T A C G A G T A C A C C ... T A A

FIGURE 1. Reading frame retrieval in genes with the trinucleotide circular code $X = \{AAC, AAT, ACC, ATC, ATT, CAG, CTC, CTG, GAA, GAC, GAG, GAT, GCC, GGC, GGT, GTA, GTC, GTT, TAC, TTC\}$ identified in genes. A frameshift is detected after a few nucleotides, specifically at most 13. The trinucleotides underlined in blue belong to X, the trinucleotides underlined in red do not belong to X.

self-complementary if each codon in the code is associated with its anticodon in the code. A circular code is C^3 if the two codes resulting of the two circular permutations of all codons of the code are also circular (reviews in [**15**, **22**] for further details on these properties).

Three approaches were developed to study circular codes (classes, hierarchies: (strong comma-free, comma-free, circular), numbers, properties of their prefixes and suffixes, window length to retrieve the word decomposition, *etc.*): (i) the flower automaton [**1**]; (ii) the necklaces LDN (letter diletter necklace) and DLN (diletter letter necklace) [**31**, **32**, **36**] extended to $(n+1)LDCCN$ (letter diletter continued closed necklaces) [**29**]; and (iii) the recent and powerful approach based on graph theory proposed by Fimmel *et al.* in 2016 [**10**] which allows to characterize the strong comma-free codes [**12**], the comma-free codes and the circular codes according to the properties of their associated graphs. The graph associated to a code is directed and depending on its paths, the class of the code can be deduced. As we shall see, even the minimum length of a sequence needed to ensure the discovery of a frameshift can be efficiently deduced from this graph.

The study of the biological function of circular codes in genes is progressing. Firstly, the circular code X (1.1) is also identified in genes of archaea, plasmids and viruses, in addition to bacteria and eukaryotes, and by two different statistical approaches [**26**, **27**]. Furthermore, motifs from the circular code X, called *X-motifs*, are significantly enriched in the genes of most organisms, from bacteria to eukaryotes [**5**, **28**], and also exist in the ribosomal and transfer RNAs (rRNAs and tRNAs) [**6**–**8**, **23**, **24**, **33**]. However, these X-motifs in genes are separated by words which are not in the code X. In order to find a theory which may give a solution to this problem, this work aims at the class of k-circular codes which is another code family of block codes. The k-circular codes are even less restrictive than circular codes and can potentially explain the gaps between the X-motifs in sequences.

This work deals with the general definition of circular codes followed by the boundaries of such. First, it is not evident from the definition itself that such a boundary exists. However, from previous works [**10**, **11**], one realises that checking whether a given code is circular can be done efficiently by building the associated graph (see Definition 3.1) and checking whether it contains a cycle. Such an argument traces back a potentially infinite task to a finite one. We delve more into the properties of a non-circular code that are captured by the cycles of its associated graph and uncover a link to the k-circularity of the code. Specifically, we determine precisely, for every finite alphabet Σ and every word length ℓ, the existence of an integer k such that every k-circular ℓ-letter code over Σ must also be circular. Further, we determine the exact boundary between k-circularity and circularity by finding the least such integer k.

Section 2 provides formal definitions and notations. We establish in Section 3 a graph characterization of the notion of k-circularity with Theorem 3.3. Section 4 states our main results: Theorems 4.1 and 4.4. The Existence-Theorem 4.1 is demonstrated with a proof based on Theorem 3.3. Section 5 is devoted to prove the Sharpness-Theorem 4.4. In Section 6, we propose an evolutionary hypothesis for the k-circular codes in gene translation. In particular, we prove that there are exactly 52 maximal 1-circular codes that encode all 20 amino acids, while the 12,964,440 maximal circular codes encode at most 18 amino acids.

2. Definitions and notations

Let Σ be an arbitrary finite alphabet of cardinality at least 2 and set $n := |\Sigma|$. To emphasise the special important case of the genetic alphabet, which has cardinality 4, we set $\mathcal{B} := \{A, C, G, T\}$. We use standard word-theory notation: $\Sigma^* := \bigcup_{\ell \geq 0} \Sigma^\ell$. In other words, Σ^* is the set of all finite words with letters in Σ, including the empty word.

If $w = w_1 \cdots w_\ell \in \Sigma^\ell$ for some $\ell \in \mathbf{N}$, then for every $j \in \{0, \ldots, \ell - 1\}$, the *circular j-shift* of w is the word $w_{j+1} \cdots w_\ell w_1 \cdots w_j$. In particular, the circular 0-shift of w is w itself. A word w' is a *circular shift* of w if w' is the circular j-shift of w for some $j \in \{0, \ldots, \ell - 1\}$. If $w' = w'_1 \cdots w'_{\ell'} \in \Sigma^{\ell'}$ then the *concatenation* of w and w' is the word

$$w \cdot w' \coloneqq w_1 \cdots w_\ell w'_1 \cdots w'_{\ell'} \in \Sigma^{\ell+\ell'}.$$

Suppose that $w = w_1 \cdot w_2$. If none of w_1 and w_2 is the empty word, then w_1 is a *proper prefix* of w and w_2 is a *proper suffix* of w.

Definition 2.1. Let Σ be a finite alphabet.

- For an integer $\ell \geq 2$, an *ℓ-letter code* is a subset of Σ^ℓ.
- For $w \in \Sigma^*$ and $X \subseteq \Sigma^*$, an *X-decomposition* of w is a tuple $(x_1, \ldots, x_t) \in X^t$ with $t \in \mathbf{N}$ such that $w = x_1 \cdot x_2 \cdots x_t$.

We now formally define the notion of circularity of a code.

Definition 2.2. Let $X \subseteq \Sigma^\ell$ be an ℓ-letter code.

- Let m be a positive integer and let $(c_1, \ldots, c_m) \in X^m$. A *circular X-decomposition* of the concatenation $c \coloneqq c_1 \cdots c_m$ is an X-decomposition of a circular shift of c.
- Let k be a non-negative integer. The code X is *k-circular* if for every $m \in \{1, \ldots, k\}$ and every m-tuple $(c_1, \ldots, c_m)$ of words in X, the concatenation $c_1 \cdots c_m$ admits a unique circular X-decomposition. Note that every code is trivially 0-circular.
- The code X is *circular* if it is k-circular for all $k \in \mathbf{N}$.

Several examples of codes that are k-circular but not $(k+1)$-circular will be given in Example 4.5, but to illustrate Definition 2.2, we state one example here.

Example 2.3. Let Σ be the binary alphabet $\{0, 1\}$. Then $X = \{0001, 0111, 1100\}$ is a 2-circular but not 3-circular 4-letter code. Indeed, computer calculations show that X is 2-circular but it is easy to see that the code X is not 3-circular since the word $w = 0111 \cdot 0001 \cdot 1100$ has a second circular X-decomposition: that of its 2-shift $1100 \cdot 0111 \cdot 0001$.

We will make use of graph theory to study the circularity of codes. To this end, we establish in the next section a graph characterization of k-circularity.

3. Graph characterization of k-circularity

A new graph approach for studying circular codes (cf. Definition 3.1) has been recently suggested [**10**]. It (non-trivially) generalises an older approach used for 2-letter words [**2**]. The original main results were formulated only for the genetic alphabet [**10**] but they could be easily extended to any letter words on any finite alphabet [**11**]. Thus, an ℓ-letter code is circular if and only if its associated graph is acyclic. Let us define the graph associated to a code.

Definition 3.1. Let ℓ be an integer greater than 1 and let $X \subseteq \Sigma^\ell$ be an ℓ-letter code. We define a graph $\mathcal{G}(X) = (V(X), E(X))$ with set of vertices $V(X)$ and set of arcs $E(X)$ as follows:

- $V(X)$ is composed of all proper prefixes and all proper suffixes of words in X,
- $E(X) \coloneqq \left\{ w_1 \to w_2 \,:\, w_1 \cdot w_2 \in X \text{ and } (w_1, w_2) \in V(X)^2 \right\}$.

The graph $\mathcal{G}(X)$ is the graph *associated* to X. For each $i \in \{1, \ldots, \ell\}$, the vertices of $\mathcal{G}(X)$ that correspond to words of length i are referred to as *i-nodes*.

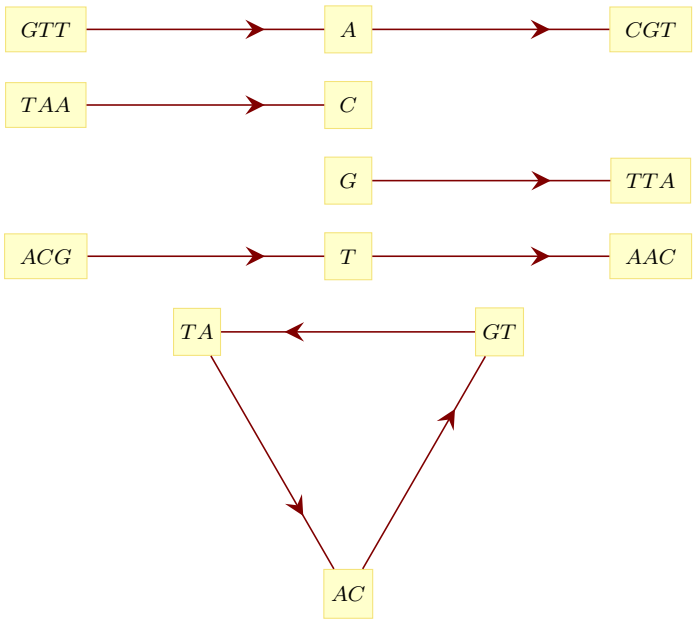

FIGURE 2. The graph $\mathcal{G}(X)$ of the tetranucleotide code $X = \{ACGT, GTTA, TAAC\}$.

Figure 2 illustrates Definition 3.1. Directed cycles in the graph associated to a code play an important role. It directly follows from Definition 3.1 that if X is an ℓ-letter code over an alphabet Σ of cardinality n, then for every arc e in $\mathcal{G}(X)$ there exists $i \in \{1, \ldots, \ell-1\}$ such that e goes from an i-node to an $(\ell - i)$-node. For each $i \in \{1, \ldots, \lfloor \ell/2 \rfloor\}$, the *$i$-component* of $\mathcal{G}(X)$ is defined to be the subgraph of $\mathcal{G}(X)$ induced by the set of j-nodes for $j \in \{i, \ell - i\}$. Each connected component of the underlying undirected graph of $\mathcal{G}(X)$ is thus contained in a j-component of $\mathcal{G}(X)$ for some $j \in \{1, \ldots, \ell - 1\}$. The next observation now readily follows.

OBSERVATION 3.2. *Given two integers ℓ and n both at least 2, let X be an ℓ-letter code over an alphabet Σ of cardinality n.*

(1) For every $j \in \{1, \ldots, \lfloor \ell/2 \rfloor\}$, the j-component of $\mathcal{G}(X)$ contains at most $n^j + n^{\ell - j}$ vertices if $j \neq \ell/2$ and at most $n^{\ell/2}$ otherwise.

(2) If ℓ is odd then $\mathcal{G}(X)$ contains no directed cycle of odd length.

(3) If ℓ is even then every directed cycle of odd length in $\mathcal{G}(X)$ is contained in the $(\ell/2)$-component.

(4) Suppose that $\mathcal{G}(X)$ contains a directed cycle of length t and let $j \in \{1, \ldots, \lfloor \ell/2 \rfloor\}$ such that this cycle is contained in the j-component. Then

$$t \leq \begin{cases} 2n^j & \text{if } j \neq \frac{\ell}{2}, \\ n^{\ell/2} & \text{if } j = \frac{\ell}{2}. \end{cases}$$

The k-circularity of dinucleotide codes was characterized in terms of graphs [**10**]:

- a dinucleotide code is 1-circular but not 2-circular if and only if its associated graph contains a Hamiltonian cycle of length 4;
- a dinucleotide code is 2-circular but not 3-circular if and only if its associated graph contains an oriented cycle of length 3 and has no Hamiltonian cycle; and
- a dinucleotide 3-circular code is circular.

Our next theorem is a natural generalization of this approach. We need the following notation. If X is a code then $g_o(X)$ and $g_e(X)$ are the respective lengths of the shortest directed cycles of odd length and of even length in $\mathcal{G}(X)$. We define $g_o(X) \coloneqq \infty$ and $g_e(X) \coloneqq \infty$ if cycles of odd length and of even length do not exist in $\mathcal{G}(X)$, respectively. If the code X is not circular then one of these numbers must be finite.

THEOREM 3.3. *Let Σ be a finite alphabet, ℓ an integer greater than 1 and $X \subseteq \Sigma^\ell$ an ℓ-letter code over Σ. Then the code X is k-circular and not $(k+1)$-circular if and only if*

$$k = \min\left\{g_o(X), \frac{g_e(X)}{2}\right\} - 1 < \infty.$$

PROOF. Suppose first that $w_1 \cdots w_r$ is a directed cycle in $\mathcal{G}(X)$. If r is even then the word $w_1 \cdots w_r$ admits two different circular X-decompositions into $r/2$ words from X, namely

$$w_1w_2|\ldots|w_{r-1}w_r \quad \text{and} \quad w_2w_3|\ldots|w_{r-2}w_{r-1}|w_rw_1.$$

It follows that X is not $\frac{1}{2}g_e(X)$-circular. If r is odd then the word $w_1 \cdots w_r w_1 \cdots w_r$ admits two different circular X-decompositions into r words from X, namely

$$w_1w_2|\ldots|w_{r-2}w_{r-1}|w_rw_1|w_2w_3|\ldots|w_{r-1}w_r \quad \text{and} \quad w_2w_3|\ldots|w_{r-1}w_r|w_1w_2|\ldots|w_{r-2}w_{r-1}|w_rw_1.$$

It follows that X is not $g_o(X)$-circular.
Conversely, suppose that both w and its circular j-shift admit a circular X-decomposition, where $j \in \{1, \ldots, \ell-1\}$. By setting $a_{r+1} \coloneqq a_1$, the word w can be written $a_1b_1|\ldots|a_rb_r$ such that $|a_i| = j$, $|b_i| = \ell - j$, and $a_ib_i, b_ia_{i+1} \in X$ for each $i \in \{1, \ldots, r\}$. It follows from Definition 3.1 that

$$W \coloneqq a_1 \to b_1 \to \ldots \to a_r \to b_r \to a_1$$

is a closed walk in $\mathcal{G}(X)$. Consequently, W either contains a directed cycle of even length, which then must be of length at most $|V(W)| = 2r$, or W decomposes into an even number of directed odd cycles, one of them thus having length at most r. Consequently, if X is not r-circular then $g_e(X) \leq 2r$ or $g_o(X) \leq r$. The conclusion follows. □

REMARK 3.4. The results from [**10**] are consistent with the statement of Theorem 3.3 above.

- If X is a 2-letter code that is 1-circular but not 2-circular, then $\mathcal{G}(X)$ contains cycles of lengths 3 and 4, and indeed $1 = \min\left\{3, \frac{4}{2}\right\} - 1$.
- If X is a 2-letter code that is 2-circular but not 3-circular, then $\mathcal{G}(X)$ contains a cycle of length 3 and no cycle of even length, and indeed $2 = \min\{3, \infty\} - 1$.

Theorem 3.3 implies, in particular, that if a code X is not circular, then the graph $\mathcal{G}(X)$ contains a cycle. By Observation 3.2, the length of a cycle in $\mathcal{G}(X)$ is bounded by a function f that depends only on the cardinality n of the alphabet and the length ℓ of the words in X. This implies the existence of a barrier $k = k(n, \ell)$ such that k-circularity implies circularity for every ℓ-letter code over an alphabet of cardinality n, for all values of ℓ and n. In the next section, we will state our main theorems that determine completely this barrier.

