On the Design of Codes for DNA Computing*

Olgica Milenkovic[1] and Navin Kashyap[2]

[1] University of Colorado, Boulder, CO 80309, USA.
Email: olgica.milenkovic@colorado.edu
[2] Queen's University, Kingston, ON, K7L 3N6, Canada.
Email: nkashyap@mast.queensu.ca

Abstract. In this paper, we describe a broad class of problems arising in the context of designing codes for DNA computing. We primarily focus on design considerations pertaining to the phenomena of secondary structure formation in single-stranded DNA molecules and non-selective cross-hybridization. Secondary structure formation refers to the tendency of single-stranded DNA sequences to fold back upon themselves, thus becoming inactive in the computation process, while non-selective cross-hybridization refers to unwanted pairing between DNA sequences involved in the computation process. We use the Nussinov-Jacobson algorithm for secondary structure prediction to identify some design criteria that reduce the possibility of secondary structure formation in a codeword. These design criteria can be formulated in terms of constraints on the number of complementary pair matches between a DNA codeword and some of its shifts. We provide a sampling of simple techniques for enumerating and constructing sets of DNA sequences with properties that inhibit non-selective hybridization and secondary structure formation. Novel constructions of such codes include using cyclic reversible extended Goppa codes, generalized Hadamard matrices, and a binary mapping approach. Cyclic code constructions are particularly useful in light of the fact we prove that the presence of a cyclic structure reduces the complexity of testing DNA codes for secondary structure formation.

1 Introduction

The field of DNA-based computation was established in a seminal paper by Adleman [2], in which he described an experiment involving the use of DNA molecules to solve a specific instance of the directed travelling salesman problem. DNA sequences within living cells of eukaryotic species appear in double helices (alternatively, *duplexes*), in which one strand of nucleotides is chemically attached to its complementary strand. However, in DNA-based computation, only relatively short *single-stranded* DNA sequences, referred to as *oligonucleotides*, are used. The computing process simply consists of allowing these oligonucleotide strands

* This work was supported in part by a research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada. Portions of this work were presented at the 2005 IEEE International Symposium on Information Theory (ISIT'05) held in Adelaide, Australia.

to self-assemble to form long DNA molecules via the process of *hybridization*. Hybridization is the process in which oligonucleotides with long regions of complementarity bond with each other. The astounding parallelism of biochemical reactions makes a DNA computer capable of parallel-processing information on an enormously large scale. However, despite its enormous potential, DNA-based computing is unlikely to completely replace electronic computing, due to the inherent unreliability of biochemical reactions, as well as the sheer speed and flexibility of silicon-based devices [30]. Nevertheless, there exist special applications for which they may represent an attractive alternative or the only available option for future development. These include cell-based computation systems for cancer diagnostics and treatment [3], and ultra-high density storage media [17]. Such applications require the design of oligonucleotide sequences that allow for operations to be performed on them with a high degree of reliability.

The process of self-assembly in DNA computing requires the oligonucleotide strands (codewords) participating in the computation to selectively hybridize in a manner compatible with the goals of the computation. If the codewords are not chosen appropriately, unwanted (non-selective) hybridization may occur. For many applications, even more detrimental is the fact that an oligonucleotide sequence may *self-hybridize*, *i.e.*, fold back onto itself, forming a secondary structure which prevents the sequence from participating in the computation process altogether[3]. For example, a large number of read-out failures in the DNA storage system described in [17] was attributed to the formation of *hairpins*, a special secondary structure formed by oligonucleotide sequences. The number of computational errors in a DNA system designed for solving an instance of a 3-SAT problem [5] were reduced by generating DNA sequences that avoid folding and undesired hybridization phenomena. Similar issues were reported in [4], where a DNA-based computer was used for breaking the Digital Encryption Standard.

Even if hybridization can be made error-free and no detrimental folding of sequences occurs, there remain other reliability issues to be dealt with. One such issue is DNA duplex stability [6],[18]: here, a hybridized pair of sequences has to remain in a duplex formation for a sufficiently long period of time in order for the extraction and sequence "sifting" processes to be performed accurately. It was observed in [6] that the stability of duplexes depends on the combinatorial structure of the sequences, more precisely, on the combination of adjacent pairs of bases present in the oligonucleotide strands.

