Abstract—To equip DNA-based data storage with random-access capabilities, Yazdi et al. (2018) prepondred DNA strands with specially chosen address sequences called primers and provided certain design criteria for these primers. We provide explicit constructions of error-correcting codes that are suitable as primer addresses and equip these constructions with efficient encoding algorithms. Specifically, our constructions take cyclic or linear codes as inputs and produce sets of primers with similar error-correcting capabilities. Using certain classes of BCH codes, we obtain infinite families of primer sets of length $n$, minimum distance $d$ with $(d + 1)\log_q n + O(1)$ redundant symbols. Our techniques involve reversible cyclic codes (1964), an encoding method of Tavares et al. (1971) and Knuth’s constructions of error-correcting codes that are suitable as primer addresses and equipping DNA strands with specially chosen address sequences by hybridization provides selective access to encoded DNA strands through the process of data storage, Yazdi et al. (2018).

In this paper, we study the problem of primer design. To introduce random-access and rewriting capabilities into DNA-based data storage, Yazdi et al. developed an architecture that allows selective access to encoded DNA strands through the process of hybridization. Their technique involves prepping information-carrying DNA strands with specially chosen address sequences called primers. Yazdi et al. provided certain design considerations for these primers [3] and also, verified the feasibility of their architecture in a series of experiments [2], [4].

We continue this investigation and provide efficient and explicit constructions of error-correcting codes that are suitable as primer addresses. Our techniques include novel modifications of Knuth’s balancing technique [5] and involve the use of reversible cyclic codes [6]. We also revisit the work of Tavares et al. (1971) that efficiently encodes messages into cyclic classes of a cyclic code and adapt their method for our codes. We note that reversible cyclic codes have been studied in another coding application for DNA computing and our techniques can be also modified for the latter application. However, due to space constraints, our exposition is focused on the code constructions for primers and technical proofs and detailed descriptions of our encoding methods are omitted. We refer interested readers to the full version for detailed proofs and code constructions for DNA computing [8].

II. PRELIMINARY AND CONTRIBUTIONS

Let $F_q$ denote the finite field of size $q$. Two cases of special interest are $q = 2$ and $q = 4$. In the latter case, we let $\omega$ denote a primitive element of $F_4$ and identify the elements of $F_4$ with the four DNA bases $\Sigma = \{A, C, T, G\}$. Specifically,

$$0 \leftrightarrow A, \quad 1 \leftrightarrow T, \quad \omega \leftrightarrow C, \quad \omega + 1 \leftrightarrow G.$$ 

Hence, for an element $x \in F_4$, its Watson-Crick complement corresponds to $x + 1$.

Let $n$ be a positive integer. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$, while $[n]$ denotes the set $\{0, 1, \ldots, n - 1\}$. For a word $a = (a_1, a_2, \ldots, a_n) \in F_q^n$, let $a[i]$ denote the $i$th symbol $a_i$ and $a[i, j]$ denote the subword of $a$ starting at position $i$ and ending at position $j$. In other words, $a[i, j] = (a_i, a_{i+1}, \ldots, a_j)$, if $i \leq j$; $a[i, j] = (a_i, a_{i+1}, \ldots, a_j)$, otherwise. Moreover, the reverse of $a$, denoted as $a^r$, is $a^r_1, a^r_2, \ldots, a^r_n$; the complement $\Pi$ of $a$ is $\Pi_1, \Pi_2, \ldots, \Pi_n$, where $\Pi = x + 1$ for $x \in F_2$ or $x \in F_4$; and the reverse-complement of $a^r$ is $\overline{a^r}$. For words $a$ and $b$, we use $ab$ to denote the concatenation of $a$ and $b$, and $a^r$ to denote the sequence of length $\ell n$ comprising $\ell$ copies of $a$.

A $q$-ary code $\mathcal{C}$ of length $n$ is a collection of words from $F_q^n$. For two words $a$ and $b$ of the same length, we use $d(a, b)$ to denote the Hamming distance between them. A code $\mathcal{C}$ has minimum Hamming distance $d$ if any two distinct codewords in $\mathcal{C}$ are at least distance $d$ apart. Such a code is denoted as $(n, d)_q$-code. Its size is given by $|\mathcal{C}|$, while its redundancy is given by $n - \log_q |\mathcal{C}|$. An $[n, k, d]_q$-linear code is an $[n, d]_q$-code that is also a $k$-dimensional vector subspace of $F_q^n$. Hence, an $[n, k, d]_q$-linear code has redundancy $n - k$.

A. Cyclic and Reversible Codes

For a vector $a \in F_q^n$, let $\sigma^i(a)$ be the vector obtained by cyclically shifting the components of $a$ to the right $i$ times. So, $\sigma^i(a) = (a_n, a_1, a_2, \ldots, a_{n-1})$. An $[n, k, d]_q$-cyclic code $\mathcal{C}$ is an $[n, k, d]_q$-linear code that is closed under cyclic shifts. In other words, $a \in \mathcal{C}$ implies $\sigma^i(a) \in \mathcal{C}$.