4. When does k-circularity imply circularity?

The main purpose of this section is to study when k-circularity implies circularity. It is easy to see that these two notions are not equivalent, the notion of k-circularity being weaker than that of circularity (see [**9**] and the examples below). Thus, it is natural to ask, given integers n and ℓ both at least 2, what is the least integer $k(n, \ell) \in \mathbf{N}$ such that every code $X \subseteq \Sigma^\ell$ that is $k(n, \ell)$-circular is circular.

(The cases where n or ℓ is 1 are trivial.) We provide a full answer to this question in our main results Theorem 4.1 (existence (and value) of a bound) and Theorem 4.4 (sharpness of the bound given).

EXISTENCE-THEOREM 4.1. *Let ℓ and n be integers both at least 2. Let Σ be an alphabet with $|\Sigma| = n$, and $X \subseteq \Sigma^\ell$ an ℓ-letter code over Σ. Set*

$$k(n,\ell) \coloneqq \begin{cases} n^{\frac{\ell-1}{2}} & \text{if } \ell \text{ is odd,} \\ n^{\ell/2} & \text{if } \ell \text{ is even and } n \text{ is odd,} \\ n^{\ell/2} - 1 & \text{if } \ell \text{ is even and } n \text{ is even.} \end{cases}$$

Then the code X is circular if and only if it is $k(n,\ell)$-circular.

Before we proceed with the proof of Theorem 4.1, let us note that it readily implies the following results with dinucleotide and trinucleotide circular codes over the genetic alphabet, which were established previously [**13**, **14**].

COROLLARY 4.2. *Let $\mathcal{B}$ be the genetic alphabet $\{A, C, G, T\}$.*

(1) A trinucleotide code $X \subseteq \mathcal{B}^3$ is circular if and only if X is 4-circular.
(2) A dinucleotide code $X \subseteq \mathcal{B}^2$ is circular if and only if X is 3-circular.

PROOF.

(1) In this case $n = 4$ and $\ell = 3$, and hence according to Theorem 4.1, the code X is circular if and only if it is $4^{\frac{3-1}{2}} = 4^1 = 4$-circular.
(2) In this case $n = 4$ and $\ell = 2$, and hence according to Theorem 4.1, the code X is circular if and only if it is $(4^{\frac{2}{2}} - 1) = 3$-circular.

□

We also apply Theorem 4.1 to the case of a binary alphabet, *i.e.* with $n = 2$, and obtain the following statement.

COROLLARY 4.3. *Let $\Sigma = \{0, 1\}$ and ℓ an integer greater than 1. A binary ℓ-letter code $X \subseteq \Sigma^\ell$ is circular if and only if X is*

(1) $2^{\frac{\ell-1}{2}}$-circular if ℓ is odd; or
(2) $(2^{\frac{\ell}{2}} - 1)$-circular if ℓ is even.

PROOF OF THE EXISTENCE-THEOREM 4.1. One direction is trivial: if X is circular then X is r-circular for every integer r, and hence for $k(n,\ell)$.
Conversely, let X be an ℓ-letter code that is r-circular but not $(r+1)$-circular. By Theorem 3.3, the graph $\mathcal{G}(X)$ contains a cycle of (even) length $2(r+1)$, or r is even and $\mathcal{G}(X)$ contains a cycle of (odd) length $r+1$.
(i) If ℓ is odd then Observation 3.2 yields that $g_o(X) = \infty$, and therefore in this case $r+1 = \frac{1}{2}g_e(X) < g_o(X)$. Because ℓ is odd, Observation 3.2 implies that

$$g_e(X) \leq \max\left\{2n^j \,:\, 1 \leq j \leq (\ell-1)/2\right\} = 2n^{\frac{\ell-1}{2}}.$$

Consequently, $r+1 \leq n^{(\ell-1)/2}$ and hence $r \leq k(n,\ell) - 1$.
(ii) Assume now that ℓ is even. Note that if $r+1 = \frac{1}{2}g_e(X)$ then by Observation 3.2,

$$r+1 \leq \max\left\{\frac{1}{2}\cdot n^{\ell/2}, n^{\ell/2-1}\right\} \leq n^{\ell/2} - 1.$$

So suppose that $r+1 = g_o(X) < \frac{1}{2}g_e(X)$. Observation 3.2 implies that $g_o(X) \leq n^{\ell/2}$, with equality only if n is odd since $g_o(X)$ is odd. Therefore,

$$r \leq \begin{cases} n^{\ell/2}-1 & \text{if } n \text{ is odd (and } \ell \text{ is even),} \\ n^{\ell/2}-2 & \text{if } n \text{ is even (and } \ell \text{ is even).} \end{cases}$$

We established that whenever

$$r \geq \begin{cases} n^{\frac{\ell-1}{2}} & \text{if } \ell \text{ is odd,} \\ n^{\ell/2} & \text{if } \ell \text{ is even and } n \text{ is odd,} \\ n^{\ell/2}-1 & \text{if } \ell \text{ is even and } n \text{ is even,} \end{cases}$$

every ℓ-letter code over an alphabet of cardinality n that is r-circular must be circular. This concludes the proof. □

The immediate question that remains open now is whether or not the bounds given in the Existence-Theorem 4.1 are sharp. This is indeed the case as our next theorem asserts. The proof takes several pages and we defer it to Section 5.

Sharpness-Theorem 4.4. *Given two integers n and ℓ both at least 2, let Σ be an alphabet with $|\Sigma| = n$ and $X \subseteq \Sigma^\ell$ an ℓ-letter code over Σ. Then $k(n,\ell)$ is the least integer r such that every code $X \subseteq \Sigma^\ell$ that is r-circular is circular.*

For the convenience of the reader, we close this section by giving explicitly codes that are $(k(n,\ell)-1)$-circular but not circular for some specific values of n and ℓ. They all come from the constructions presented in the proof of Theorem 4.4. These codes are the smallest existing ones, and they are also minimal regarding Theorem 3.3, as we will explain at the beginning of Section 5. Further examples for the case where ℓ is odd and $n \geq 3$ are given at the end of Section 5 after the general construction is explained.

Example 4.5.

(1) Let $n = 2$ and $\ell = 3$. Every 2-circular 3-letter code is circular, and there are 1- but not 2-circular 3-letter codes:
$$X(2,3) = \{010, 101\}.$$

(2) Let $n = 2$ and $\ell = 4$. Every 3-circular 4-letter code is circular, and there are 2- but not 3-circular 4-letter codes over $\{0,1\}$:
$$X(2,4) = \{0001, 0111, 1100\}.$$

(3) Let $n = 2$ and $\ell = 5$. Every 4-circular 5-letter code is circular, and there are 3- but not 4-circular 5-letter codes:
$$X(2,5) = \{00100, 01110, 10001, 11011\}.$$

(4) Let $n = 2$ and $\ell = 7$. Every 8-circular 7-letter code is circular, and there are 7- but not 8-circular 8-letter codes:
$$X(2,7) = \{0001000, 0010010, 0101100, 0111110, 1000001, 1011011, 1100101, 1110111\}.$$

(5) Let $n = 3$ and $\ell = 4$. Every 9-circular 4-letter code is circular, and there are 8- but not 9-circular 4-letter codes over $\{0,1,2\}$:
$$X(3,4) = \{0001, 0102, 0210, 1011, 1120, 2021, 2122, 2200\}.$$

(6) Let $n = 4$ and $\ell = 3$ (that is, trinucleotides over the genetic alphabet). Every 4-circular 3-letter code is circular, and there are 3- but not 4-circular 3-letter code over $\mathcal{B}$:

$$X(4,3) = \{AGC, ATT, CAA, CTG, GCC, GAT, TCA, TGG\}.$$

(7) Let $n = 4$ and $\ell = 4$ (that is, tetranucleotides over the genetic alphabet). Every 15-circular 4-letter code is circular, and there are 14- but not 15-circular 4-letter codes over $\mathcal{B}$:

$$X(4,4) = \{AAAC, ACAG, AGAT, ATCA, CACC, CCCG, CGCT,$$
$$CTGA, GAGC, GCGG, GGGT, GTTA, TATC, TCTG, TGAA\}.$$

5. Proof of the Sharpness-Theorem 4.4

In this section we prove the Sharpness-Theorem 4.4, *i.e.* we prove that $k(n,\ell)$ is the least integer such that every code $X \subseteq \Sigma^\ell$ that is $k(n,\ell)$-circular is circular. For all integers n and ℓ both at least 2, we shall construct a code that is $(k(n,\ell)-1)$-circular but not circular. In order to do so, we construct codes such that the graphs associated to them contain a unique cycle, of length $k(n,\ell)$ if $k(n,\ell)$ is odd and, as required, of length $2k(n,\ell)$ otherwise. Theorem 3.3 guarantees then that such a code is $(k(n,\ell)-1)$-circular but not circular. We break the demonstration into three cases: when ℓ is even (Lemma 5.1) and when ℓ is odd (Lemma 5.2). The case when ℓ is odd is again split into two cases: when ℓ is odd and $n = 2$ (Lemma I.1) and finally when ℓ is odd and $n \geq 3$ (Lemma I.2). The proofs of Lemmas I.1 and I.2 are quite long and are therefore contained in Appendix I. However, there are new mathematical tools for analysing codes, which may identify additional properties in the genetic code in the future. We indicate here the main idea of the constructions. We already would like to point out that, although the approaches to establish Lemmas I.1 and I.2 are similar, there are some differences. It seems therefore more natural to separate the binary case, which also helps to comprehend the strategy in a more restricted case. Given an integer $n \geq 2$, let $\Sigma_n = \{0, \ldots, n-1\}$ be the alphabet of cardinality n consisting of the first n integers. We first deal with the case when ℓ is even.

LEMMA 5.1. *If n and ℓ are integers both at least 2 and ℓ is even, then there is an ℓ-letter code over Σ that is $(k(n,\ell)-1)$-circular but not circular where $k(n,\ell) \coloneqq n^{\ell/2}$ if n is odd and $k(n,\ell) \coloneqq n^{\ell/2} - 1$ if n is even.*

PROOF. Note that $k(n,\ell)$ is always odd in this setting. The situation where $n = 2 = \ell$ is trivial: every code is 0-circular, and the code $\{00\}$ is not 1-circular.
We now assume that $(n,\ell) \neq (2,2)$. We construct an ℓ-letter code $X(n,\ell)$ over Σ_n such that $\mathcal{G}(X(n,\ell))$ contains a unique cycle, which is of odd length $k(n,\ell)$. It thus follows that $X(n,\ell)$ is $(k(n,\ell)-1)$-circular but not $k(n,\ell)$-circular by Theorem 3.3. The code is constructed by using n-adic representations of numbers below $n^{\ell/2}$; see Equation (5.1).
We first rule out a boundary case: if $n = 2$ and $\ell = 4$ then one verifies directly that $X(2,4) \coloneqq \{0001, 1100, 0111\}$ is 2-circular but not 3-circular. Indeed, the associated graph $\mathcal{G}(X(2,4))$, depicted in Figure 3, contains a unique directed cycle, which has length 3.
From now on, we assume that $(n,\ell) \neq (2,4)$, so that $n \geq 3$ or $\ell \geq 6$. Given a length $i \leq \ell/2$ and an integer $x < n^i$, we define $(x)_i$ to be the word of length i over the alphabet Σ_n representing x written in basis n. For example, if $n = 3$ then $(8)_4 = 0022$. If $w = (x)_i$ then we also write $x = [\![w]\!]$. To improve readability, we use Y and Z to respectively refer to $n-2$ and $n-1$ when writing words in Σ_n^*.
Let $X(n,\ell)$ be defined as

$$X(n,\ell) \coloneqq \left\{(x)_{\ell/2} \cdot (x+1)_{\ell/2} \ : \ x \in \mathbf{Z}_{k(n,\ell)}\right\}, \tag{5.1}$$

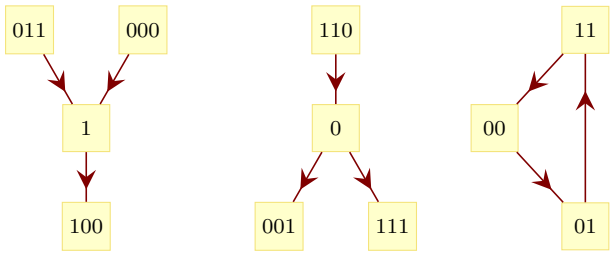

FIGURE 3. The graph $\mathcal{G}(X(2,4))$ associated to the binary 4-letter code $X(2,4) = \{0001, 1100, 0111\}$.

where the addition is in $\mathbf{Z}_{k(n,\ell)}$. Note that $|X| = k(n,\ell)$. For instance if $n = 2$ and $\ell = 6$, then $k(n,\ell) = n^{\ell/2} - 1 = 7$, and

$$X(2,6) = \{000001, 001010, 010011, 011100, 100101, 101110, 110000\}.$$

We assert that the graph $\mathcal{G}(X(n,\ell))$ associated to $X(n,\ell)$ has a unique cycle, of odd length $k(n,\ell)$. It then follows from Theorem 3.3 that the ℓ-letter code $X(n,\ell)$ is $(k(n,\ell)-1)$-circular but not $k(n,\ell)$-circular.

We begin by proving a useful assertion on the code $X^-(n,\ell)$ defined as

$$X^-(n,\ell) := \left\{(x)_{\ell/2} \cdot (x+1)_{\ell/2} \ : \ x \in \mathbf{Z}_{k(n,\ell)} \setminus \{k(n,\ell)-1\}\right\}.$$

(A). *The graph $G^- := \mathcal{G}(X^-(n,\ell))$ is acyclic. Moreover, for every $i \in \{1, \ldots, \ell/2-1\}$, every directed path of length 2 in G^- with a middle $(\ell-i)$-node begins at the i-node Z^i and ends at the i-node 0^i. In particular, all out-neighbours of 0^i in G^- have out-degree 0 and all in-neighbours of Z^i in G^- have in-degree 0.*

PROOF. For each $i \in \{1, \ldots, \ell/2\}$, let C_i be the i-component of G^-. Note that $C_{\ell/2}$ is a path of length $k(n,\ell)-1$, traversing all nodes $(x)_{\ell/2}$ in increasing order, for $x \in \mathbf{Z}_{k(n,\ell)}$. It follows that $C_{\ell/2}$ is acyclic. We now fix $i \in \{1, \ldots, \ell/2-1\}$ and focus our attention on the component C_i, which has no odd cycle. Let $y = y_1 \cdot y_2 \cdot y_3$ be an $(\ell-i)$-node of G, where $|y_1| = |y_3| = \ell/2 - i$ and $|y_2| = i$. Let us prove that if G^- contains two nodes x and x' such that $x \to y \to x'$, then $x = Z^i$ and $x' = 0^i$.

(1) By the definition of G^-, if there is an arc from a node x to y, then x is an i-node and it holds in $\mathbf{Z}_{k(n,\ell)}$ that
$$[\![y_2 \cdot y_3]\!] = [\![x \cdot y_1]\!] + 1.$$
There are two possible cases:
   (a) $y_2 = x$ and $[\![y_3]\!] = [\![y_1]\!] + 1$; or
   (b) $[\![y_2]\!] = [\![x]\!] + 1$, $y_1 = Z^{\ell/2-i}$ and $y_3 = 0^{\ell/2-i}$.
Note that $y_1 \neq y_3$ in both cases.