It must be pointed out that the problem of designing sets of codewords that have properties suitable for DNA computing purposes can be considered to be partially solved from the computational point of view. There exist many software packages, such as the Vienna package [29] and the mfold web server [32], that can predict the secondary structure of a single-stranded DNA (or RNA) sequence. But such procedures can often be computationally expensive when large numbers of sequences are sought, or if the sequences are long. Furthermore, they do not

[3] This is not a problem with all DNA-based systems; there exist DNA-based computer logic circuits for which specific folding patterns are actually required by the system architecture itself [25].

provide any insight into the combinatorial nature of the problems at hand. Such insight is extremely valuable from the perspective of functional genomics, for which one of the outstanding principles is that the folding structure of a sequence is closely related to its biological function [7].

Until now, the focus of coding for DNA computing [1],[8],[10],[14],[18],[23] was on constructing large sets of DNA codewords with fixed base frequencies (constant GC-content) and prescribed minimum distance properties. When used in DNA computing experiments, such sets of codewords are expected to lead to very rare hybridization errors. The largest families of linear codes avoiding hybridization errors were described in [10], while bounds on the size of such codes were derived in [18] and [14]. As an example, it was shown in [10] that there exist 94595072 codewords of length 20 with minimum Hamming distance $d = 5$ and with exactly 10 $\mathbf{G}/\mathbf{C}$ bases. In comparison, without disclosing their design methods, Shoemaker *et al.* reported [24] the existence of only 9105 DNA sequences of length 20, at Hamming distance at least 5, free of secondary structure at temperatures of $61 \pm 5\ ^oC$. Since ambient temperature and chemical composition have a significant influence on the secondary structure of oligonucleotides, it is possible that this number is even smaller for other environmental parameters.

The aim of this paper is to provide a broad description of the kinds of problems that arise in coding for DNA computing, and in particular, to stress the fact that DNA code design must take secondary structure considerations into account. We provide the necessary biological background and terminology in Section 2 of the paper. Section 3 contains a detailed description of the secondary structure considerations that must go into the design of DNA codes. By studying the well-known Nussinov-Jacobson algorithm for secondary structure prediction, we show how the presence of a cyclic structure in a DNA code reduces the complexity of the problem of testing the codewords for secondary structure. We also use the algorithm to argue that imposing constraints on the number of complementary base pair matches between a DNA sequence and some of its shifts could inhibit the occurrence of sequence folding. In Section 4, consider the enumeration of sequences satisfying some of these shift constraints. Finally, in Section 5, we provide a sampling of techniques for constructing cyclic DNA codes with properties that are believed to limit non-selective hybridization and/or self-hybridization. Among the many possible approaches for code design, those resulting in large families with simple descriptions are pursued.

2 Background and Notation

We start by introducing some basic definitions and concepts relating to DNA sequences. The oligonucleotide[4] sequences used for DNA computing are *oriented* words over a four-letter alphabet, consisting of four *bases* — two *purines*, adenine (**A**) and guanine (**G**), and two *pyrimidines*, thymine (**T**) and cytosine (**C**). A

[4] Usually, the word 'oligonucleotide' refers to single-stranded nucleotide chains consisting of a few dozen bases; we will however use the same word to refer to single-stranded DNA sequences composed of any number of bases.

DNA strand is oriented due to the asymmetric structure of the sugar-phosphate backbone. It is standard to designate one end of a strand as $3'$ and the other as $5'$, according to the number of the free carbon molecule. Only strands of opposite orientation can hybridize to form a stable duplex. A *DNA code* is simply a set of (oriented) sequences over the alphabet $Q = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$.

Each purine base is the *Watson-Crick complement* of a unique pyrimidine base (and *vice versa*) — adenine and thymine form a complementary pair, as do guanine and cytosine. We describe this using the notation $\overline{\mathbf{A}} = \mathbf{T}$, $\overline{\mathbf{T}} = \mathbf{A}$, $\overline{\mathbf{C}} = \mathbf{G}$, $\overline{\mathbf{G}} = \mathbf{C}$. The chemical ties between the two WC pairs are different — $\mathbf{C}$ and $\mathbf{G}$ pair through three hydrogen bonds, while $\mathbf{A}$ and $\mathbf{T}$ pair through two hydrogen bonds. We will assume that hybridization only occurs between complementary base pairs, although certain semi-stable bonds between mismatched pairs form relatively frequently due to biological mutations.