Cyclic codes are well-studied because of their rich algebraic structure. In the theory of cyclic codes (see for example, MacWilliams and Sloane [9, Chapter 7]), we identify a word $c = (c_i)_{i \in [n]}$ of length $n$ with the polynomial $\sum_{i=0}^{n-1} c_iX^i$. Given a cyclic code $\mathcal{C}$ of length $n$ and dimension $k$, there exists a unique monic polynomial $g(X)$ of degree $n-k$ such that $\mathcal{C}$ is given by the set $\{m(X)g(X) : \deg m < k\}$. The polynomial $g(X)$ is referred to as the generator polynomial of $\mathcal{C}$ and we write $\mathcal{C} = \langle g(X) \rangle$. We continue this discussion on this algebraic structure in the full version [8], where we exploit certain polynomial properties for efficient encoding.

When $d$ is fixed, there exists a class of Bose-Chaudhuri-Hocquenghem (BCH) codes that are cyclic codes whose redundancy is asymptotically optimal.

Theorem 1 (Primitive narrow-sense BCH codes [10, Theorem 10]). Fix $m \geq 1$ and $2d \leq 2^m - 1$. Set $n = 2^m - 1$ and $t = [(d - 1)/2]$. There exists an $[n, k, d]_2$-cyclic code $\mathcal{C}$ with $k \geq n - tm$. In other words, $\mathcal{C}$ has redundancy at most $t \log_2(n + 1)$. 

978-1-5386-9291-2/19/$31.00 ©2019 IEEE 91

ISIT 2019
A cyclic code $\mathcal{C}$ is called reversible if $a \in \mathcal{C}$ implies $a^\tau \in \mathcal{C}$. A reversible cyclic code is also known as an LCD (linear code with complementary dual) cyclic code and has been studied extensively [6], [11]–[13]. In this paper, reversible cyclic codes containing the all-one vector are of particular interest. Suppose that $\mathcal{C}$ is one such code. Then for any word $a \in \mathcal{C}$, both its complement $\overline{a} = a + 1^n$ and its reverse-complement $a^{rc} = a^\tau + 1^n$ belong to $\mathcal{C}$.

Recently, Li et al. [12] explored two other classes of BCH codes and determined their minimum distances and dimensions. These codes are reversible cyclic and contain the all-one vector.

**Theorem 2** (Li et al. [12]). Let $m \geq 2$, $m \neq 3$ and $1 \leq \tau \leq \lceil m/2 \rceil$. Let $q$ be even and set $n = q^m - 1$ and $d = q^\tau - 1$. There exists an $[n, k, d]_q$-reversible cyclic code $\mathcal{C}$ that contains $1^n$ and has dimension

$$k = \begin{cases} n - (d + q - 1)m, & \text{if } m \geq 5 \text{ and } \tau = \frac{m+1}{2}; \\ n - (d-1)m, & \text{otherwise}. \end{cases}$$

In other words, $\mathcal{C}$ has redundancy at most $(d-1) \log_q(n+1)$.

**B. Balanced Codes**

A binary word of length $n$ is balanced if $[n/2]$ or $[n/2]$ bits are zero, while a quaternary word of length $n$ is GC-balanced if $[n/2]$ or $[n/2]$ symbols are either $\mathbb{G}$ or $\mathbb{C}$. A binary (or quaternary) code is balanced (resp. GC-balanced) if all its codewords are balanced (resp. GC-balanced).

Motivated by applications in laser disks, Knuth [5] studied balanced binary codes and proposed an efficient method to encode an arbitrary binary message to a binary balanced codeword by introducing $\log_2 n$ redundant bits. Recently, Weber et al. [14] extended Knuth’s scheme to include error-correcting capabilities. Specifically, their construction takes two input codes of distance $d$: a linear code of length $n$ and a short balanced code $\mathcal{C}_p$, and outputs a long balanced code of distance $d$. Even though the balanced code $\mathcal{C}_p$ is only required to be of size $n$, it is unclear how to find one efficiently, especially when $d$ grows with $n$.

On the other hand, GC-balanced codes have been extensively studied in the context of DNA computing and DNA-based storage (see [2], [15], [16] for a survey). However, most constructions are based on search heuristics or apply to a restricted set of parameters. Recently, Yazdi et al. [3] introduced the coupling construction (Lemma 6) that takes two binary error-correcting codes, one of which is balanced, as inputs and outputs a GC-balanced error-correcting code. As with the construction of Weber et al. [14], it is unclear how to find the balanced binary error-correcting code efficiently.