(2) On the other hand, if there is an arc from y to a node x', then x' is an i-node and it holds in $\mathbf{Z}_{k(n,\ell)}$ that
$$[\![y_3 \cdot x']\!] = [\![y_1 \cdot y_2]\!] + 1.$$
Again there are two possible cases:
   (a) $y_3 = y_1$ and $[\![x']\!] = [\![y_2]\!] + 1$; or
   (b) $[\![y_3]\!] = [\![y_1]\!] + 1$, $y_2 = Z^i$ and $x' = 0^i$.

Let us assume that y has both an ingoing arc $x \to y$ and an outgoing arc $y \to x'$ — where it might be that $x = x'$. Because of the arc $x \to y$, we know that $y_1 \neq y_3$, so in particular Case (2.a) is impossible,

hence we must fall into Case (2.b). Therefore, $x' = 0^i$ and $[\![y_3]\!] = [\![y_1]\!] + 1$, and we infer that the arc $x \to y$ fell into case (1.a), so $x = y_2 = Z^i$. This proves the "moreover" part of the assertion. It directly follows that 0^i does not belong to a directed cycle in C_i, and consequently no $(\ell - i)$-node belongs to a directed cycle in C_i. Because all arcs in C_i are between an i-node and an $(\ell - i)$-node, we deduce that C_i is acyclic, thereby concluding the proof of (A).

We now show that $\mathcal{G}(X)$ contains exactly one cycle, which spans its whole component $C_{\ell/2}$. The code X is obtained from X^- by adding the word $Z^{\ell/2-1}Y0^{\ell/2}$ if n is even, or the word $Z^{\ell/2}0^{\ell/2}$ if n is odd. Let us see how $\mathcal{G}(X)$ is obtained from G^-. As noted earlier, the $\ell/2$-component of G^- is a path of length $n^{\ell/2} - 1$ traversing all nodes $(x)_{\ell/2}$ for $x \in \mathbf{Z}_{k(n,\ell)}$ in increasing order. Because the $\ell/2$-component of $\mathcal{G}(X)$ is obtained from that of G^- by adding an arc from $(k(n,\ell)-1)_{\ell/2}$ to $(0)_{\ell/2}$, we precisely obtain a cycle of length $k(n,\ell)$. For each $i \in \{1, \ldots, \ell/2 - 1\}$, the component C_i is obtained from the i-component of G^- by adding an arc outgoing from the i-node Z^i and an arc ingoing to the i-node 0^i. This creates no cycle of length at least 4, since G^- contains no directed path from 0^i to Z^i. The only cycles that might be created are therefore of length 2, and there are two possible ones. The first possibility is to create a cycle containing precisely the i-node 0^i and either the $(\ell - i)$-node $Z^{\ell/2-1}Y0^{\ell/2-i}$ if n is even, or the $(\ell - i)$-node $Z^{\ell/2}0^{\ell/2-i}$ if n is odd. If this cycle is created, then

$$\begin{cases} [\![Z^{i-1}Y0^{\ell/2-i}]\!] = [\![0^iZ^{\ell/2-i}]\!] + 1 & \text{if } n \text{ is even,} \\ [\![Z^i0^{\ell/2-i}]\!] = [\![0^iZ^{\ell/2-i}]\!] + 1 & \text{if } n \text{ is odd.} \end{cases}$$

This is possible only if $i = 1$, and either $Y = 1$ and n is even, or $Z = 1$ and n is odd. However, the parity of n is the same as that of $Y = n - 2$, and different from that of $Z = n - 1$, so this first possible cycle is not created.
The other possible cycle is the one containing the i-node Z^i and either the $(\ell - i)$-node $Z^{\ell/2-i-1}Y0^{\ell/2}$ if n is even, or the $(\ell - i)$-node $Z^{\ell/2-i}0^{\ell/2}$ if n is odd. If this cycle is created, then

$$\begin{cases} [\![0^{\ell/2-i}Z^i]\!] = [\![Z^{\ell/2-i-1}Y0^i]\!] + 1 & \text{if } n \text{ is even,} \\ [\![0^{\ell/2-i}Z^i]\!] = [\![Z^{\ell/2-i}0^i]\!] + 1 & \text{if } n \text{ is odd.} \end{cases}$$

If n is odd then the equality implies that $i = \ell/2$, which is not the case. If n is even then the equality implies that $\ell/2 - i = 1$ and $Y = 0$, that is, $n = 2$. It would follow that $[\![01^i]\!] = [\![0^{i+1}]\!] + 1$, implying $i = 1$ and hence $\ell = 4$, which is not the case as we assumed that $(n,\ell) \neq (2,4)$.
This completes the proof of Lemma 5.1. □

We now proceed with the case where ℓ is odd. The full proof is split into two cases given in detail in Appendix I (Lemmas I.1 and I.2). Here we indicate the strategy and main steps of the proof.

LEMMA 5.2. *Let n and ℓ be integers both at least 2 and assume that ℓ is odd. There is an ℓ-letter code over Σ_n that is $(n^{(\ell-1)/2} - 1)$-circular but not circular.*

SKETCH OF PROOF. Let us first deal with the case where ℓ is odd and $n = 2$. We have to show that there is an ℓ-letter code over Σ_n that is $(2^{(\ell-1)/2} - 1)$-circular but not circular. Let us fix $k \coloneqq k(2,\ell) = 2^{\frac{\ell-1}{2}}$. The aim is to construct a binary code X such that its graph $\mathcal{G}(X)$ contains a unique cycle, of length $2k$. Let $s \coloneqq \frac{\ell-3}{2}$, $S \subseteq \{0,1\}^s$ be a subset of binary words of length s and set

$$X_S \coloneqq \{a \cdot y \cdot a \cdot y \cdot a \ : \ a \in \{0,1\}, y \in S\} \cup \{a \cdot y \cdot \overline{a} \cdot y \cdot a \ : \ a \in \{0,1\}, y \in \{0,1\}^s \setminus S\}$$

where for a binary word w, the *complement* of w is the word $\overline{w}$ obtained from w by complementing each of its letters. It will be shown in Lemma I.1 (Appendix I) that all but one components of $\mathcal{G}(X)$ are

acyclic if $S \subseteq 01\{0,1\}^{s-2}$ and that there is a choice for such a set S so that the remaining component consists of exactly one cycle, which solves the binary case.
The case where ℓ is odd and $n > 2$ is more delicate. As before, one sets $s := \frac{\ell-3}{2}$ and $k := n^{\frac{\ell-1}{2}}$. We shall define in Lemma I.2 (Appendix I) a mapping φ from Σ_n^s to the family of 3-letter codes over Σ_n with very specific properties so that the associated code

$$X_\varphi := \{a \cdot y \cdot b \cdot y \cdot c \ : \ y \in \Sigma_n^s \text{ and } abc \in \varphi(y)\}$$

satisfies our requirements. All the details for the construction of φ can be found in Lemma I.2 but we would like to notice that in the binary case, we have $\varphi(y) = \{010, 101\}$ if $y \notin S$ and $\varphi(y) = \{000, 111\}$ otherwise. □

Lemmas 5.1 and 5.2 together end the proof of our Sharpness-Theorem 4.4.

6. Biological consequences

Almost all living organisms use the same standard genetic code (SGC) to translate 64 trinucleotides (codons) into 20 amino acids and the stop signal. Many hypotheses have been proposed to explain the origin of the genetic code (*e.g.* reviewed in [**19**, **20**]), including the frozen accident theory stating that the genetic code was created randomly and stayed frozen ever since, the stereochemical theory based on stereochemical relationships between amino acids and specific codons [**35**, **40**], the adaptive theory suggesting that the genetic code was shaped to be maximally robust [**16**, **38**], and the coevolution theory of the genetic code with amino acid biosynthetic pathways [**39**]. However, it is likely that all these models combined to play a part in the evolution of the genetic code.
The adaptive theory proposes that the genetic code was optimized to minimize the effects of errors during transcription and translation. The most common source of errors, known as missense errors, is the incorrect reading of a codon and the resulting incorporation of the wrong amino acid. From a theoretical point of view, we will show below the existence of a hierarchy of circular codes and k-circular codes allowing a codon decoding without ambiguity, from strong to flexible reading frame constraints. Of course, this model does not exclude the possibility of evolutionary phases with reading frame errors, such as the ribosomal frameshifts that are observed, for example, in the genes of today's viruses (to cite a topical example, the -1 ribosomal frameshift in the gene *orf1ab* of SARS-CoV-2, NCBI identification MT072688).
Circular codes could have operated in the primitive soup for constructing the modern standard genetic code. Figure 4 proposes an evolutionary hypothesis of the genetic code based on a growing combinatorial hierarchy of k-circular trinucleotide codes where $k \in \{1, 2, 3, 4\}$ (see Theorem 4.1 with $n = 4$ and $\ell = 3$, and Corollary 4.2). Evolution would have started with the circular (4-circular) trinucleotide codes X_p with an increasing complexity according to the maximal path length p (from 1 to 8) in their associated graph $\mathcal{G}(X_p)$. As the maximal path length p is related to the window nucleotide length of reading frame retrieval, the circular codes X_1 (strong comma-free) and X_2 (comma-free) are more constrained than those in X_8. The maximal C^3-self-complementary trinucleotide circular code X observed in genes (1.1) belongs to the class X_8. Then evolution continued with the three classes of k-circular codes, where $k \in \{1, 2, 3\}$, which are less constrained than the classes of circular codes. As such, these three classes with a partial circularity property could represent an intermediate step in the code evolution between circular codes which have a complete circularity property, and the extant genetic code SGC where the circularity property is totally lost (Figure 4). As a consequence, the circular code motifs, in particular the X-motifs of the circular code X (1.1) which are significantly enriched in the genes of

most organisms [**5**, **28**], always retrieve the reading frame. In contrast, the k-circular code motifs may not always find out the reading frame.

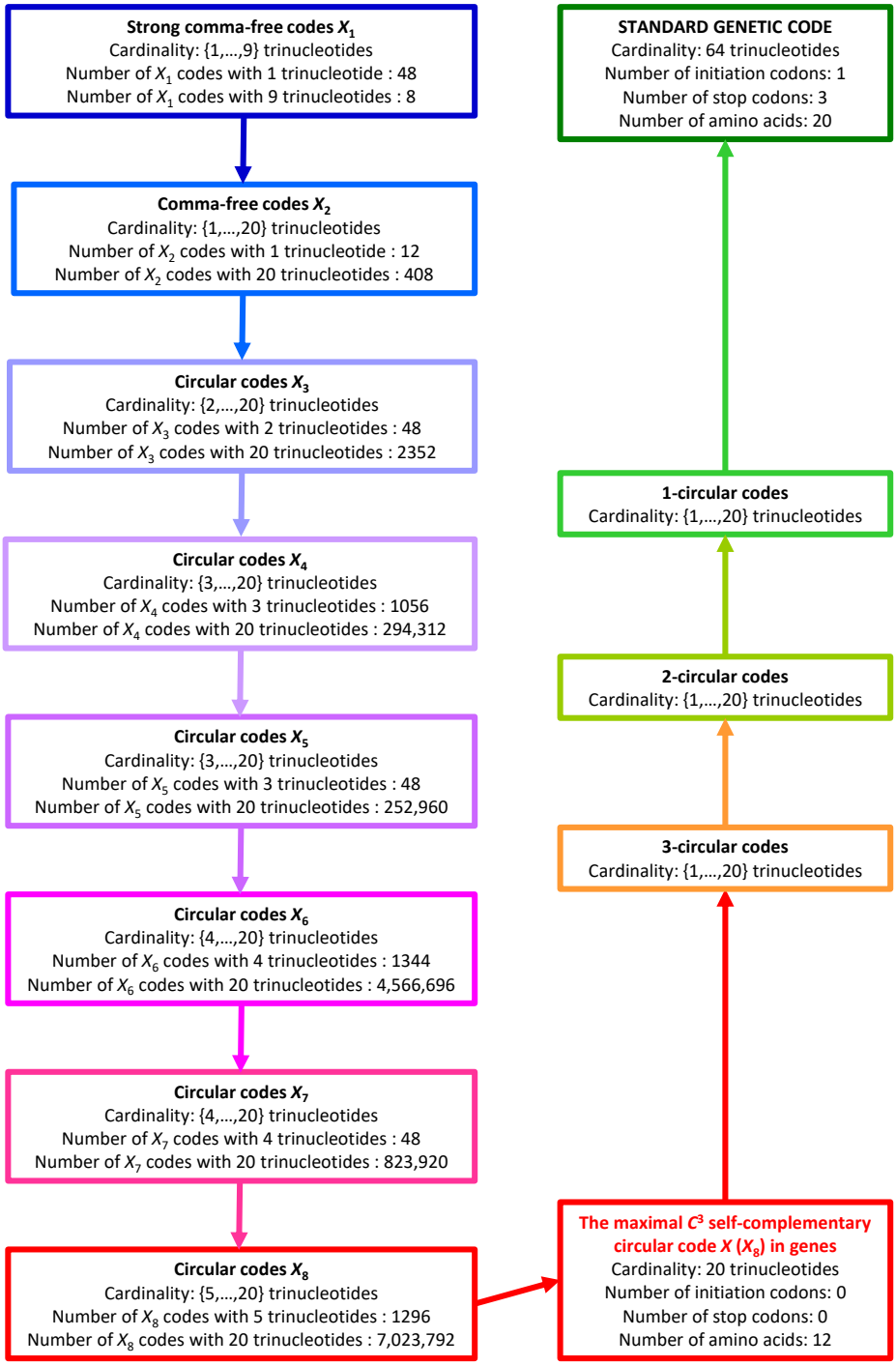

FIGURE 4. A combinatorial hierarchy of circular (4-circular) trinucleotide codes X_p, where p is the maximal path length associated with the graph $\mathcal{G}(X_p)$, and k-circular trinucleotide codes, where $k \in \{1, 2, 3\}$ (see Theorem 4.1 with $n = 4$ and $\ell = 3$, and Corollary 4.2).

This code evolution is also supported by the number of amino acids which are coded by the circular codes. There are 12,964,440 maximal circular codes. No maximal circular code among these 12,964,440 ones codes for 20 or 19 amino acids with SGC. Ten maximal circular codes code for 18 amino acids with SGC (see [**30**, Introduction]). Interestingly, we identify 52 maximal 1-circular codes among $3^{20} = 3,486,784,401$ ones, *i.e.* with a very low probability $\approx 10^{-8}$, which code for the 20 amino acids (see the list given in Appendix II or [**25**, Table 2] where they were called bijective genetic codes without permuted trinucleotides $WPTBGC$ before the theory of k-circular codes was developed in this work). It has already been verified [**13**] that these 52 maximal 1-circular codes cannot be 2-circular*. Moreover, the following construction based on graph theory underlines once more the distinguished role of this class of 52 maximal 1-circular codes and gives a theoretical framework for earlier computer results [**25**].

A *bipartite graph* is a graph $\mathcal{G} = (V, E)$ such that the set of vertices V is the disjoint union of two sets A and B such that any edge in $e \in E$ is of the form $e = (a, b)$ for some $a \in A$ and $b \in B$, *i.e.* no edge connects two vertices both from A or two vertices both from B. A *perfect matching* of $\mathcal{G}$ is a subset $M \subseteq E$ such that the edges of M form a bijection between the sets A and B. A perfect matching can thus exist only if A and B have the same cardinality. We are now ready to prove the existence of the 52 maximal 1-circular codes that encode all 20 amino acids.