Let $\mathbf{q} = q_1 q_2 \dots q_n$ be a word of length n over the alphabet Q. For $1 \leq i \leq j \leq n$, we will use the notation $\mathbf{q}_{[i,j]}$ to denote the subsequence $q_i q_{i+1} \dots q_j$. Furthermore, the sequence obtained by reversing $\mathbf{q}$, *i.e.*, the sequence $q_n q_{n-1} \dots q_1$, will be denoted by $\mathbf{q}^R$. The *Watson-Crick complement*, or *reverse-complement*, of $\mathbf{q}$ is defined to be $\mathbf{q}^{RC} = \overline{q_n}\,\overline{q_{n-1}} \dots \overline{q_1}$, where $\overline{q_i}$ denotes the Watson-Crick complement of q_i. For any pair of length-n words $\mathbf{p} = p_1 p_2 \dots p_n$ and $\mathbf{q} = q_1 q_2 \dots q_n$ over the alphabet Q, the Hamming distance $d_H(p, q)$ is defined as usual to be the number of positions i at which $p_i \neq q_i$. We further define the *reverse Hamming distance* between the words $\mathbf{p}$ and $\mathbf{q}$ to be $d_H^R(\mathbf{p}, \mathbf{q}) = d_H(\mathbf{p}, \mathbf{q}^R)$. Similarly, their *reverse-complement Hamming distance* is defined to be $d_H^{RC}(\mathbf{p}, \mathbf{q}) = d_H(\mathbf{p}, \mathbf{q}^{RC})$. For a DNA code $\mathcal{C}$, we define its minimum (Hamming) distance, minimum reverse (Hamming) distance, and minimum reverse-complement (Hamming) distance in the obvious manner:

$$d_H(\mathcal{C}) = \min_{\mathbf{p},\mathbf{q}\in\mathcal{C},\mathbf{p}\neq\mathbf{q}} d_H(\mathbf{p}, \mathbf{q}), \qquad d_H^R(\mathcal{C}) = \min_{\mathbf{p},\mathbf{q}\in\mathcal{C}} d_H^R(\mathbf{p}, \mathbf{q})$$

$$d_H^{RC}(\mathcal{C}) = \min_{\mathbf{p},\mathbf{q}\in\mathcal{C}} d_H^{RC}(\mathbf{p}, \mathbf{q})$$

We also extend the above definitions of sequence complements, reversals, d_H, d_H^R and d_H^{RC} to sequences and codes over an arbitrary alphabet $\mathcal{A}$, for an appropriately defined complementation map from $\mathcal{A}$ onto $\mathcal{A}$. For example, for $\mathcal{A} = \{0, 1\}$, we define complementation as usual via $\overline{0} = 1$ and $\overline{1} = 0$.

Hybridization between a pair of distinct DNA sequences is referred to as *cross-hybridization*, to distinguish it from self-hybridization or sequence folding. The distance measures defined above come into play when evaluating cross-hybridization properties of DNA words under the assumption of a perfectly rigid DNA backbone. As an example, consider two DNA codewords $3' - \mathbf{AAGCTA} - 5'$ and $3' - \mathbf{ATGCTA} - 5'$ at Hamming distance one from each other. For such a pair of codewords, the reverse complement of the first codeword, namely $3' - \mathbf{TAGCTT} - 5'$, will show a very large affinity to hybridize with the second codeword. In order to prevent such a possibility, one could impose a minimum Hamming distance constraint, $d_H(\mathcal{C}) \geq d_{\min}$, for some sufficiently large value of $d_{\min}$. On the other hand, in order to prevent unwanted hybridization between two

DNA codewords, one could try to ensure that the reverse-complement distance between all codewords is larger then a prescribed threshold, i.e. $d^{RC}(\mathcal{C}) \geq d^{RC}_{\min}$. Indeed, if the reverse-complement distance between two codewords is small, as for example in the case of the DNA strands $3' - \mathbf{AAGCTA} - 5'$ and $3' - \mathbf{TACCTT} - 5'$, then there is a good chance that the two strands will hybridize.

Hamming distance is not the only measure that can be used to assess DNA cross-hybridization patterns. For example, if the DNA sugar-phosphate backbone is taken to be a perfectly elastic structure, then it is possible for bases not necessarily at the same position in two strands to pair with each other. Here, it is assumed that bases not necessarily at the same position in two strands can pair with each other. For example, consider the two sequences $3' - \mathbf{A}_1^{(1)}\mathbf{A}_2^{(1)}\mathbf{C}_1^{(1)}\mathbf{C}_2^{(1)}\mathbf{A}_3^{(1)}\mathbf{G}_1^{(1)}\mathbf{A}_4^{(1)}\mathbf{A}_5^{(1)} - 5'$ and $3' - \mathbf{G}_3^{(2)}\mathbf{G}_2^{(2)}\mathbf{T}_3^{(2)}\mathbf{T}_2^{(2)}\mathbf{A}_1^{(2)}\mathbf{G}_2^{(2)}\mathbf{G}_1^{(2)}\mathbf{T}_1^{(2)} - 5'$. Under the "perfectly elastic backbone" model, hybridization between the subsequences of not necessarily consecutive bases, $3' - \mathbf{A}_2^{(1)}\mathbf{C}_1^{(1)}\mathbf{C}_2^{(1)}\mathbf{A}_3^{(1)}\mathbf{A}_4^{(1)} - 5'$ and $5' - \mathbf{T}_1^{(2)}\mathbf{G}_1^{(2)}\mathbf{G}_2^{(2)}\mathbf{T}_2^{(2)}\mathbf{T}_3^{(2)} - 3'$, is plausible. The relevant distance measure for this model is the *Levenshtein distance* [15], which for a pair of sequences $\mathbf{p}$ and $\mathbf{q}$, is defined to be smallest number, $d_L(\mathbf{p}, \mathbf{q})$, of insertions and deletions needed to convert $\mathbf{p}$ to $\mathbf{q}$. A study of DNA codes with respect to this metric can be found in [8]. The recent work of D'yachkov *et al.* [9] considers a distance measure that is a slight variation on the Levenshtein metric, and seems to fit better in the DNA coding context than the Hamming or Levenshtein metrics.