In this work, we avoid these requirements of additional balanced codes. We provide a construction that takes a binary cyclic code (or two binary linear codes) and outputs a binary balanced code (resp. a GC-balanced code) with error-correcting capabilities.

**C. Primer Codes**

In order to introduce random access to DNA-based data storage systems, Yazdi et al. [3] proposed the following criteria for the design of primer addresses.

**Definition 3.** A code $\mathcal{C}$ of length $n$ is $\kappa$-weakly mutually uncorrelated ($\kappa$-WMU) if for all $\ell \geq \kappa$, no proper prefix of length $\ell$ of a codeword appears as a suffix of another codeword (including itself). In other words, for any two codewords $a, b \in \mathcal{C}$, not necessarily distinct, and $\kappa \leq \ell < n$, $a[1, \ell] \neq b[n - \ell + 1, n]$. When $\mathcal{C}$ is 1-WMU, we say that $\mathcal{C}$ is mutually uncorrelated (MU).

**Definition 4.** A code $\mathcal{C}$ of length $n$ is said to avoid primer dimer byproducts of effective length $f$ (f-APD) if the reverse complement and the complement of any substring of length $f$ in a codeword does not appear in as a substring of another codeword (including itself). In other words, for any two codewords $a, b \in \mathcal{C}$, not necessarily distinct, and $1 \leq i, j \leq n - f$, we have $\overline{a}[i, i + f - 1] \neq \overline{b}[j, j + f - 1], b[j + f - 1, j]$.

**Definition 5.** A code $\mathcal{C} \in \mathbb{F}_q^n$ is an $(n, d; \kappa, f)_q$-primer code if the following are satisfied:

(P1) $\mathcal{C}$ is an $(n, d)_q$-code;

(P2) $\mathcal{C}$ is $\kappa$-WMU;

(P3) $\mathcal{C}$ is an $f$-APD code.

Furthermore, if $\mathcal{C}$ is balanced or GC-balanced, then $\mathcal{C}$ is an $(n, d; \kappa, f)_q$-balanced primer code.

Yazdi et al. [3] provided a number of constructions for WMU codes which satisfy some combinations of the constraints (P1), (P2) and (P3). In particular, Yazdi et al. provided the following coupling construction.

**Lemma 6** (Coupling Construction - Yazdi et al. [3]). For $i \in \mathbb{Z}$, let $\mathcal{C}_1$ be an $(n, d_1)_2$-code of size $M_1$. Define the map $\Psi : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \Sigma^n$ such that $\Psi(a, b) = c$ where for $i \in \{n\}$,

$$c_i = \begin{cases} 0, & \text{if } a_ib_i = 00; \\ 1, & \text{if } a_ib_i = 01; \\ \mathbb{G}, & \text{if } a_ib_i = 11. \end{cases}$$

Then the code $\mathcal{C}_e \triangleq \{\Psi(a, b) : a \in \mathcal{C}_1, b \in \mathcal{C}_2\}$ is an $(n, d)_4$-code of size $M_1M_2$, where $d = \min\{d_1, d_2\}$. Furthermore,

(i) if $\mathcal{C}_1$ is balanced, $\mathcal{C}_e$ is GC-balanced;

(ii) if $\mathcal{C}_2$ is $\kappa$-WMU, then $\mathcal{C}_e$ is also $\kappa$-WMU;

(iii) if $\mathcal{C}_2$ is an $f$-APD code, then $\mathcal{C}_e$ is also an $f$-APD code.

Yazdi et al. also provided an iterative construction for primer codes satisfying all the constraints, i.e. balanced primer codes. However, the construction requires a short balanced primer code and a collection of subcodes, some of which are disjoint. Hence, it is unclear whether the code can be constructed efficiently and whether efficient encoding is possible.

In this work, we provide constructions that take cyclic, reversible cyclic or linear codes as inputs and produce primer or balanced primer codes as outputs. Using known families of cyclic codes given by Theorems 1 and 2, we obtain infinite families of primer codes and provide explicit upper bounds on the redundancy.

**D. Our Contributions**

(A) In Section III, we propose efficient methods to construct both balanced and GC-balanced error-correcting codes. Unlike previous methods that require short balanced error-correcting codes, our method uses only cyclic and linear codes as inputs. Furthermore, our method always increases the redundancy only by $\log_2 n + O(1)$ (where $n$ is the block length), regardless of the value of the minimum distance.

(B) In Section IV, we provide three constructions of primer codes. For general parameters, the first construction produces a class of $(n, d; \kappa, f)_q$-balanced primer codes whose redundancy is $(d + 1) \log_q (n + O(1))$, while the other two rely on cyclic codes and use less redundancy albeit for a specific set of parameters. In particular, we have a class of $(n, d; \kappa, \kappa)_q$-balanced primer codes with redundancy $(d + 1) \log_q (n + 1)$. 