LEMMA 6.1. *There are exactly* 52 *maximal* 1*-circular codes that encode all* 20 *amino acids.*

PROOF. We first need to recall some facts [**13**]. An equivalence class $[c]$ of some codon $c \in \mathcal{B}^3$ consists of c and its circular shifts, *e.g.* $[ATC] = \{ATC, TCA, CAT\}$. The equivalence class is called *complete* if it contains three elements, *i.e.* if $c \notin \{AAA, CCC, GGG, TTT\}$. It was shown [**13**] that there are 20 complete equivalence classes $D_1, \ldots, D_{20}$ and that each complete equivalence class encodes three different amino acids or two amino acids and the stop signal. Moreover, it was also shown [**13**] that each maximal 1-circular code encoding all 20 amino acids must contain the following 7 codons:

$$TGG, ATG, TTC, AAG, GAG, GAC, GGC,$$

which respectively encode the following 7 amino acids:

$$\text{Trp, Met, Phe, Lys, Glu, Asp, Gly,}$$

and belong to the complete equivalence classes $D_2, D_8, D_{11}, D_{15}, D_{18}, D_{19}$.

The idea of the proof is to construct the following bipartite graph $\mathcal{G} = (V, E)$, displayed in Figure 5, where V is the union of the two disjoint sets

$$D = \{D_1, D_3, D_4, D_6, D_7, D_9, D_{10}, D_{12}, D_{13}, D_{14}, D_{16}, D_{17}, D_{20}\}$$

and

$$A = \{\text{Val, Tyr, Thr, Ser, Pro, Leu, Ile, His, Glu, Cys, Asp, Arg, Ala}\}.$$

Moreover, the set of edges E consists of all pairs (D_i, aa) such that there is a codon in D_i that encodes the amino acid aa. It is now easy to see that the maximal 1-circular codes that encode all 20 amino acids are in bijective correspondence with the perfect matchings of the constructed graph $\mathcal{G}$. An application of established algorithms for calculating perfect matchings of a graph, *e.g.* the Hungarian algorithm, now yields the list of 52 perfect matchings of $\mathcal{G}$. This completes the proof. □

During this work, a new relation came to light between the combinatorial hierarchy of circular codes (Figure 4) and their probability measure of reading frame coding RFR (or reading frame retrieval)

*The first and last author would like to remark that it was incorrectly claimed [**13**] that there are only 2 maximal 1-circular codes that code for all 20 amino acids.

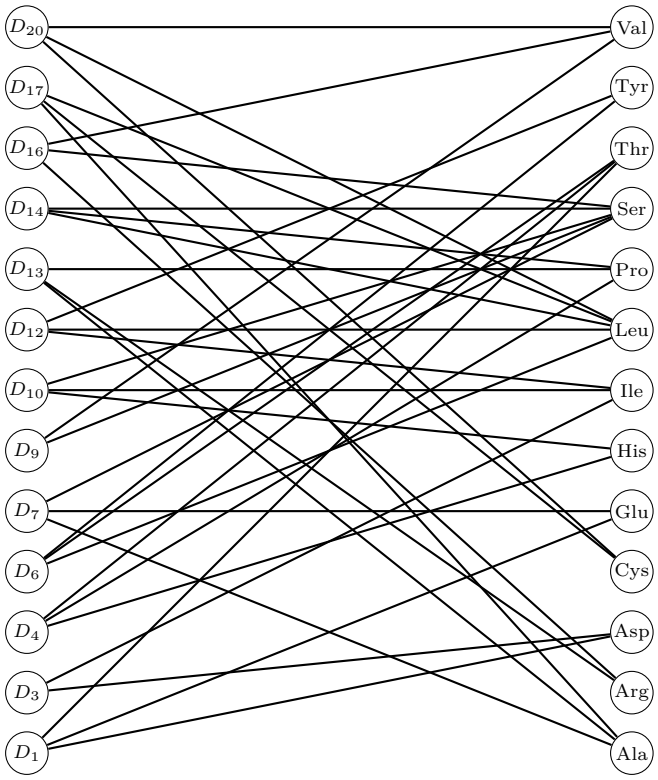

FIGURE 5. The bipartite graph $\mathcal{G} = (V, E)$ where V is composed of the 13 equivalence classes in $D = \{D_1, D_3, D_4, D_6, D_7, D_9, D_{10}, D_{12}, D_{13}, D_{14}, D_{16}, D_{17}, D_{20}\}$ and the 13 amino acids in $A = \{$Val, Tyr, Thr, Ser, Pro, Leu, Ile, His, Glu, Cys, Asp, Arg, Ala$\}$, and E is the set of all pairs (D_i, aa) such that there is a codon in D_i that encodes the amino acid aa.

within two successive codons (Figure 6; see the method in [**25**] for detail). This RFR measure ranges from 1/3 (one chance out of three to retrieve the reading frame among the three possible frames in genes) with the random codes, *e.g.* B^3, to 1 (the reading frame is always retrieved) with the strong comma-free codes and the comma-free codes. Remember that: (i) the maximal size of strong comma-free codes cannot exceed 9 trinucleotides, and there are only 8 codes belonging to this class; and (ii) there are 408 maximal (size of 20 trinucleotides) comma-free codes. The $12,964,440$ maximal circular codes have RFR values in the range $[72.7, 100]$ (%) including the 216 maximal C^3 self-complementary circular codes in the range $[77.2, 90.1]$ (%). The maximal C^3 self-complementary circular codes X identified in genes (1.1) has a RFC value equal to 81.3 (%). Interestingly, the identified 52 maximal 1-circular codes have RFR values in the range $[62.2, 71.6]$ (%). Thus, they have an ability to retrieve the reading frame weaker than the maximal circular code of the lowest RFR value [72.7] (%). On the other hand, the genetic code $\mathcal{B}^3$, obviously not circular, has a RFR probability of course equal to 1/3, as with all random codes (depicted in Figure 6). The 4 unitary codes $\{AAA\}$, $\{CCC\}$, $\{GGG\}$ and $\{TTT\}$, which are not circular, are also random codes with a RFR probability equal to 1/3. It is very interesting to point out from an amino acid coding point of view that the loss of the reading frame in a sequence of such a unitary code does not lead to the coding of an amino acid different from the one coded in the reading frame, in contrast to the 60 remaining unitary codes which are circular and comma-free (48 strong comma-free). In summary, the growing combinatorial hierarchy

of k-circular trinucleotide codes is associated with a decreasing probability hierarchy of reading frame coding RFR.

The main property of circular codes which has been reported since 1996, is the nucleotide window length for retrieving the reading frame, *e.g.* 13 nucleotides with the circular code X in genes. The relation identified above shows that, from our point of view, the circular codes may retrieve the reading frame in genes according to two properties (property (i) being classic in coding theory, property (ii) being a new proposition): (i) always retrieved, *i.e.* without error, using a nucleotide window length, but then with a slow process; and (ii) retrieved with high frequencies, *i.e.* not always as some errors may occur, within two successive codons, thus with a fast process.

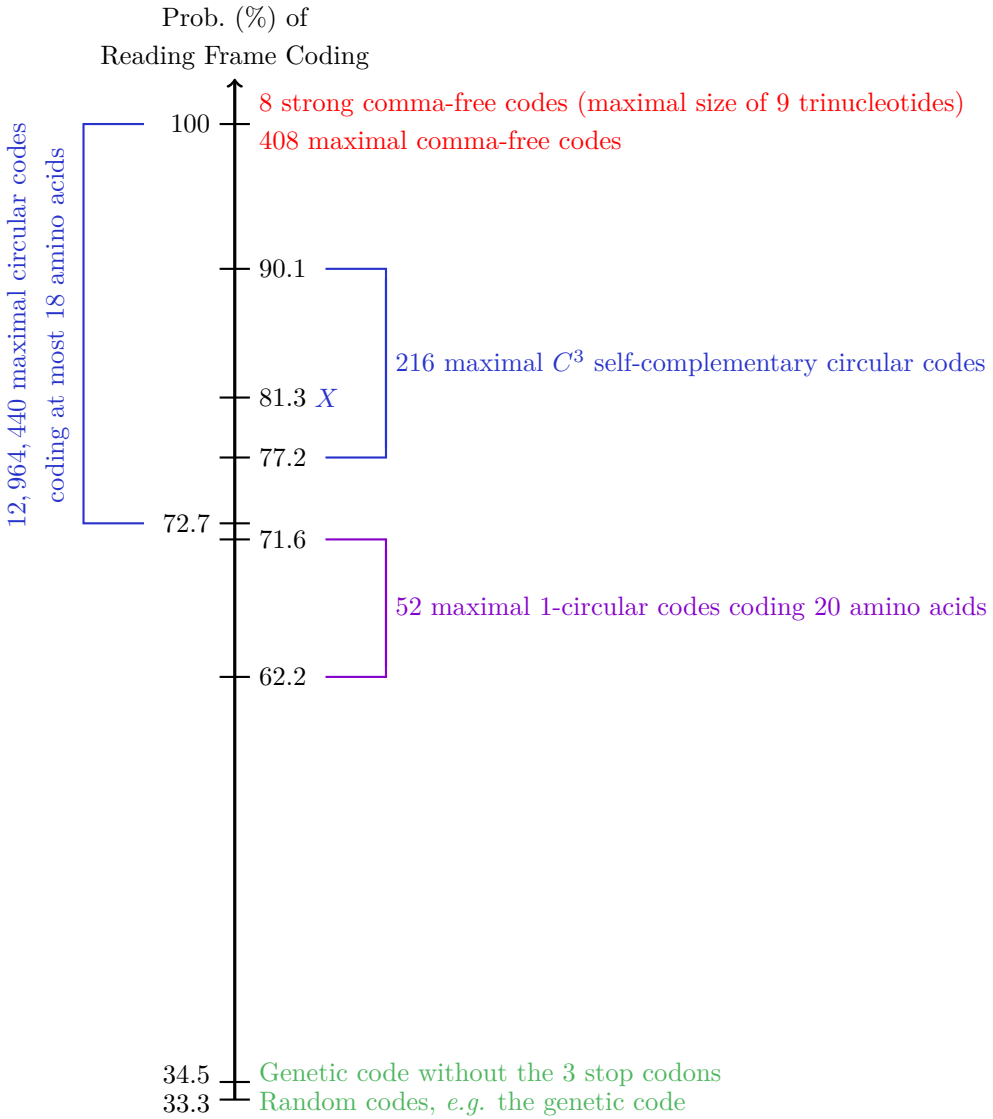

FIGURE 6. A probability hierarchy of reading frame coding within two successive codons with the circular (4-circular) trinucleotide codes and the 52 maximal 1-circular trinucleotide codes coding the 20 amino acids (updated from [**25**, Figure 1]).

7. Conclusion

In the present work we found for all possible sizes n of a given finite alphabet Σ and for all word lengths ℓ, the (sharp) boundary $k(n,\ell)$ from which the $k(n,\ell)$-circularity of an ℓ-letter code X over Σ implies its circularity. This result is important: obviously from a mathematical point of view, and also from a biological perspective.

The theoretical work developed here opens several biological research themes which can be investigated in the future: a classification of genes according to circular and k-circular code motifs as functionally conserved ("ancestral") genes may contain more circular code motifs compared to functionally specific genes; a construction of phylogenetic trees and alignments using these classes of motifs; a localization of circular and k-circular code motifs in a gene as the circular code motifs may be associated with regions in a gene where the ribosomal translation accuracy is important, *e.g.* after or before a splice site in the eukaryotic genes; the amino acid coding may also differ between these different circular code motifs; and many more.

Appendix I. Proof of the Sharpness-Theorem 4.4 in the case where ℓ is odd

In order to complete the proof of Theorem 4.4, we need to give a detailed proof of Lemma 5.2 where only the main construction steps were indicated. Recall that, given an integer $n \geq 2$, we have $\Sigma_n = \{0, \ldots, n-1\}$.

I.1. Case where ℓ is odd and $n = 2$.

LEMMA I.1. *Let ℓ be an odd integer greater than 2. There is an ℓ-letter code over Σ_2 that is $(2^{(\ell-1)/2} - 1)$-circular but not circular.*

PROOF. Set $k \coloneqq k(2,\ell) = 2^{\frac{\ell-1}{2}}$. As mentioned in Lemma 5.2 we show that there exists a binary code X that is $(k-1)$-circular but not circular, and more specifically that there exists a code X such that $\mathcal{G}(X)$ contains a unique cycle, of length $2k$. For $a \in \{0,1\}$, we set $\overline{a} \coloneqq 1 - a$ and we call $\overline{a}$ the *complement* of a. If w is a binary word, the *complement* of w is the word $\overline{w}$ obtained from w by complementing each of its letters.

Let $s \coloneqq \frac{\ell-3}{2}$, and let $S \subseteq \{0,1\}^s$ be a given subset of binary words of length s. We construct the code X_S associated to S as mentioned in the sketch of the proof of Lemma 5.2 given earlier:

$$X_S \coloneqq \{a \cdot y \cdot a \cdot y \cdot a \ : \ a \in \{0,1\}, y \in S\} \cup \{a \cdot y \cdot \overline{a} \cdot y \cdot a \ : \ a \in \{0,1\}, y \in \{0,1\}^s \setminus S\}\,.$$

We observe that if $\ell \in \{3,5\}$ (and hence $s \in \{0,1\}$) then the code $X_\varnothing$ is indeed $(k-1)$-circular but not k-circular, for instance by applying Theorem 3.3.

In the sequel we prove that if $S \subseteq 01\{0,1\}^{s-2}$, then all but one components of $\mathcal{G}(X_S)$ are acyclic and that there is a choice of S such that the remaining component consists of one cycle only. This then yields the desired result. We thus assume from now on that $\ell \geq 7$, and hence $s \geq 2$.

Given a set S, the $\lfloor \ell/2 \rfloor$-component of $\mathcal{G}(X_S)$ is called the *middle component*.

(B). *The middle component of the graph $\mathcal{G}(X_\varnothing)$ is a union of cycles, while for each $i \in \{1, \ldots, \lfloor \ell/2 \rfloor - 1\}$, the $(\ell - i)$-nodes are either sources or sinks, and hence the i-component C_i of $\mathcal{G}(X_\varnothing)$ is acyclic and contains only paths of length at most 2.*

PROOF. Recall that $\lfloor \ell/2 \rfloor = s + 1$. In the middle component C_{s+1} of $\mathcal{G}(X_\varnothing)$, every possible $(s+1)$-node $a \cdot y \cdot a'$ with $a, a' \in \{0,1\}$ and $y \in \{0,1\}^{s-1}$ is present, and has exactly one in-going arc, which comes from the $(s+2)$-node $a' \cdot a \cdot y \cdot \overline{a'}$, and one out-going arc, which goes to the $(s+2)$-node $\overline{a} \cdot y \cdot a' \cdot a$. On the other hand, every $(s+2)$-node of the form $a \cdot y \cdot \overline{a}$ (so with complementary

first and last letters and where $|y| = s$) is present, and has exactly one in-going arc, which comes from the $(s+1)$-node $\overline{a} \cdot y$, and one out-going arc, which goes to the $(s+1)$-node $y \cdot a$. So the middle component is a 2-regular digraph (that is, 1-out-regular and 1-in-regular), and hence a union of cycles of total length $2 \times 2^{s+1} = 2k$.