Another important code design consideration linked to the process of oligonucleotide hybridization pertains to the GC-content of sequences in a DNA code. The *GC-content*, $w_{GC}(\mathbf{q})$, of a DNA sequence $\mathbf{q} = q_1 q_2 \ldots q_n$ is defined to be the number of indices i such that $q_i \in \{\mathbf{G}, \mathbf{C}\}$. A DNA code in which all codewords have the same GC-content, w, is called a *constant GC-content code*. The constant GC-content requirement assures similar thermodynamic characteristics for all codewords, and is introduced in order to ensure that all hybridization operations take place in parallel, *i.e.*, roughly at the same time. The GC-content is usually required to be in the range of 30–50% of the length of the code.

One other issue associated with hybridization that we will mention is that of the stability of the resultant DNA duplexes. The duplexes formed during the hybridization phase of the computation process must remain paired for the entire duration of the long "post-processing" phase in which the sequences are extracted and sifted through to determine the result of the computation. As observed in [6], the stability of DNA duplexes depends closely on the sequence of bases in the individual strands; thus, it should be possible to take duplex stability into account while designing DNA codes. We will, however, not touch upon this topic further in this paper.

3 Secondary Structure Considerations

Probably the most important criterion in designing codewords for DNA computing purposes is that the codewords should not form secondary structures that

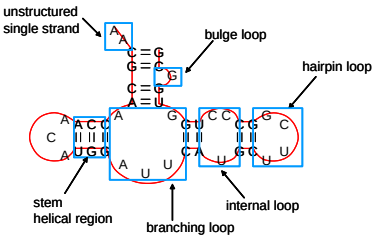


Fig. 1. DNA/RNA secondary structure model (reprinted from [19]).

cause them to become computationally inactive. A secondary structure is formed by a chemically active oligonucleotide sequence folding back onto itself by complementary base pair hybridization. As a consequence of the folding, elaborate spatial structures are formed, the most important components of which are loops (including branching, internal, hairpin and bulge loops), stem helical regions, as well as unstructured single strands[5]. Figure 1 illustrates these structures for an RNA strand[6]. It has been shown experimentally that the most important factors influencing the secondary structure of a DNA sequence are the number of base pairs in stem regions, the number of base pairs in a hairpin loop region as well as the number of unpaired bases.

For a collection of interacting entities, one measure commonly used for assessing the system's property is the free energy. The stability and form of a secondary configuration is usually governed by this energy, the general rule-of-thumb being that a secondary structure minimizes the free energy associated with a DNA sequence. The free energy of a secondary structure is determined by the energy of its constituent pairings, and consequently, its loops. Now, the energy of a pairing depends on the bases involved in the pairing as well as all bases adjacent to it. Adding complication is the fact that in the presence of other neighboring pairings, these energies change according to some nontrivial rules.

Nevertheless, some simple dynamic programming techniques can be used to *approximately* determine base pairings in a secondary structure of a oligonucleotide DNA sequence. Among these techniques, the Nussinov-Jacobson (NJ) folding algorithm [22] is one of the simplest and most widely used schemes.

[5] We do not consider more complicated structures such as the so-called "pseudoknots"; the general problem of determining secondary structure including pseudoknots is known to be NP-complete.

[6] Oligonucleotide DNA sequences are structurally very similar to RNA sequences, which are by their very nature single-stranded, and consist of the same bases as DNA strands, except for thymine being replaced by uracil (**U**).