92
III. BALANCED ERROR-CORRECTING CODES

The celebrated Knuth’s balancing technique [5] is a linear-time algorithm that maps a binary message of length \( m \) to a balanced word of length \( m + \lceil \log_2 m \rceil \). The method first finds an index \( z \) such that flipping the first \( z \) bits yields a balanced word \( c \). Then a short balanced word \( p \) is appended to represent \( z \). Hence, \( cp \) is the resulting codeword and the redundancy is equal to the length of \( p \) which is \( \lceil \log_2 m \rceil \). The crucial observation demonstrated by Knuth is that such a balancing index \( z \) always exists.

Recently, Weber et al. [14] modified Knuth’s balancing technique to endow the code with error-correcting capabilities. Their method requires two error-correcting codes as inputs: an \((m, d)\) code \( C_m \) and a short \((p, d)\) balanced code \( C_p \), where \( |C_p| \geq m \). Given a message, they first encode it into a codeword \( m \in C_m \). Then they find the balancing index \( z \) of \( m \) and flip the first \( z \) bits to obtain a balanced \( c \). Using \( C_p \), they encode \( z \) into a balanced word \( p \) and the resulting codeword is \( cp \). Since both \( C_m \) and \( C_p \) have distance \( d \), the resulting code has minimum distance \( d \).

Now, this method introduces \( p \) additional redundant bits and since \( p \) is necessarily at least \( d \), the method introduces more than \( \log_2 n \) bits of redundancy when \( d \) is big. Furthermore, the method requires an explicit short balanced code \( C_p \). We overcome this obstacle in our next two constructions. Specifically, Construction A and B require only a cyclic code and two linear codes, respectively. Both constructions do not require short balanced codes and introduces only \( \log_2 n + 1 \) additional bits of redundancy, regardless the value of \( d \).

A. Binary Balanced Error-Correcting Codes

Let \( n \) be odd. In contrast with Knuth’s balancing technique, we always flip the first \((n + 1)/2\) bits of a word \( a \). However, this does not guarantee a balanced word. Nevertheless, if we consider all cyclic shifts of \( a \), i.e. \( \sigma^i(a) \) for \( i \in [n] \), then flipping the first \((n + 1)/2\) bits of one of these shifts must yield a balanced word.

Formally, let \( \phi : \mathbb{F}_2^n \to \mathbb{F}_2^n \) be the map where \( \phi(a) = a + \left(\begin{array}{c} n+1 \end{array}\right)/2(a^{n-1}/2) \). In other words, the map \( \phi \) flips the first \((n + 1)/2\) bits of \( a \). We have the following crucial lemma.

**Lemma 7.** Let \( n \) be odd. For \( a \in \mathbb{F}_2^n \), we can find \( i \in [n] \) such that \( \phi(\sigma^i(a)) \) has weight either \((n-1)/2\) or \((n+1)/2\).

Given a cyclic code \( B \) of length \( n \), we define the following equivalence relation: \( a \sim b \) if and only if \( a = \sigma^i(b) \) for some \( i \in [n] \); and partition the codewords \( B \) into classes. We use \( B/\sim_{cyc} \) to denote a set of representatives.

**Construction A.** Let \( n \) be an odd integer.

**INPUT:** An \([n, k, d]\)-cyclic code \( B \).

**OUTPUT:** A balanced \((n + 1, d')\)-code \( C \) of size at least \( 2^k/n \) where \( d' = 2[d/2] \).

- Let \( u_1, u_2, \ldots, u_m \) be the set of representatives \( B/\sim_{cyc} \).
- For each \( u_i \), find \( j_i \in [n] \) such that \( \phi(\sigma^{j_i}(u_i)) \) has weight \((n-1)/2\) or \((n+1)/2\).
- For \( i \in [m] \), append a check bit to \( \phi(\sigma^{j_i}(u_i)) \) so that its weight is \((n+1)/2\) and denote the modified vector as \( v_i \).

In other words,

\[
\begin{align*}
v_i &= \begin{cases} 
\phi(\sigma^{j_i}(u_i))0, & \text{if } wt(\phi(\sigma^{j_i}(u_i))) = \frac{n-1}{2}; \\
\phi(\sigma^{j_i}(u_i))1, & \text{if } wt(\phi(\sigma^{j_i}(u_i))) = \frac{n+1}{2}; 
\end{cases}
\end{align*}
\]

- Set \( C = \{v_i : 1 \leq i \leq m\} \).

**Theorem 8.** Construction A is correct. In other words, \( C \) is a balanced \((n + 1, 2\lfloor d/2 \rfloor)\)-code of size at least \( 2^k/n \).

Let \( d \) be even and set \( t = d/2 - 1 \). If we apply Construction A to the family of primitive narrow-sense BCH \([n', k, d]\)-cyclic codes, where \( n' = 2^m - 1 = n - 1 \), we obtain a family of balanced codes with redundancy at most \((t + 1)\log_2 n + 1 \).