Let $i \in \{1, \ldots, s\}$. Suppose, contrary to the statement, that v is an $(\ell - i)$-node in C_i that is neither a source nor sink. It follows that C_i contains an arc $e = u \to v$ and an arc $e' = v \to u'$, where u and u' are i-nodes. The arc e corresponds to a word $a \cdot y \cdot \overline{a} \cdot y \cdot a$ with $|y| = s$. Let us write $y = y_1 \cdot y_2$ with $|y_1| = i - 1$, so that $u = a \cdot y_1$ and $v = y_2 \cdot \overline{a} \cdot y_1 \cdot y_2 \cdot a$. Similarly, the arc e' corresponds to a word $a' \cdot y' \cdot \overline{a'} \cdot y' \cdot a'$ with $|y'| = s$. Writing $y' = y'_1 \cdot y'_2$ with $|y'_2| = i - 1$, we obtain $u' = y'_2 \cdot a'$ and $v = a' \cdot y'_1 \cdot y'_2 \cdot \overline{a'} \cdot y'_1$. Consequently

$$y_2 \cdot \overline{a} \cdot y_1 \cdot y_2 \cdot a = a' \cdot y'_1 \cdot y'_2 \cdot \overline{a'} \cdot y'_1,$$

so in particular

$$y_2 \cdot \overline{a} = a' \cdot y'_1 \quad \text{and} \quad y_2 \cdot a = \overline{a'} \cdot y'_1.$$

This yields the double contradiction that the first letter of y_2 should be equal both to a' and to $\overline{a'}$, and that the last letter of y'_1 should be equal both to a and to $\overline{a}$. This ends the proof of (B).

(C). *If $S \subseteq 01\{0,1\}^{s-2}$ then all components of $\mathcal{G}(X_S)$ but the middle one are acyclic.*

Proof. Let $S \subseteq 01\{0,1\}^{s-2}$ and consider the component C_i of $\mathcal{G}(X_S)$, where $i \in \{1, \ldots, s\}$. Suppose that C_i contains a path $u \to v \to u'$ of length 2 where u and u' are i-nodes (possibly equal), and hence v is an $(\ell - i)$-node. By (B), at least one of these two arcs comes from a word of the form $a \cdot y \cdot a \cdot y \cdot a$ with $y \in S$ and $a \in \{0,1\}$. Proceeding exactly as in the proof of (B), we infer that, actually, both arcs come from such words. So let $(y, y') \in S^2$ such that $u \to v$ corresponds to the word $a \cdot y \cdot a \cdot y \cdot a$ and $v \to u'$ corresponds to the word $a' \cdot y' \cdot a' \cdot y' \cdot a'$. Write $y = y_1 \cdot y_2$ and $y' = y'_1 \cdot y'_2$ with $|y_1| = i - 1 = |y'_2|$. It follows that $u = a \cdot y_1$, $u' = y'_2 \cdot a'$ and

$$y_2 \cdot a \cdot y_1 \cdot y_2 \cdot a = v = a' \cdot y'_1 \cdot y'_2 \cdot a' \cdot y'_1. \tag{I.1}$$

In particular, $y_1 = y'_2$ and hence

$$u' = y_1 \cdot a'. \tag{I.2}$$

We next observe that $|y_1| > 0$, that is, $i \geq 2$. Indeed if $i = 1$, then $y'_1, y_2 \in S$. Since $S \subseteq 01\{0,1\}^{s-2}$, we know that $y_2 \cdot a \neq a' \cdot y'_1$ (they differ on the second letter), which contradicts (I.1).

We now prove that $u' \neq 0^i$. Suppose first that $i \geq 3$. Then $|y_1| \geq 2$ so y_1 starts with 01 as $y_1 \cdot y_2 \in S$. Therefore u' starts with 01. Suppose now that $i = 2$. Then $y_1 = 0$ and y_2 starts with 1. Since $y_2 \cdot a = a' \cdot y'_1$, it follows that $a' = 1$ and hence $u' \neq 0^i$.

Suppose now, for a contradiction, that C_i contains a cycle $\mathcal{C}$. Every i-node on $\mathcal{C}$ is the last vertex of a path of length 2 (possibly closed) contained in $\mathcal{C}$. Therefore, all i-nodes on $\mathcal{C}$ contain a 1. It follows from this that $\mathcal{C}$ contains an i-node that starts with 1. Indeed, let $u = 0^j 1 \cdot y$ be an i node on $\mathcal{C}$, with $j \geq 1$ and $|y| = i - j - 1$. Then by (I.2) the i-node u' that is at directed distance 2 from u on $\mathcal{C}$ starts with $0^{j-1}1$. The assertion follows by (finite) induction. However, every i-node on $\mathcal{C}$ starts with the prefix of an element of S, and therefore cannot start with a 1. This contradiction concludes the proof of (C).

(D). *There exists $S \subseteq 01\{0,1\}^{s-2}$ such that the middle component of $\mathcal{G}(X_S)$ is connected.*

We need the following property to establish (D).

(D1). *Assume that there exist $S \subseteq \{0,1\}^s$ and $y \in \{0,1\}^s \setminus S$ such that the middle component of $\mathcal{G}(X_S)$ contains two different cycles $\mathcal{C}_0$ and $\mathcal{C}_1$ such that y is a prefix of an $(s+1)$-node in $\mathcal{C}_i$ for each $i \in \{0,1\}$. Then the middle component of $\mathcal{G}(X_{S+y})$ contains one cycle fewer than that of $\mathcal{G}(X_S)$.*

Proof. Note that $\mathcal{C}_0$ and $\mathcal{C}_1$ are necessarily disjoint. Without loss of generality, assume that $y \cdot a$ is an $(s+1)$-node on $\mathcal{C}_a$ for $a \in \{0,1\}$. Because $y \notin S$, it follows from the definition of X_S that for each $a \in \{0,1\}$, the cycle $\mathcal{C}_a$ contains the path

$$\overline{a} \cdot y \to a \cdot y \cdot \overline{a} \to y \cdot a.$$

By the definition of X_{S+y}, the only change between the middle components of $\mathcal{G}(X_S)$ and $\mathcal{G}(X_{S+y})$ consists in the replacement of these two paths by

$$a \cdot y \to a \cdot y \cdot a \to y \cdot a$$

for each $a \in \{0,1\}$, see Figure 7. This merges $\mathcal{C}_0$ and $\mathcal{C}_1$ into a single cycle containing all i-nodes in $\mathcal{C}_0$ and $\mathcal{C}_1$, and leaves the other cycles in the middle component unchanged. The conclusion of (D1) follows.

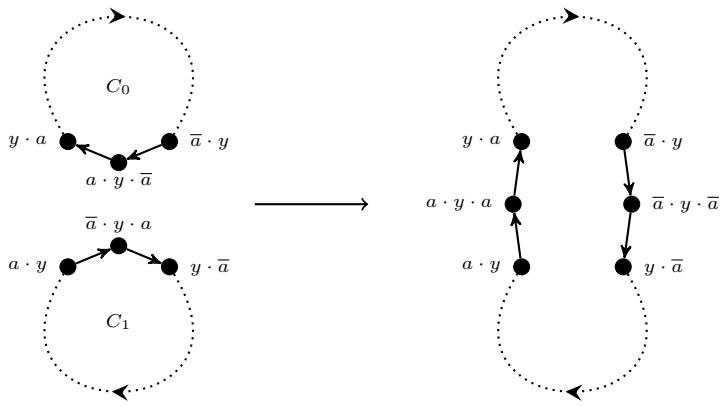

Figure 7. Schematic view of the merging of two cycles in the proof of (D1).

We analyse the following algorithm to finish the proof of (D). Given a set S, let Y_S be the set of vertices $y \in \{0,1\}^s \setminus S$ satisfying the hypothesis of (D1). For $w \in \{0,1\}^{s+1}$, let $\mathcal{C}_S(w)$ be the set of all the $(s+1)$-nodes in the same connected component of $\mathcal{G}(X_S)$ as w. In particular, this component is a directed cycle of even length, alternating between $(s+1)$-nodes and $(s+2)$-nodes. If w_1 and w_2 belong to $\mathcal{C}_S(w)$, then w_2 is *consecutive to* w_1 if the (directed) distance from w_1 to w_2 on this directed cycle is 2. The algorithm is as follows.
We start by setting $S \coloneqq \varnothing$. Then, while there exists y in $Y_S \cap 01\{0,1\}^{s-2}$ that is the prefix of a word in $\mathcal{C}_S(0^{s+1})$, we add y to S.
Note that the algorithm terminates. From now on, we let S be the set returned by this algorithm.

(D2). *Set $\mathcal{C}_0 \coloneqq \mathcal{C}_S(0^{s+1})$. Assume that each $y \in 01\{0,1\}^{s-2}$ appears either twice or never as the prefix of an $(s+1)$-node in $\mathcal{C}_0$. Then $\mathcal{C}_0$ contains all $(s+1)$-nodes, and hence has length $2k$.*

Proof. For an $(s+1)$-node $w = a_1 \cdots a_{s+1} \in \{0,1\}^{s+1}$ and an integer $j \in \{0, \ldots, s+1\}$, we define the *complement j-shift* of w to be the $(s+1)$-node $a_{j+1} \cdots a_{s+1} \overline{a_1} \cdots \overline{a_j}$. So w is its own complement

0-shift and the complement $\overline{w}$ of w is the complement $(s+1)$-shift of w. We notice that if w is an $(s+1)$-node, then the node consecutive to w on $\mathcal{C}_\varnothing(w)$ is the complement 1-shift of w. It thus follows (by finite induction) that the complement j-shift of w also belongs to $\mathcal{C}_\varnothing(w)$ for each $j \in \{0, \dots, s+1\}$. In particular, $\mathcal{C}_\varnothing(w)$ is stable under taking complements, and contains a word starting with 01 for every $w \in \{0,1\}^{s+1}$.

Another useful remark is the one made at the end of the proof of (D1), that $\mathcal{C}_{\mathrm{S}}(x) \subseteq \mathcal{C}_{S+y}(x)$ for every $x \in \{0,1\}^{s+1}$ and $y \in Y_S$. Consequently,

$$\forall x, x' \in \{0,1\}^{s+1}, \quad \mathcal{C}_{\mathrm{S}}(x) = \mathcal{C}_{\mathrm{S}}(x') \Rightarrow \mathcal{C}_{S+y}(x) = \mathcal{C}_{S+y}(x'). \tag{I.3}$$

Since $\mathcal{C}_\varnothing(w)$ is stable under taking complements for every $w \in \{0,1\}^{s+1}$, we deduce from (I.3) that so is $\mathcal{C}_0$. Therefore if y satisfies the hypothesis of (D2), then so does $\overline{y}$, and hence all words in $10\{0,1\}^{s-2}$ satisfy the hypothesis of (D2).

Let $w \coloneqq (01)^{(s+1)/2}$ if s is odd and $w \coloneqq 1(01)^{s/2}$ otherwise. Our next goal is to demonstrate that $\mathcal{C}_0$ contains w. From this we shall deduce that $\mathcal{C}_0$ contains all $(s+1)$-nodes starting with 01, and as a consequence all $(s+1)$-words. We prove by induction on $j \in \{0, \dots, s-1\}$ that $\mathcal{C}_0$ contains the word w_j, defined to be $(01)^{j/2+1}1^{s-j-1}$ if j is even and $1(01)^{(j+1)/2}1^{s-j-1}$ otherwise. First, $w_0 \in \mathcal{C}_0$ by (I.3) since $w_0 \in \mathcal{C}_\varnothing(0^{s+1})$. Suppose now that $j \geq 1$ and $w_{j-1} \in \mathcal{C}_0$. Write $w_{j-1} = w'_{j-1} \cdot 1$. Because w_{j-1} is the complement 1-shift of $0 \cdot w'_{j-1}$, and $\mathcal{C}_0$ is stable under complement shifts, we know that $0 \cdot w'_{j-1} \in \mathcal{C}_0$. (Indeed, the complement 1-shift of any $(s+1)$-word W is the complement s-shift of the complement of W.) Since w'_{j-1} starts with either 01 or 10, we know by assumption that $w'_{j-1} \cdot 0$ must belong to $\mathcal{C}_0$. Consequently, the stability of $\mathcal{C}_0$ under taking complement 1-shifts implies that $1 \cdot w'_{j-1} \in \mathcal{C}_0$. Since $w_j \in \{0 \cdot w'_{j-1}, 1 \cdot w'_{j-1}\}$, it follows that $w_j \in \mathcal{C}_0$, which finishes the induction. We conclude that w belongs to $\mathcal{C}_0$ since $w = w_{s-1}$.

A similar argument now allows us to show that $\mathcal{C}_0$ contains all $(s+1)$-nodes that start with 01. Indeed, let $x \in \{0,1\}^{s-1}$. We want to show that $01 \cdot x \in \mathcal{C}_0$. For every $j \in \{1, \dots, s-1\}$, let x_j be the factor of length $s+1$ of $w \cdot x$ that starts at the j-th letter. We prove by induction on $j \in \{1, \dots, s-1\}$ that x_j belongs to $\mathcal{C}_0$. We have seen that $x_1 = w$ belongs to $\mathcal{C}_0$. Suppose that $j \in \{2, \dots, s-1\}$ and x_{j-1} belongs to $\mathcal{C}_0$. Writing $x_{j-1} = a \cdot x'_{j-1}$ with $a \in \{0,1\}$, the $(s+1)$-node consecutive to x_{j-1} on $\mathcal{C}_0$ is either $x'_{j-1} \cdot 0$ or $x'_{j-1} \cdot 1$. Since x'_{j-1} starts with either 01 or 10, we know by assumption that both $x'_{j-1} \cdot 0$ and $x'_{j-1} \cdot 1$ belong to $\mathcal{C}_0$. Since x_j is one of these, this finishes the induction. It follows that $01 \cdot x = x_{s-1} \in \mathcal{C}_0$, and hence $\mathcal{C}_0$ indeed contains all words in $01\{0,1\}^{s-1}$.

We are now ready to prove that $\mathcal{C}_0$ contains all $(s+1)$-nodes. Indeed, if w is an $(s+1)$-node, then (I.3) implies that $\mathcal{C}_{\mathrm{S}}(w)$ contains an $(s+1)$-node starting with 01 since $\mathcal{C}_\varnothing(w)$ does, and hence $\mathcal{C}_{\mathrm{S}}(w) = \mathcal{C}_0$. This concludes the proof of (D2).

It follows from (D2) that the set S constructed by the algorithm satisfies the conclusion of (D). It follows from (C) and (D) that $\mathcal{G}(X_S)$ contains a unique cycle, of length $2k$. This concludes the proof of Lemma I.1. □

Here are the possible sets S provided by the algorithm in the proof of Lemma I.1 when $3 \leq \ell \leq 15$:

- $S = \varnothing$ if $\ell \in \{3, 5\}$;
- $S = \{01\}$ if $\ell = 7$;
- $S = \{011\}$ if $\ell = 9$;
- $S = \{0100, 0101, 0111\}$ if $\ell = 11$;
- $S = \{01000, 01011, 01100, 01111, x\}$ where $x \in \{01001, 01110\}$ if $\ell = 13$; and
- $S = \{010000, 010001, 010101, 010111, 011011, 011100, 011110, 011111, x\}$ if $\ell = 15$, with $x \in \{011010, 010110\}$.

I.2. Case where ℓ is odd and $n \geq 3$.