**Corollary 9.** Let \( d \) be even. There exists a family of \((n, d)\)-balanced codes with redundancy at most \((t + 1)\log_2 n + 1 \), where \( t = d/2 - 1 \).

In contrast, if we apply the technique of Weber et al. [14] to the same family of codes, the balanced \((n, d)\)-codes have redundancy approximately \((t + 1)\log_2 n + (t + 1/2)\log_2 \log_2 n \).

Hence, we reduce the redundancy by \((t + 1/2)\log_2 \log_2 n \) bits.

Finally, we consider the encoding complexity for our construction.

Given a vector \( a \), we can find in linear time the index \( i \) such that \( wt(\phi(\sigma^i(a))) \in \{(n-1)/2, (n+1)/2\} \). Thus, it remains to provide an efficient method to enumerate a set of representatives for the cyclic classes. This problem was solved completely by Tavares et al. [7], [17] and the solution uses the polynomial representation of cyclic codewords. Furthermore, the encoding method can be adapted for constructions in later sections. We review Tavares’ method in detail and discuss our modifications in the full version [8].

B. GC-Balanced Error-Correcting Codes

A direct application of the coupling construction in Lemma 6 and Corollary 9 yields a family of GC-balanced \((n, d)\)-codes with redundancy at most \(d\log_2 n \). However, this construction requires cyclic codes of length \( n - 1 \). The following construction removes the need for cyclic codes.

**Construction B.**

**INPUT:** An \([n + p, n, d]\)-linear code \( A \) and an \((n, d)\)-2-code \( B \) of size \( 2^p n \).

**OUTPUT:** A balanced \((n, d)\)-code \( C \) of size \( 2^p M \).

- Given \( m \in \mathbb{F}_2^n \), let \( j_m \) be the balancing index of \( m \) and \( a_m \) be the balanced word obtained from \( m \) by flipping the first \( j_m \) bits.
- Consider a systematic encoder for \( A \). For \( a_m \in \mathbb{F}_2^n \), let \( a_m p_m \) be the corresponding codeword in \( A \).
- Finally, since \( B \) is of size \( 2^p M \), we may assume without loss of generality an encoder \( \phi_B : [M] \times [n] \times \mathbb{F}_2^n \to B \). We set \( b_{m,i} = \phi_B(i, j_m, p_m) \).
- Set \( C \triangleq \{\Psi(a_m, b_{m,i}) : m \in \mathbb{F}_2^n, i \in [M]\} \).

**Theorem 10.** Construction B is correct. In other words, \( C \) is a GC-balanced \((n, d)\)-code of size \( 2^p M \).

**Corollary 11.** Fix \( d \) and set \( t = [(d - 1)/2] \). There exists an GC-balanced \((n, d)\)-code with redundancy at most \((2t + 1)\log_4 n + 2t \) symbols for sufficiently large \( n \).

IV. ERROR-CORRECTING PRIMER CODES

In this section, we provide three constructions of primer codes: one direct modification of Yazdi et al. that yields primer codes for general parameters and the other two that rely on cyclic codes and have lower redundancy for a specific set of parameters.
A. MU Codes that Avoid Primer Dimer Byproducts

Yazdi et al. [3] constructed a set of mutually uncorrelated primers that avoids primer dimer byproducts.

Definition 12. A code $A$ is $\ell$-APD-constrained if for each $a \in A$,

- $a$ ends with one,
- $a$ contains $0^\ell 1$ as a substring exactly once,
- $a$ does not contain $0^\ell$ as a substring.

Lemma 13 (Yazdi et al. [3, Lemma 5]). Let $n$, $f$, $\ell$, $r$ be positive integers such that $n = rf + \ell + 1$ and $\ell + 3 \leq f$. Suppose that $A$ is an $\ell$-APD-constrained code of length $f$. Then the code $C = \{0^{\ell}1 a_1 a_2 \ldots a_r : a \in A^r\}$ is both MU and $(2f)\text{-APD}$ and its size is $|A|^r$.

The following construction equips the primer code in Lemma 13 with error-correcting capabilities.

Construction C. Let $f$, $r$, $d$ and $\ell$ be positive integers where $\ell + 3 \leq f$ and $p + |p/(\ell - 1)| + 1 \leq f$.

INPUT: An $[rf + p, rf, d, 2]$-linear code $B$ and a binary $\ell$-APD-constrained code $A$ of length $f$.

OUTPUT: A binary $(n, d; 1, 2f)$-primer code $C$ of length $n = rf + p + |p/(\ell - 1)| + \ell + 2$ and size $|A|^r$.

- Consider a systematic encoder for $B$.
- For every message $a \in \mathbb{F}_2^f$, let $ap_a$ be the corresponding codeword in $B$.
- For the vector $pa_a$, we insert an one after every $(\ell - 1)$ bits and append a one. In other words, we insert $\lfloor p/(\ell - 1)\rfloor + 1$ ones and we call the resulting vector $p'_{a_a}$.
- Set $C = \{0^{\ell}p'_{a_a} : a \in A^r\}$.

Next, for fixed values of $r$ and $d$, we describe a family of $(n; d, 1, f)$-primer codes with $n = rf + o(f)$ and redundancy at most $t \log_2 n + O(1)$, where $t = \lfloor (d - 1)/2 \rfloor$. Specifically, we provide the constructions for the input codes $B$ and $A$ in Construction C.

Lemma 14. For $\ell \geq 8$, set $f = 2^{t-4}$. Then there exists an $\ell$-APD-constrained code $A$ of size $(f - \ell - 2)/2 2^{t-4}$ and a linear-time encoding algorithm that maps $[f - \ell - 2] \times \mathbb{F}_2^f$ to $A$.

Hence, for $\ell \geq 8$, we choose $f = 2^{t-4}$. For the input code $B$, we shorten an appropriate BCH code given in Theorem 1 to obtain an $[rf + p, rf, d, 2]$-linear code with redundancy $p \leq \ell \log_2 n + \ell + 1$. Hence, applying Construction C, we obtain a primer code with $(n; d, 1, f)$-primer codes with $n = rf + p + |p/(\ell - 1)| + \ell + 2$. Observe that for sufficiently large $\ell$, we have that $rf < n < (r + 1)f$. After some algebraic manipulations (see [8] for details), we have the following theorem.

Theorem 15. Fix $r$ and $d$ and set $t = \lfloor (d - 1)/2 \rfloor$. Then there exists a family of $(n; d, 1, f)$-primer codes with $n = rf + o(f)$ and redundancy at most $(t + 1) \log_2 n + O(1)$. Furthermore, there exists a linear-time encoding algorithm for these primer codes.

Applying Lemma 6, we obtain primer codes over $\{A, T, C, G\}$.

Corollary 16. Fix $r$ and $d$, and set $t = \lfloor (d - 1)/2 \rfloor$.

(i) There exists a family of $(n; d, 1, f)$-primer codes with $n = rf + o(f)$ and redundancy at most $(2t + 1) \log_4 n + O(1)$.

(ii) There exists a family of balanced $(n; d, 1, f)$-primer codes with $n = rf + o(f)$ and redundancy at most $(d + 1) \log_4 n + O(1)$.

B. Almost GC-Balanced $\kappa$-Mutually Uncorrelated Only

Using cyclic codes and modifying Construction A, we obtain almost balanced primer codes that satisfy conditions (P1) and (P2) only. Here, a code is almost balanced if the weight (or GC-content) of every word belongs to $\{[n/2 - 1, [n/2], [n/2], [n/2] + 1\}$.

Let $n$ be odd and we abuse notation by using $\phi$ to denote the map $\phi : \mathbb{F}_2^n \to \mathbb{F}_4^n$ where $\phi(a) = a + q^{(n+1)/2}(q - 1)/2n$. In other words, $\phi$ switches $A$ with $C$ and $T$ with $G$, and vice versa, in the first $(n + 1)/2$ coordinates of $a$. We have the following analogue of Lemma 7.

Lemma 17. For $a \in \mathbb{F}_2^n$, we can find $i \in [n]$ such that $\phi(\sigma_i(a))$ is GC-balanced.

Construction D. Let $n$ be odd, $k \leq \lfloor (n + 1)/4 \rfloor$ and $q \in \{2, 4\}$.

INPUT: An $[n, k, d|\text{cycl}]$ cyclic code $B$ containing $1^n$.

OUTPUT: An almost balanced $(n, d; k + 1, n)$-primer code $B$ of size at least $q^k/n$.

- Let $u_1, u_2, \ldots, u_m$ be the set of representatives $\forall i \sim \text{cyc}$.
- For each $u_i$, find $j_i \in [n]$ such that $\phi(\sigma_i(u_i))$ is either balanced or GC-balanced.
- Let $\mu = (n - 1)/2$. For each $u_i$, set $v_i = \begin{cases} \sigma_i(u_i) + \omega^{i+1}q^{n-1}, & \text{if } q = 2, \\ \sigma_i(u_i) + \omega^{i+1}q^{n-1}, & \text{if } q = 4. \end{cases}$

Set $\mathcal{C} = \{v_i : 1 \leq i \leq m\}$.

Theorem 18. Construction D is correct, i.e., $\mathcal{C}$ is an almost balanced $(n, d; k + 1, n)$-primer code of size at least $q^k/n$.