LEMMA I.2. *If n and ℓ are integers greater than 2 and ℓ is odd, then there is an ℓ-letter code over Σ_n that is $(n^{(\ell-1)/2}-1)$-circular but not circular.*

PROOF. As before, we set $s \coloneqq (\ell-3)/2$ and $k \coloneqq k(n,l) = n^{(\ell-1)/2}$. As mentioned in the proof of Lemma 5.2 we shall define a mapping φ from Σ_n^s to the family of 3-letter codes over Σ_n that has some special desired properties. We set

$$X_\varphi \coloneqq \{a \cdot y \cdot b \cdot y \cdot c \ : \ y \in \Sigma_n^s \text{ and } abc \in \varphi(y)\}.$$

Notice that in the binary case, we had $\varphi(y) = \{010, 101\}$ if $y \notin S$ and $\varphi(y) = \{000, 111\}$ otherwise. Our goal is to define φ such that X_φ is $(k-1)$-circular but not circular.

(E). *Let $a \in \Sigma_n$ and $i \in \{1, \ldots, s\}$.*

(1) If $abb \notin \bigcup_{y \in \Sigma_n^s} \varphi(y)$ for every $b \in \Sigma_n$, then for every $w \in \Sigma_n^{i-1}$ the component C_i of $\mathcal{G}(X_\varphi)$ has no directed path of length 2 starting at $a \cdot w$.

(2) If $bba \notin \bigcup_{y \in \Sigma_n^s} \varphi(y)$ for every $b \in \Sigma_n$, then for every $w \in \Sigma_n^{i-1}$ the component C_i of $\mathcal{G}(X_\varphi)$ has no directed path of length 2 ending at $a \cdot w$.

In particular, if at least one of (1) or (2) holds for every $a \in \Sigma_n$, then C_i is acyclic.

PROOF. Suppose that v is an $(\ell-i)$-node in the component C_i that is the middle vertex of a directed path of length 2. It follows that C_i contains an arc $e = u \to v$ and an arc $e' = v \to u'$, where u and u' are i-nodes. The arc e corresponds to a word $a \cdot y \cdot b \cdot y \cdot c$ with $y \in \Sigma_n^s$ and $abc \in \varphi(y)$. Let us write $y = y_1 \cdot y_2$ with $|y_1| = i-1$, so that $u = a \cdot y_1$ and $v = y_2 \cdot b \cdot y_1 \cdot y_2 \cdot c$. Similarly, the arc e' corresponds to a word $a' \cdot y' \cdot b' \cdot y' \cdot c'$ with $y' \in \Sigma_n^s$ and $a'b'c' \in \varphi(y')$. Writing $y' = y'_1 \cdot y'_2$ with $|y'_2| = i-1$, we obtain $u' = y'_2 \cdot c'$ and $v = a' \cdot y'_1 \cdot y'_2 \cdot b' \cdot y'_1$. Consequently

$$y_2 \cdot b \cdot y_1 \cdot y_2 \cdot c = a' \cdot y'_1 \cdot y'_2 \cdot b' \cdot y'_1,$$

so in particular

$$y_2 \cdot b = a' \cdot y'_1 \quad \text{and} \quad y_2 \cdot c = b' \cdot y'_1.$$

This implies that $a' = b'$ and $b = c$, which yields both statements of (E).

Let D be a digraph. A vertex with exactly one neighbour in D (which can be either an out-neighbour or an in-neighbour) is a *leaf* of D. The digraph D is a *hairy cycle* if it is obtained from a directed cycle $\mathcal{C}$ by adding leaves linked to $\mathcal{C}$.

(F). *If for every $y \in \Sigma_n^s$ the graph $\mathcal{G}(\varphi(y))$ is a union of hairy cycles such that every 1-letter word belongs to a cycle, then the middle component of $\mathcal{G}(X_\varphi)$ is also a union of hairy cycles such that every $(s+1)$-letter word belongs to a cycle.*

PROOF. For every $(s+2)$-word $w = a \cdot y \cdot b$ where $|y| = s$, the arcs incident to w in $\mathcal{G}(X_\varphi)$ are in natural bijection with the arcs incident to ab in $\mathcal{G}(\varphi(y))$. It thus follows by assumption that in $\mathcal{G}(X_\varphi)$, the vertex w either is a leaf or w has exactly one in-neighbour and one out-neighbour. Consider now any $(s+1)$-word $w = a \cdot y = y' \cdot a'$ where $|y| = s = |y'|$. By assumption, w has exactly one out-neighbour $b \cdot y \cdot c$ such that $a \to bc$ belongs to a cycle of $\mathcal{G}(\varphi(y))$ and, similarly, exactly one in-neighbour $b' \cdot y' \cdot c'$ such that $b'c' \to a'$ belongs to a cycle of $\mathcal{G}(\varphi(y'))$. The conclusion of (F) follows.

Fix $y \in \Sigma_n^s$. The code $\varphi(y)$ is to be chosen among the family $\mathcal{F}$: specifically, we define the 3-letter codes $X_{2,n}, \ldots, X_{n,n}$, which generate $\mathcal{F}$ by letter permutations. We make sure that for each $i \in \{2, \ldots, n\}$, the code $X_{i,n}$ satisfies the assumptions of (E) and (F) and, in addition, is such that $\mathcal{G}(X_{i,n})$ is the disjoint union of a hairy cycle of length $2i$ and of $(n-i)$ hairy cycles of length 2.
We identify Σ_n to $\mathbf{Z}_n$ and perform all arithmetic operations on letters in $\mathbf{Z}_n$, unless the letter is explicitly referenced as a member of another cyclic group $\mathbf{Z}_i$.
We now deal with the case $n \geq 4$, the case where $n = 3$ being dealt with in the same manner, except with different sets $X_{i,n}$ given later on. We set

$$\begin{aligned} Y_1 &\coloneqq \{010, 100\}, \\ Y_2 &\coloneqq \{010, 101\}, \\ Y_3 &\coloneqq \{311, 231, 013, 132, 320\}, \\ \forall i \geq 4, \quad Y_i &\coloneqq \bigcup_{j \in \mathbf{Z}_i} \big\{(j-1)(j+1)(j),\ (j+1)(j)(j)\big\}. \end{aligned}$$

For each $j \in \mathbf{Z}_n$, we define $B_{j,n} \coloneqq \big\{(j)(j+1)(j),\ (j+1)(j)(j)\big\}$, and for each $i \in \{1, \ldots, n\}$,

$$X_{i,n} \coloneqq Y_i \cup \bigcup_{j=i}^{n-1} B_{j,n}.$$

For example, when $n = 5$ we obtain

$$\begin{aligned} X_{1,5} &\coloneqq \{010, 100\} \cup \{121, 211, 232, 322, 343, 433, 404, 044\}, \\ X_{2,5} &\coloneqq \{010, 101\} \cup \{232, 322, 343, 433, 404, 044\}, \\ X_{3,5} &\coloneqq \{311, 231, 013, 132, 320\} \cup \{343, 433, 404, 044\}, \\ X_{4,5} &\coloneqq \{310, 100, 021, 211, 132, 322, 203, 033\} \cup \{404, 044\}, \\ X_{5,5} &\coloneqq \{410, 100, 021, 211, 132, 322, 243, 433, 304, 044\}. \end{aligned}$$

(G). *For each $i \in \{1, \ldots, n\}$,*

(1) the graph $\mathcal{G}(Y_i)$ is a hairy cycle of length $2i$ (with leaves attached only to 1-nodes) if $i \notin \{1, 3\}$; the graph $\mathcal{G}(Y_1)$ is the disjoint union of a hairy cycle of length 2 and an arc, while $\mathcal{G}(Y_3)$ is the disjoint union of a hairy cycle of length 6 and a star with centre 3;

(2) for each $j \in \mathbf{Z}_n$, the graph $\mathcal{G}(B_{j,n})$ is the disjoint union of a hairy cycle of length 2 and an arc starting at $j+1$;

(3) the graph $\mathcal{G}(X_{i,n})$ is the union of one hairy cycle of length $2i$ and $n-i$ hairy cycles of length 2. Each 1-node contained in a (hairy) cycle of length 2 is called a fixed point *of $X_{i,n}$.*

Proof. (1) The statement holds if $i \in \{1, 2, 3, 4\}$, as one directly checks on the digraphs depicted in Figures 8, 9 and 10. We now proceed by induction on $i \in \{4, \ldots, n\}$, the statement being true if $i = 4$. Let $i \in \{5, \ldots, n\}$. Notice that Y_{i+1} is obtained from Y_i by removing the three words $(i-1)10$, $0(i-1)(i-1)$, $(i-2)0(i-1)$ and adding the five words $(i-1)0i$, $i(i-1)(i-1)$, $(i-2)i(i-1)$, $i10$ and $0ii$. It follows that $\mathcal{G}(Y_{i+1})$ is obtained from $\mathcal{G}(Y_i)$ by replacing the path

$$(i-2) \to 0(i-1) \to (i-1) \to 10 \to 0$$

by the path

$$(i-2) \to i(i-1) \to i-1 \to 0i \to i \to 10 \to 0$$

along with the attached leaves as presented in Figure 11. The conclusion follows.

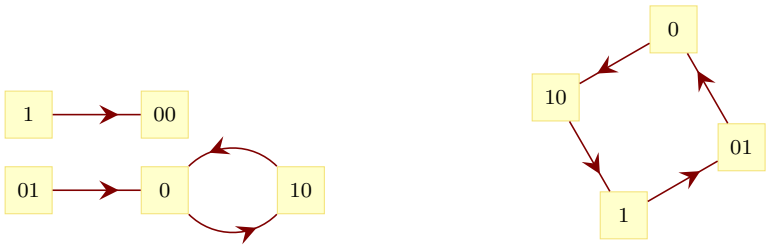

FIGURE 8. The graphs $\mathcal{G}(Y_1)$ (left) and $\mathcal{G}(Y_2)$ (right).

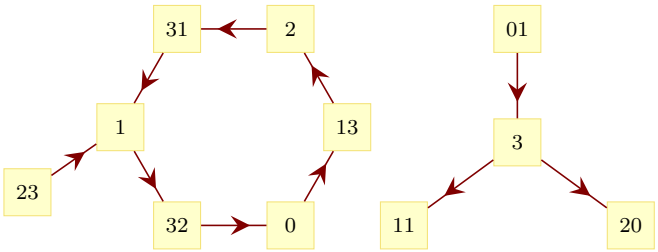

FIGURE 9. The graph $\mathcal{G}(Y_3)$.

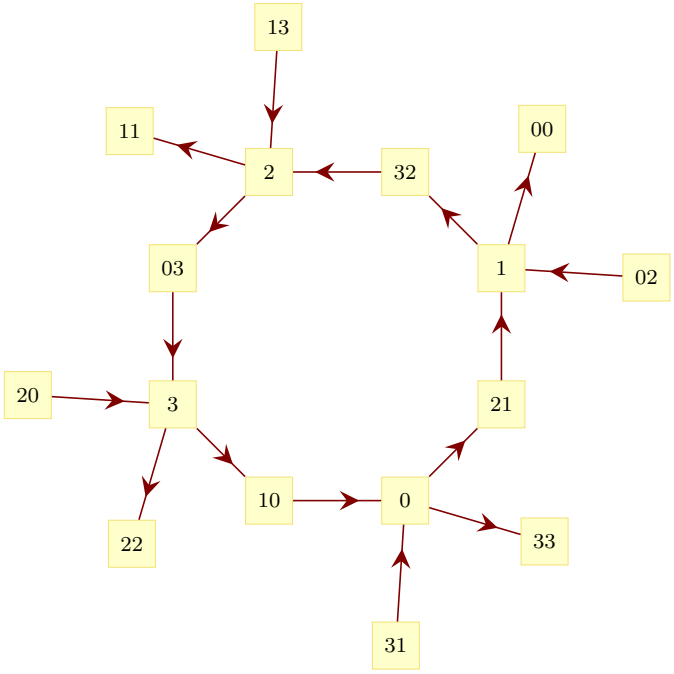

FIGURE 10. The graph $\mathcal{G}(Y_4)$.

(2) Since $B_{j,n}$ is obtained from Y_1 by applying the permutation $x \mapsto j + x$ to every letter, the graph $\mathcal{G}(B_{j,n})$ is isomorphic to $\mathcal{G}(Y_1)$, presented in Figure 8, with $j+1$ being isomorphic to 1. Moreover, if $j \neq j'$ then $\mathcal{G}(B_{j,n})$ and $\mathcal{G}(B_{j',n})$ have disjoint sets of vertices unless $j' \in \{j-1, j+1\}$, in which case the two graphs intersect on exactly one 1-node, which belongs to a cycle in exactly one of them. Therefore, $\mathcal{G}\left(\cup_{j\in\mathbf{Z}_n} B_{j,n}\right)$ is the union of n hairy cycles, each containing exactly one 1-node.

(3) Observe that if $i \in \{1, \ldots, n-1\}$ then Y_i and $B_{j,n}$ are codes over disjoint alphabets whenever $j \in \{i, \ldots, n-2\}$. Note also that since $i \leq n-1$, the graphs $\mathcal{G}(Y_i)$ and $\mathcal{G}(B_{n-1,n})$ intersect precisely on

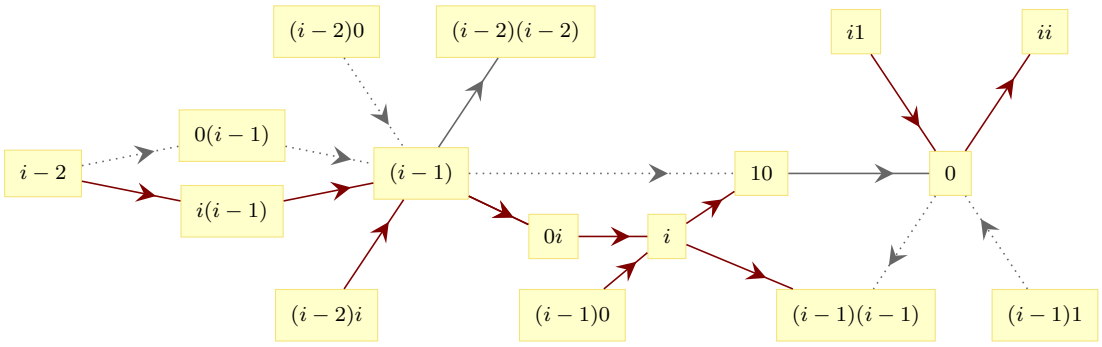

FIGURE 11. The changes in the middle component in the proof of (G)(1). All grey arcs belong to $\mathcal{G}(Y_i)$, and all plain arcs belong to $\mathcal{G}(Y_{i+1})$.

the 1-node 0, which is a leaf in $\mathcal{G}(B_{n-1,n})$. The conclusion now follows from (1) and (2), which ends the proof of (G).

Notice that if $\ell = 3$ (and hence $s = 0$ and $k = n$) then the code X_φ is equal to $\varphi(\varepsilon)$, where ε is the empty word; consequently, it suffices to choose $\varphi(\varepsilon) = X_{n,n}$ by (G).
We now assume that $\ell > 3$, and hence $s \geq 1$. We are ready to build φ, which we do in several steps. We let φ_i be the function φ defined at step i. We start by defining an order on Σ_n^s as follows. Given $y \in \Sigma_n^s$, let $\max(y)$ be the largest letter occurring in y, and for $a \in \mathbf{Z}_n$ let $|y|_a$ be the number of occurrences of a in y. Let $o(y) \coloneqq \big(\max(y), |y|_{\max(y)}\big)$ and fix a total order $\prec$ on Σ_n^s compatible with the lexicographic order with respect to o. For every word $w \in \Sigma_n^{s+1}$, let $\mathcal{C}_\varphi(w)$ be the set of all $(s+1)$-nodes of the connected component of $\mathcal{G}(X_\varphi)$ that contains w. We write $\mathcal{C}_i$ for $\mathcal{C}_{\varphi_i}\big(0^{s+1}\big)$.
We set $\varphi_0(y) \coloneqq \bigcup_{j \in \mathbf{Z}_n} B_{j,n}$ for each $y \in \Sigma_n^s$. At each step, we consider the smallest word $y \in \Sigma_n^s$ not considered yet according to the order defined above, and we obtain φ_{i+1} from φ_i by changing $\varphi_i(y)$. In addition to the properties mentioned before, we make sure that the following invariant is satisfied.