C. $\kappa$-Mutually Uncorrelated Codes that Avoid Primer Dimer Byproducts of Length $\kappa$

Using reversible cyclic codes, we further reduce the redundancy for primer codes in the case when $\kappa = f$.

Definition 19. Let $g(X)$ be the generator polynomial of a reversible cyclic code $B$ of length $n$ and dimension $k$ that contains $1^n$. Set $h(X) = (X^n - 1)g(X)$. Then $\{h(X), p_1(X), p_2(X), \ldots, p_P(X)\}$ is a set of polynomials of $(g, k)$-re-generating if the following hold:

(R1) $h(X)$ divides $h(X)$;
(R2) $h(1) \neq 0$;
(R3) $h(X) = X^d h^*(X^{-1}) h^*(0)$, where $d^* = \deg h^*$;
(R4) $h^*(X)$ does not divide $X^i p_i(X) - p_i(X)$ for all $i, j \in [P]$ and $s \in [n - 1]$;
(R5) $h^*(X)$ does not divide $X^i p_i(X) - X^k p_i(X)$ for all $i, j \in [P]$ and $s \leq s \leq n - k$;
(R6) $h^*(X)$ does not divide $X^{s+k-1} p_i(X^{-1}) - p_i(X)$ for all $i, j \in [P]$ and $0 \leq s \leq n - k$.

(S7) $\deg p_i(X) < \deg h^*$ for all $i \in [P]$.

Construction E.

INPUT: An $[n, k, d|\text{cycl}]$-reversible cyclic code $B$ containing $1^n$ with generator polynomial $g(X)$ and a $(g, k)$-re-generating set of polynomials $\{h^*(X), p_1(X), p_2(X), \ldots, p_P(X)\}$.

OUTPUT: An $(n, d; k, k)$-primer code $\mathcal{C}$ of size $q^k P$, where $k^* = k - \deg h^*$.

- Set $\mathcal{C} = \{(m(X) h^*(X) + p_i(X)) g(X) : \deg m \leq k^*, i \in [P]\}$.

Theorem 20. Construction E is correct. In other words, $\mathcal{C}$ is an $(n, d; k, k)$-primer code.

We illustrate Construction E via an example.
Example 21. Set $n = 15$ and $q = 4$. Let $g(x) = x^6 + x^5 + (\omega + 1)x^4 + x^3 + (\omega + 1)x^2 + x + 1$ be the generator polynomial of an $[15, 9, 4]$-reversible cyclic code that contains $1^n$. Consider $h^*(X) = X^4 + \omega X^3 + \omega^2 X^2 + \omega X + 1$ and
\[
p_1 = \omega, \quad p_{10} = \omega^2 x^2 + (\omega + 1)x^2 + x + 1, \\
p_2 = \omega + 1, \quad p_{11} = \omega x^2 + (\omega + 1)x^2 + x + 1, \\
p_3 = 1, \quad p_{12} = \omega x^2 + x^2, \\
p_4 = \omega x + \omega, \quad p_{13} = \omega x^2 + x^2 + x + 1, \\
p_5 = (\omega + 1)x + x + 1, \quad p_{14} = (\omega + 1)x^3 + \omega x^2 + \omega x + \omega, \\
p_6 = x + 1, \quad p_{15} = (\omega + 1)x^3 + \omega x^2 + (\omega + 1)x + x + 1, \\
p_7 = \omega x^2 + \omega x + \omega + 1, \quad p_{16} = (\omega + 1)x^3 + x^2 + x + 1, \\
p_8 = \omega x^2 + (\omega + 1)x^2 + x + 1, \quad p_{17} = \omega^2 x + \omega^2 x^2 + (\omega + 1)x + x + 1, \\
p_9 = \omega x^2 + (\omega + 1)x^2 + x + 1.
\]
We can verify that the set \( \{h^*(X), p_1(X), \ldots, p_{17}(X)\} \) is \((g, 9)\)-rc-generating. Therefore, \( k^* = 15 - 6 - 4 = 5 \) and the size of the \((15, 5, 9, 4)\)-primer code have size \( 17(4^5) \geq 2^{14}\). In contrast, for their experiment, Yazdi et al. constructed a set of weakly mutually uncorrelated primers of length 16, distance four and size four. Specifically, they set \( C_1 = \{01^701^7, 10^710^7\} \) and \( C_2 \) to be an extended BCH \([16, 11, 4]\)-cyclic code. Then they applied the coupling construction to obtain an \((16, 4, 9, 16)\)-primer code of size \( 2^{12}\). Therefore, Construction E provides a larger set of primers using less bases, while improving the minimum distance and avoiding primer dimer products at the same time.

Finally, applying Construction E to the class of reversible cyclic codes in Theorem 2, we obtain a family of primer codes that has efficient encoding algorithms.