(H). *When we consider a word $y \in \Sigma_n^s$, the set $\mathcal{C}_i$ is composed of all the circular shifts of the words $a \cdot y'$ for $y' \prec y$ and $a \in \Sigma_n$.*

We thus start with 0^s. Note that $\mathcal{C}_0 = \{0^{s+1}\}$. We set $\varphi_1(0^s) \coloneqq X_{n,n}$. By (F) and (G), we know that $\mathcal{C}_1$ contains all circular shifts of $a \cdot 0^s$ for $a \in \Sigma_n$. We now consider step i: let $y \in \Sigma_n^s$ such that each $y' \prec y$ has been considered. Let $A \coloneqq \{a \in \Sigma_n \ : \ a \cdot y \in \mathcal{C}_i\}$. Notice that if $a \in A$ then all circular shifts of $a \cdot y$ belong to $\mathcal{C}_i$ thanks to (H). Furthermore, $0 \in A$. Indeed, let $b \coloneqq \max(y)$ and write $y = y_1 \cdot b \cdot y_2$. Because $0^s \prec y$, we know that $b \neq 0$. Consequently $y_2 \cdot 0 \cdot y_1 \prec y$. Therefore (H) implies that all circular shifts of $b \cdot y_2 \cdot 0 \cdot y_1$ belong to $\mathcal{C}_i$, so in particular $0 \cdot y$ does.
Let $m \coloneqq |A| - 1$. We let $\varphi_{i+1}(y)$ be the code obtained from $X_{n-m,n}$ by permuting the letters such that the set of fixed points is $A \setminus \{0\}$. We need to prove that $\mathcal{C}_{i+1}$ is still stable under circular shifts and contains $a \cdot y$ for each $a \in \Sigma_n$.
To see this, first note that for every $(s+1)$-node w, if the arcs incident to w in $\mathcal{G}(X_{\varphi_i})$ and in $\mathcal{G}(X_{\varphi_{i+1}})$ are not the same, then y is a prefix or a suffix of w. It follows that the only arcs incident to a given node in $\mathcal{C}_i$ that may change at step i are those forming a path of the form $a \cdot y \to (a+1) \cdot y \cdot a \to y \cdot a$ for some $a \in A$. If $a \in A \setminus \{0\}$ then a is a fixed point of $\varphi_{i+1}(y)$, and hence there still is a path of length 2 from $a \cdot y$ to $y \cdot a$ in $\mathcal{G}(X_{\varphi_{i+1}})$. It follows that $\mathcal{C}_i \subseteq \mathcal{C}_{i+1}$. Let $a \in \Sigma_n \setminus A$ and consider $w = y \cdot a$, so that $w \notin \mathcal{C}_i$. By (H), this implies that every s-factor of w is greater than or equal to y. Therefore, writing $0, a_1, \ldots, a_{n-m-1}$ the 1-letter words contained in the cycle of length $2(n-m)$ of $\mathcal{G}(\varphi_{i+1}(y))$ (in

cyclic order), we infer that $\mathcal{G}(X_{\varphi_{i+1}})$ contains a directed path going (in order) through the following $(s+1)$-nodes:

$$0 \cdot y,\ y \cdot a_1,\ \ldots, a_1 \cdot y,\ y \cdot a_2,\ \ldots,\ a_2 \cdot y, \ldots, a_{n-m-1} \cdot y,\ y \cdot 0.$$

Consequently, $\mathcal{C}_{i+1}$ satisfies (H).

We let φ be $\varphi_{n^{s+1}}$. We observe that for every $y \in Y$, the code $\varphi(y)$ contains no word of the form aab, and hence by (E) the component C_i is acyclic for each $i \in \{1, \ldots, s\}$. Moreover, by (F) and (G) the middle component C_{s+1} of $\mathcal{G}(X_\varphi)$ is a union of hairy cycles. By (H), all $(s+1)$-nodes belong to $\mathcal{C}_\varphi(0^{s+1})$, and hence C_{s+1} is composed of a unique hairy cycle, of length $2k$. This concludes the case where $n \geq 4$.

When $n = 3$, we keep the same procedure and always choose $\varphi(y)$ among one of the following four codes (without any letter permutation).

$$\begin{aligned}
X_{3,3} &\coloneqq \{001, 011, 102, 022, 212, 120\},\\
X_{2,3} &\coloneqq \{002, 022, 212, 120, 101, 011\},\\
X'_{2,3} &\coloneqq \{001, 011, 202, 022, 121, 210\},\\
X_{1,3} &\coloneqq \{012, 120, 201\}.
\end{aligned}$$

We observe that (E) and (G) are still satisfied by these codes, which ends the proof of Lemma I.2. □

Theorem 4.4 is now implied by Lemmas 5.1, I.1 and I.2. We conclude by giving three examples.

Example I.3.

(1) If $\ell = 5$ and $n = 3$, then $s = 1$ and the order $\prec$ over Σ_3 can be chosen to be the natural order over integers. One has $\varphi(0) = \varphi(1) = X_{3,3}$ and $\varphi(2) = X_{2,3}$. The obtained code is then

$$\begin{aligned}
\{&00001, 00101, 10002, 00202, 20102, 10200,\\
&01011, 01111, 11012, 01212, 21112, 11210,\\
&02022, 02222, 22122, 12220, 12021, 02121\}.
\end{aligned}$$

(2) If $\ell = 7$ and $n = 3$, then $s = 2$ and the order $\prec$ over Σ_3^2 can be chosen to be

$$00 \prec 01 \prec 10 \prec 11 \prec 02 \prec 20 \prec 12 \prec 21 \prec 22.$$

One has

$$\begin{aligned}
&\varphi(00) = X_{3,3}, && \varphi(01) = X_{3,3}, && \varphi(10) = X_{2,3},\\
&\varphi(11) = X_{3,3}, && \varphi(02) = X_{2,3}, && \varphi(20) = X_{1,3},\\
&\varphi(12) = X_{2,3}, && \varphi(21) = X_{1,3}, && \varphi(22) = X_{2,3}.
\end{aligned}$$

(3) If $\ell = 7$ and $n = 4$, then $s = 2$ and the order $\prec$ over Σ_4^2 can be chosen to be

$$00 \prec 01 \prec 10 \prec 11 \prec 02 \prec 20 \prec 12 \prec 21 \prec 22 \prec 03 \prec 30 \prec 13 \prec 31 \prec 23 \prec 32 \prec 33.$$

Let $\widetilde{X}_{3,4}$ and $\widetilde{X}_{2,4}$ be respectively obtained from $X_{3,4}$ and $X_{2,4}$ by permuting 1 and 3 so that their respective sets of fixed points are $\{1\}$ and $\{1, 2\}$. One has

$$\begin{aligned}
&\varphi(00) = X_{4,4}, && \varphi(01) = X_{4,4}, && \varphi(10) = \widetilde{X}_{3,4}, && \varphi(11) = X_{4,4},\\
&\varphi(02) = \widetilde{X}_{3,4}, && \varphi(20) = \widetilde{X}_{2,4}, && \varphi(12) = \widetilde{X}_{3,4}, && \varphi(21) = \widetilde{X}_{2,4},\\
&\varphi(22) = \widetilde{X}_{3,4}, && \varphi(03) = \widetilde{X}_{2,4}, && \varphi(30) = X_{1,4}, && \varphi(13) = \widetilde{X}_{2,4},\\
&\varphi(31) = X_{1,4}, && \varphi(23) = \widetilde{X}_{2,4}, && \varphi(32) = X_{1,4}, && \varphi(33) = \widetilde{X}_{2,4}.
\end{aligned}$$

Appendix II. List of 52 maximal 1-circular codes encoding all 20 amino acids.

The list below gives all 52 perfect matchings of the graph $\mathcal{G}$ from Lemma 6.1. The amino acids are listed in the order of their assignments to the 13 complete equivalence classes $D_1, \ldots, D_{20}$.

Code: 01 $D_1 \ldots D_{20}$: Asp Ile His Thr Glu Val Ser Tyr Ala Pro Arg Cys Leu
Code: 02 $D_1 \ldots D_{20}$: Asp Ile His Thr Glu Val Ser Tyr Ala Pro Arg Leu Cys
Code: 03 $D_1 \ldots D_{20}$: Asp Ile His Thr Glu Val Ser Tyr Pro Leu Arg Ala Cys
Code: 04 $D_1 \ldots D_{20}$: Asp Ile Pro Thr Glu Ser His Tyr Ala Leu Arg Cys Val
Code: 05 $D_1 \ldots D_{20}$: Asp Ile Pro Thr Glu Ser His Tyr Arg Leu Val Ala Cys
Code: 06 $D_1 \ldots D_{20}$: Asp Ile Pro Thr Glu Val His Tyr Ala Ser Arg Cys Leu
Code: 07 $D_1 \ldots D_{20}$: Asp Ile Pro Thr Glu Val His Tyr Ala Ser Arg Leu Cys
Code: 08 $D_1 \ldots D_{20}$: Asp Ile Pro Thr Glu Val His Tyr Arg Leu Ser Ala Cys
Code: 09 $D_1 \ldots D_{20}$: Asp Ile Thr Leu Glu Ser His Tyr Ala Pro Arg Cys Val
Code: 10 $D_1 \ldots D_{20}$: Asp Ile Thr Leu Glu Ser His Tyr Arg Pro Val Ala Cys
Code: 11 $D_1 \ldots D_{20}$: Asp Ile Thr Leu Glu Val His Tyr Arg Pro Ser Ala Cys
Code: 12 $D_1 \ldots D_{20}$: Asp Ile Thr Leu Glu Val His Tyr Pro Ser Arg Ala Cys
Code: 13 $D_1 \ldots D_{20}$: Asp Ile Thr Tyr Glu Ser His Leu Ala Pro Arg Cys Val
Code: 14 $D_1 \ldots D_{20}$: Asp Ile Thr Tyr Glu Ser His Leu Arg Pro Val Ala Cys
Code: 15 $D_1 \ldots D_{20}$: Asp Ile Thr Tyr Glu Val His Leu Arg Pro Ser Ala Cys
Code: 16 $D_1 \ldots D_{20}$: Asp Ile Thr Tyr Glu Val His Leu Pro Ser Arg Ala Cys
Code: 17 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Ser Ile Tyr Arg Pro Val Cys Leu
Code: 18 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Ser Ile Tyr Arg Pro Val Leu Cys
Code: 19 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Ser Ile Tyr Pro Leu Arg Cys Val
Code: 20 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Val Ile Tyr Arg Pro Ser Cys Leu
Code: 21 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Val Ile Tyr Arg Pro Ser Leu Cys
Code: 22 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Val Ile Tyr Pro Ser Arg Cys Leu
Code: 23 $D_1 \ldots D_{20}$: Glu Asp His Thr Ala Val Ile Tyr Pro Ser Arg Leu Cys
Code: 24 $D_1 \ldots D_{20}$: Glu Asp His Thr Ser Val Ile Tyr Ala Pro Arg Cys Leu
Code: 25 $D_1 \ldots D_{20}$: Glu Asp His Thr Ser Val Ile Tyr Ala Pro Arg Leu Cys
Code: 26 $D_1 \ldots D_{20}$: Glu Asp His Thr Ser Val Ile Tyr Pro Leu Arg Ala Cys
Code: 27 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Ser His Ile Arg Pro Val Cys Leu
Code: 28 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Ser His Ile Arg Pro Val Leu Cys
Code: 29 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Ser His Ile Pro Leu Arg Cys Val
Code: 30 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Val His Ile Arg Pro Ser Cys Leu
Code: 31 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Val His Ile Arg Pro Ser Leu Cys
Code: 32 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Val His Ile Pro Ser Arg Cys Leu
Code: 33 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ala Val His Ile Pro Ser Arg Leu Cys
Code: 34 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ser Val His Ile Ala Pro Arg Cys Leu
Code: 35 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ser Val His Ile Ala Pro Arg Leu Cys
Code: 36 $D_1 \ldots D_{20}$: Glu Asp Thr Tyr Ser Val His Ile Pro Leu Arg Ala Cys
Code: 37 $D_1 \ldots D_{20}$: Thr Asp His Leu Glu Ser Ile Tyr Ala Pro Arg Cys Val
Code: 38 $D_1 \ldots D_{20}$: Thr Asp His Leu Glu Ser Ile Tyr Arg Pro Val Ala Cys
Code: 39 $D_1 \ldots D_{20}$: Thr Asp His Leu Glu Val Ile Tyr Arg Pro Ser Ala Cys
Code: 40 $D_1 \ldots D_{20}$: Thr Asp His Leu Glu Val Ile Tyr Pro Ser Arg Ala Cys
Code: 41 $D_1 \ldots D_{20}$: Thr Asp His Tyr Glu Ser Ile Leu Ala Pro Arg Cys Val
Code: 42 $D_1 \ldots D_{20}$: Thr Asp His Tyr Glu Ser Ile Leu Arg Pro Val Ala Cys

Code: 43 $D_1 \dots D_{20}$: Thr Asp His Tyr Glu Val Ile Leu Arg Pro Ser Ala Cys
Code: 44 $D_1 \dots D_{20}$: Thr Asp His Tyr Glu Val Ile Leu Pro Ser Arg Ala Cys
Code: 45 $D_1 \dots D_{20}$: Thr Asp His Tyr Glu Val Ser Ile Ala Pro Arg Cys Leu
Code: 46 $D_1 \dots D_{20}$: Thr Asp His Tyr Glu Val Ser Ile Ala Pro Arg Leu Cys
Code: 47 $D_1 \dots D_{20}$: Thr Asp His Tyr Glu Val Ser Ile Pro Leu Arg Ala Cys
Code: 48 $D_1 \dots D_{20}$: Thr Asp Pro Tyr Glu Ser His Ile Ala Leu Arg Cys Val
Code: 49 $D_1 \dots D_{20}$: Thr Asp Pro Tyr Glu Ser His Ile Arg Leu Val Ala Cys
Code: 50 $D_1 \dots D_{20}$: Thr Asp Pro Tyr Glu Val His Ile Ala Ser Arg Cys Leu
Code: 51 $D_1 \dots D_{20}$: Thr Asp Pro Tyr Glu Val His Ile Ala Ser Arg Leu Cys
Code: 52 $D_1 \dots D_{20}$: Thr Asp Pro Tyr Glu Val His Ile Arg Leu Ser Ala Cys

The list below gives all 52 maximal 1-circular codes encoding all 20 amino acids in lexicographical order.

$\{AAC, AAG, ACC, AGT, ATA, ATG, CAG, CAT, CCT, CGC, CTA, GAC, GAG, GCT, GGC, GTC, TAT, TGG, TGT, TTC\}$
$\{AAC, AAG, ACC, AGT, ATA, ATG, CAG, CAT, CCT, CGC, GAC, GAG, GCT, GGC, GTC, TAC, TGG, TGT, TTA, TTC\}$
$\{AAC, AAG, ACC, AGT, ATA, ATG, CAG, CAT, CCT, CGT, CTA, GAC, GAG, GCC, GGC, GTT, TAT, TGC, TGG, TTC\}$
$\{AAC, AAG, ACC, AGT, ATA, ATG, CAG, CAT, CCT, CGT, GAC, GAG, GCC, GGC, GTT, TAC, TGC, TGG, TTA, TTC\}$
$\{AAC, AAG, ACC, ATA, ATG, CAG, CAT, CCG, CGT, CTA, GAC, GAG, GCT, GGC, GTA, TAT, TCC, TGG, TGT, TTC\}$
$\{AAC, AAG, ACC, ATA, ATG, CAG, CAT, CCG, CGT, GAC, GAG, GCT, GGC, GTA, TAC, TCC, TGG, TGT, TTA, TTC\}$
$\{AAC, AAG, ACC, ATA, ATG, CAG, CAT, CCT, CGC, CTA, GAC, GAG, GCT, GGC, GTA, TAT, TCG, TGG, TGT, TTC\}$
$\{AAC, AAG, ACC, ATA, ATG, CAG, CAT, CCT, CGC, GAC, GAG, GCT, GGC, GTA, TAC, TCG, TGG, TGT, TTA, TTC\}$
$\{AAC, AAG, ACT, AGT, ATA, ATG, CAG, CAT, CCA, CGC, CTC, GAC, GAG, GCT, GGC, GTC, TAT, TGG, TGT, TTC\}$
$\{AAC, AAG, ACT, AGT, ATA, ATG, CAG, CAT, CCA, CGT, CTC, GAC, GAG, GCC, GGC, GTT, TAT, TGC, TGG, TTC\}$
$\{AAC, AAG, ACT, ATA, ATG, CAC, CAG, CCG, CGT, CTC, GAC, GAG, GCT, GGC, GTA, TAT, TCA, TGG, TGT, TTC\}$
$\{AAC, AAG, ACT, ATA, ATG, CAC, CAG, CCT, CGT, CTG, GAC, GAG, GCC, GGC, GTA, TAT, TCA, TGG, TGT, TTC\}$
$\{AAC, AAG, ACT, ATA, ATG, CAC, CAG, CCT, CGT, GAC, GAG, GCC, GGC, GTA, TAT, TCA, TGC, TGG, TTC, TTG\}$
$\{AAC, AAG, ACT, ATA, ATG, CAG, CAT, CCA, CGC, CTC, GAC, GAG, GCT, GGC, GTA, TAT, TCG, TGG, TGT, TTC\}$
$\{AAC, AAG, ACT, ATA, ATG, CAG, CAT, CCA, CGT, CTG, GAC, GAG, GCC, GGC, GTA, TAT, TCC, TGG, TGT, TTC\}$
$\{AAC, AAG, ACT, ATA, ATG, CAG, CAT, CCA, CGT, GAC, GAG, GCC, GGC, GTA, TAT, TCC, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACA, AGT, ATC, ATG, CAC, CAG, CCT, CGC, CTA, GAC, GAG, GCT, GGC, GTC, TAT, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, AGT, ATC, ATG, CAC, CAG, CCT, CGC, GAC, GAG, GCT, GGC, GTC, TAC, TGG, TGT, TTA, TTC\}$
$\{AAG, AAT, ACA, AGT, ATC, ATG, CAC, CAG, CCT, CGT, CTA, GAC, GAG, GCC, GGC, GTT, TAT, TGC, TGG, TTC\}$
$\{AAG, AAT, ACA, AGT, ATC, ATG, CAC, CAG, CCT, CGT, GAC, GAG, GCC, GGC, GTT, TAC, TGC, TGG, TTA, TTC\}$
$\{AAG, AAT, ACA, AGT, ATG, ATT, CAG, CAT, CCA, CGC, CTC, GAC, GAG, GCT, GGC, GTC, TAC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, AGT, ATG, ATT, CAG, CAT, CCA, CGT, CTC, GAC, GAG, GCC, GGC, GTT, TAC, TGC, TGG, TTC\}$
$\{AAG, AAT, ACA, ATC, ATG, CAC, CAG, CCG, CGT, CTA, GAC, GAG, GCT, GGC, GTA, TAT, TCC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, ATC, ATG, CAC, CAG, CCG, CGT, GAC, GAG, GCT, GGC, GTA, TAC, TCC, TGG, TGT, TTA, TTC\}$
$\{AAG, AAT, ACA, ATC, ATG, CAC, CAG, CCT, CGC, CTA, GAC, GAG, GCT, GGC, GTA, TAT, TCG, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, ATC, ATG, CAC, CAG, CCT, CGC, GAC, GAG, GCT, GGC, GTA, TAC, TCG, TGG, TGT, TTA, TTC\}$
$\{AAG, AAT, ACA, ATG, ATT, CAC, CAG, CCG, CGT, CTC, GAC, GAG, GCT, GGC, GTA, TAC, TCA, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, ATG, ATT, CAC, CAG, CCT, CGT, CTG, GAC, GAG, GCC, GGC, GTA, TAC, TCA, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, ATG, ATT, CAC, CAG, CCT, CGT, GAC, GAG, GCC, GGC, GTA, TAC, TCA, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACA, ATG, ATT, CAG, CAT, CCA, CGC, CTC, GAC, GAG, GCT, GGC, GTA, TAC, TCG, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, ATG, ATT, CAG, CAT, CCA, CGT, CTG, GAC, GAG, GCC, GGC, GTA, TAC, TCC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACA, ATG, ATT, CAG, CAT, CCA, CGT, GAC, GAG, GCC, GGC, GTA, TAC, TCC, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACC, AGC, ATG, ATT, CAA, CAT, CCG, CGT, CTC, GAC, GAG, GCT, GGC, GTA, TAC, TGG, TGT, TTC\}$

$\{AAG, AAT, ACC, AGC, ATG, ATT, CAA, CAT, CCT, CGT, CTG, GAC, GAG, GCC, GGC, GTA, TAC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACC, AGC, ATG, ATT, CAA, CAT, CCT, CGT, GAC, GAG, GCC, GGC, GTA, TAC, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACC, AGT, ATG, ATT, CAA, CAT, CCG, CGT, CTC, GAC, GAG, GCA, GGC, GTT, TAC, TGC, TGG, TTC\}$
$\{AAG, AAT, ACC, AGT, ATG, ATT, CAA, CAT, CCT, CGC, CTG, GAC, GAG, GCA, GGC, GTC, TAC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACC, AGT, ATG, ATT, CAA, CAT, CCT, CGC, GAC, GAG, GCA, GGC, GTC, TAC, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACC, ATG, ATT, CAA, CAT, CCG, CGT, CTG, GAC, GAG, GCA, GGC, GTA, TAC, TCC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACC, ATG, ATT, CAA, CAT, CCG, CGT, GAC, GAG, GCA, GGC, GTA, TAC, TCC, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACC, ATG, ATT, CAA, CAT, CCT, CGC, CTG, GAC, GAG, GCA, GGC, GTA, TAC, TCG, TGG, TGT, TTC\}$
$\{AAG, AAT, ACC, ATG, ATT, CAA, CAT, CCT, CGC, GAC, GAG, GCA, GGC, GTA, TAC, TCG, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACT, AGC, ATC, ATG, CAA, CAC, CCG, CGT, CTC, GAC, GAG, GCT, GGC, GTA, TAT, TGG, TGT, TTC\}$
$\{AAG, AAT, ACT, AGC, ATC, ATG, CAA, CAC, CCT, CGT, CTG, GAC, GAG, GCC, GGC, GTA, TAT, TGG, TGT, TTC\}$
$\{AAG, AAT, ACT, AGC, ATC, ATG, CAA, CAC, CCT, CGT, GAC, GAG, GCC, GGC, GTA, TAT, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACT, AGT, ATC, ATG, CAA, CAC, CCG, CGT, CTC, GAC, GAG, GCA, GGC, GTT, TAT, TGC, TGG, TTC\}$
$\{AAG, AAT, ACT, AGT, ATC, ATG, CAA, CAC, CCT, CGC, CTG, GAC, GAG, GCA, GGC, GTC, TAT, TGG, TGT, TTC\}$
$\{AAG, AAT, ACT, AGT, ATC, ATG, CAA, CAC, CCT, CGC, GAC, GAG, GCA, GGC, GTC, TAT, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACT, ATC, ATG, CAA, CAC, CCG, CGT, CTG, GAC, GAG, GCA, GGC, GTA, TAT, TCC, TGG, TGT, TTC\}$
$\{AAG, AAT, ACT, ATC, ATG, CAA, CAC, CCG, CGT, GAC, GAG, GCA, GGC, GTA, TAT, TCC, TGC, TGG, TTC, TTG\}$
$\{AAG, AAT, ACT, ATC, ATG, CAA, CAC, CCT, CGC, CTG, GAC, GAG, GCA, GGC, GTA, TAT, TCG, TGG, TGT, TTC\}$
$\{AAG, AAT, ACT, ATC, ATG, CAA, CAC, CCT, CGC, GAC, GAG, GCA, GGC, GTA, TAT, TCG, TGC, TGG, TTC, TTG\}$

References

[1] D. G. Arquès and C. J. Michel, *A complementary circular code in the protein coding genes*, Journal of Theoretical Biology **182** (1996), 45–58.

[2] A. H. Ball and L. J. Cummings, *Extremal digraphs and comma-free codes*, Ars Combinatoria **1** (1976), no. 1, 239–251.

[3] J. Berstel, D. Perrin, and C. Reutenauer, *Codes and automata*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2005.

[4] F. H. Crick, J. S. Griffith, and L. E. Orgel, *Codes without commas*, Proceedings of the National Academy of Sciences U.S.A. **43** (2003), 416–421.

[5] G. Dila, C. J. Michel, O. Poch, R. Ripp, and J. D. Thompson, *Evolutionary conservation and functional implications of circular code motifs in eukaryotic genomes*, Biosystems **175** (2019), 57–74.

[6] G. Dila, R. Ripp, C. Mayer, O. Poch, C. J. Michel, and J. D. Thompson, *Circular code motifs in the ribosome: a missing link in the evolution of translation?*, RNA **25** (2019), 1714–1730.

[7] K. El Soufi and C. J. Michel, *Circular code motifs in the ribosome decoding center*, Computational Biology and Chemistry **52** (2014), 9–17.

[8] ———, *Circular code motifs near the ribosome decoding center*, Computational Biology and Chemistry **59** (2014), 158–176.

[9] E. Fimmel, S. Giannerini, D. L. Gonzalez, and L. Strüngmann, *Circular codes, symmetries and transformations*, Journal of Mathematical Biology **70** (2015), 1623–1644.

[10] E. Fimmel, C. J. Michel, and L. Strüngmann, *n-nucleotide circular codes in graph theory*, Philosophical Transactions of the Royal Society A **374** (2016), 20150058.

[11] ———, *Diletter circular codes over finite alphabets*, Mathematical Biosciences **294** (2017), 120–129.

[12] ———, *Strong comma-free codes in genetic information*, Bulletin of Mathematical Biology **79** (2017), 1796–1819.

[13] E. Fimmel and L. Strüngmann, *On the hierarchy of trinucleotide n-circular codes and their corresponding amino acids*, Journal of Theoretical Biology **364** (2015), 113–120.

[14] ———, *Maximal dinucleotide comma-free codes*, Journal of Theoretical Biology **389** (2016), 206–213.

[15] ———, *Mathematical Fundamentals for the noise immunity of the genetic code*, BioSystems **164** (2018), 186–198.

[16] S. J. Freeland and L. D. Hurst, *The genetic code is one in a million*, Journal of Molecular Evolution **47** (1998), 238–248.

[17] S. W. Golomb, B. Gordon, and L. R. Welch, *Comma-free codes*, Canadian Journal of Mathematics **10** (1958), 202–209.

[18] S. W. Golomb, L. R. Welch, and M. Delbrück, *Construction and properties of comma-free codes*, Biologiske Meddelelser, Kongelige Danske Videnskabernes Selskab **23** (1958), 1–34.

[19] E. V. Koonin, *Frozen accident pushing 50: stereochemistry, expansion, and chance in the evolution of the genetic code*, Life **7** (2017), 1–13.

[20] E. V. Koonin and A. S. Novozhilov, *Origin and evolution of the genetic code: the universal enigma*, IUBMB Life **61** (2009), 99–111.

[21] N. H. Lam, *Completing comma-free codes*, Theoretical computer science **301** (2003), 399–415.

[22] C. J. Michel, *A 2006 review of circular codes in genes*, Computers and Mathematics with Applications **55** (2008), 984–988.

[23] ———, *Circular code motifs in transfer and 16S ribosomal RNAs: a possible translation code in genes*, Computational Biology and Chemistry **37** (2012), 24–37.

[24] ———, *Circular code motifs in transfer RNAs*, Computational biology and chemistry **45** (2013), 17–29.

[25] ———, *A genetic scale of reading frame coding*, Computational biology and chemistry **355** (2014), 83–94.

[26] ———, *The maximal C^3 self-complementary trinucleotide circular code X in genes of bacteria, eukaryotes, plasmids and viruses*, Journal of Theoretical Biology **380** (2015), 156–177.

[27] ———, *The maximal C^3 self-complementary trinucleotide circular code X in genes of bacteria, archaea, eukaryotes, plasmids and viruses*, Life **7** (2017), no. 2, 1–16.

[28] C. J. Michel, V. Nguefack Ngoune, O. Poch, R. Ripp, and J. D. Thompson, *Enrichment of circular code motifs in the genes of the yeast Saccharomyces cerevisiae*, Life **7** (2017), no. 52, 1–20.

[29] C. J. Michel and G. Pirillo, *Identification of all trinucleotide circular codes*, Computational Biology and Chemistry **34** (2010), 122–125.

[30] ———, *A permuted set of a trinucleotide circular code coding the 20 amino acids in variant nuclear codes*, Journal of Theoretical Biology **319** (2013), 116–121.

[31] C. J. Michel, G. Pirillo, and M. A. Pirillo, *Varieties of comma free codes*, Computer and Mathematics with Applications **55** (2008), 989–996.

[32] ———, *A relation between trinucleotide comma-free codes and trinucleotide circular codes*, Theoretical Computer Science **401** (2008), 17–26.

[33] C. J. Michel and J. D. Thompson, *Identification of a circular code periodicity in the bacterial ribosome: origin of codon periodicity in genes?*, RNA Biology **17** (2020), 571–583.

[34] M. W. Nirenberg and J. H. Matthaei, *The dependence of cell-free protein synthesis in* E. coli *upon naturally occurring or synthetic polyribonucleotides*, Proceedings of the National Academy of Sciences U.S.A. **37** (1961), 1588–1602.

[35] S. R. Pelc and M. G. Welton, *Stereochemical relationship between coding triplets and amino-acids*, Nature **209** (1966), 868–870.

[36] G. Pirillo, *A characterization for a set of trinucleotides to be a circular code*, by C. Pellegrini, P. Cerrai, P. Freguglia, V. Benci, G. Israel Determinism, Holism, and Complexity, Kluwer, 2003.

[37] J. D. Watson and F. H. Crick, *Molecular structure of nucleic acids*, Nature **171** (1953), 737–738.

[38] C. R. Woese, *Order in the genetic code*, Proceedings of the National Academy of Sciences of the United States of America **54** (1965), 71–75.

[39] J. T. Wong, *A co-evolution theory of the genetic code*, Proceedings of the National Academy of Sciences of the United States of America **72** (1975), 1909–1912.

[40] M. Yarus, *The genetic code and RNA-amino acid affinities*, Life **7** (2017), 1–16.