Corollary 22. Let \( m \geq 6 \) and \( 1 \leq \tau \leq \lceil m/2 \rceil \). Set \( n = 4^m - 1 \) and \( d = 4^\tau - 1 \). There exists an \((n, d; k, k)\)-primer code of size \( 4k - 2n\), where
\[
k = \begin{cases} 
    n - (d - 3)m, & \text{if } m \text{ is odd and } \tau = \frac{m+1}{2}; \\
    n - (d - 1)m, & \text{otherwise.}
\end{cases}
\]
So, there is a family of \((n, d; k, k)\)-primer codes with \( d \approx \sqrt{n} \), \( k \approx n - \sqrt{n} \log_4 n \), and redundancy at most \((d + 1) \log_4 (n + 1)\).

V. CONCLUSION

We provide efficient and explicit methods to construct balanced codes, primer codes and DNA computing codes with error-correcting capabilities. Using certain classes of BCH codes as inputs, we obtain infinite families of \((n, d)\)-codes satisfying our constraints with redundancy \( C_d \log n + O(1) \). Here, \( C_d \) is a constant dependent only on \( d \) and we provide a summary of our constructions and the corresponding value of \( C_d \) in Table I. Note that in all our constructions, we have \( C_d \leq d + 1 \). On the other hand, the sphere-packing bound requires \( C_d \geq \lceil (d - 1)/2 \rceil \). Therefore, it remains open to provide efficient and explicit constructions that reduce the value of \( C_d \) further.

ACKNOWLEDGMENT

Research of Y. M. Chee, H. M. Kiah and H. Wei is partially funded by the Singapore Ministry of Education under grant MOE2015-T2-2-086.

REFERENCES

[1] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, “Portable and error-free DNA-based data storage,” Scientific reports, vol. 7, no. 1, p. 5011, 2017.
[2] S. Yazdi, H. M. Kiah, E. R. Garza, J. Ma, H. Zhao, and O. Milenkovic, “DNA-based storage: Trends and methods,” IEEE Trans. Molecular, Biological, Multi-Scale Commun., vol. 1, no. 3, pp. 230–248, 2015.
[3] S. Yazdi, H. M. Kiah, R. Gabrys, and O. Milenkovic, “Mutually uncorrelated primers for DNA-based data storage,” IEEE Transactions on Information Theory, vol. 64, no. 9, pp. 6283–6296, 2018.
[4] S. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," Scientific Reports, vol. 5, no. 14138, 2015.
[5] D. Knuth, “Efficient balanced codes,” IEEE Trans. Inform. Theory, vol. 32, no. 1, pp. 51–53, 1986.
[6] J. L. Massey, “Reversible codes,” Inf. Control, vol. 7, no. 3, pp. 369–380, 1964.
[7] S. E. Tavares, P. E. Allard, and S. G. S. Shiva, “On the decomposition of cyclic codes into cyclic classes,” Inf. Control, vol. 18, no. 4, pp. 342–354, 1971.
[8] Y. M. Chee, H. M. Kiah, and H. Wei, “Efficient and explicit balanced primer codes,” arXiv preprint arXiv:1901.01023, 2019.
[9] J. F. MacWilliams and N. J. A. Sloane, The theory of error-correcting codes. Elsevier, 1977.
[10] S. A. Aly, A. Klappenecker, and P. K. Sarvepalli, “On quantum and classical bch codes,” IEEE Transactions on Information Theory, vol. 53, no. 3, pp. 1183–1188, March 2007.
[11] C. Li, C. Ding, and S. Li, “LCD cyclic codes over finite fields,” IEEE Trans. Inform. Theory, vol. 63, no. 7, pp. 4344–4356, 2017.
[12] S. Li, C. Ding, and H. Liu, “Two families of LCD BCH codes,” IEEE Trans. Inform. Theory, vol. 63, no. 9, pp. 5699–5717, 2017.
[13] K. Tzeng and C. Hartmann, “On the minimum distance of certain reversible cyclic codes,” IEEE Trans. Inform. Theory, vol. 16, no. 5, pp. 644–646, 1970.
[14] J. H. Weber, K. A. S. Immink, and H. C. Ferreira, “Error-correcting balanced knuth codes,” IEEE Trans. Inform. Theory, vol. 58, no. 1, pp. 82–89, 2012.
[15] D. Limbachiya, B. Rao, and M. K. Gupta, “The art of DNA strings: sixteen years of DNA coding theory,” arXiv preprint arXiv:1607.00266, 2016.
[16] K. A. S. Immink and K. Cai, “Properties and constructions of constrained codes for dna-based data storage,” arXiv preprint arXiv:1812.06798, 2018.
[17] S. E. Tavares, P. E. Allard, and S. G. S. Shiva, “A note on the decomposition of cyclic codes into cyclic classes,” Inf. Control, vol. 22, no. 1, pp. 100–106, 1975.