3.1 The Nussinov-Jacobson Algorithm

The NJ algorithm is based on the assumption that in a DNA sequence $q_1 q_2 \dots q_n$, the energy of interaction, $\alpha(q_i, q_j)$, between the pair of bases (q_i, q_j) is independent of all other base pairs. The interaction energies $\alpha(q_i, q_j)$ are negative quantities whose values usually depend on the actual choice of the base pair (q_i, q_j). One frequently used set of values for RNA sequences is [7]

$$\alpha(q_i, q_j) = \begin{cases} -5 & \text{if } (q_i, q_j) \in \{(\mathbf{G}, \mathbf{C}), (\mathbf{C}, \mathbf{G})\} \\ -4 & \text{if } (q_i, q_j) \in \{(\mathbf{A}, \mathbf{T}), (\mathbf{T}, \mathbf{A})\} \\ -1 & \text{if } (q_i, q_j) \in \{(\mathbf{G}, \mathbf{T}), (\mathbf{T}, \mathbf{G})\}. \end{cases}$$

The value of -1 used for the pairs $(\mathbf{G}, \mathbf{T})$ and $(\mathbf{T}, \mathbf{G})$ indicates a certain frequency of bonding between these mismatched pairs. We will, however, focus our attention only on pairings between Watson-Crick complements. In addition, in order to simplify the discussion, we will restrict our attention to a uniform interaction energy model with $\alpha(q_i, q_j) = -1$ whenever q_i and q_j are Watson-Crick complements and $\alpha(q_i, q_j) = 0$ otherwise.

Let $E_{i,j}$ denote the minimum free energy of the subsequence $q_i \dots q_j$. The independence assumption allows us to compute the minimum free energy of the sequence $q_1 q_2 \dots q_n$ through the recursion

$$E_{i,j} = \min \begin{cases} E_{i+1,j-1} + \alpha(q_i, q_j), \\ \quad E_{i,k-1} + E_{k,j}, \qquad i < k \leq j, \end{cases} \tag{1}$$

where $E_{i,i} = E_{i,i-1} = 0$ for $i = 1, 2, ..., n$. The value of $E_{1,n}$ is the minimum free energy of a secondary structure of $q_1 q_2 \dots q_n$. Note that $E_{1,n} \leq 0$. A large negative value for the free energy, $E_{1,n}$, of a sequence is a good indicator of the presence of a secondary structure in the physical DNA sequence.

The NJ algorithm can be described in terms of free-energy tables, an example of which is shown in Figure 2. In a free-energy table, the entry at position (i, j) (the top left position being (1,1)), contains the value of $E_{i,j}$. The table is filled out by initializing the entries on the main diagonal and on the first lower sub-diagonal of the matrix to zero, and calculating the energy levels according to the recursion in (1). The calculations proceed successively through the upper diagonals: entries at positions $(1, 2), (2, 3), ..., (n-1, n)$ are calculated first, followed by entries at positions $(1, 3), (2, 4), ..., (n-2, n)$, and so on. Note that the entry at $(i, j), j > i$, depends on $\alpha(i, j)$ and the entries at $(i, l), l = i, \dots, j-1, (l, j), l = i+1, \dots, n-1$, and $(i + 1, j - 1)$. The complexity of the NJ algorithm is $O(n^3)$, since each of the $O(n^2)$ entries requires $O(n)$ computations [19].

The minimum-energy secondary structure itself can be found by the *backtracking algorithm* [22] which retraces the steps of the NJ algorithm (for a description of the backtracking algorithm, the reader is referred to [19]). Figure 2 shows the minimum-energy structure of the sequence **GGGAAATCC**, as determined by the backtracking algorithm. The trace-back path through the free-energy table is indicated by the boldface entries in the table.

From a DNA code design point of view, it would be of considerable interest to determine a set of amenable properties that oligonucleotide sequences should

	G	**G**	**G**	**A**	**A**	**A**	**T**	**C**	**C**
G	0	0	0	0	0	0	-1	-2	**-3**
G	0	0	0	0	0	0	-1	-2	**-3**
G	*	0	0	0	0	0	-1	**-2**	-2
A	*	*	0	0	0	0	**-1**	-1	-1
A	*	*	*	0	0	**0**	-1	-1	-1
A	*	*	*	*	0	**0**	-1	-1	-1
T	*	*	*	*	*	0	0	0	0
C	*	*	*	*	*	*	0	0	0
C	*	*	*	*	*	*	*	0	0

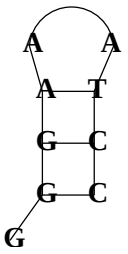


Fig. 2. Free-energy table for the sequence **GGGAAATCC**, along with its secondary structure as obtained by backtracking through the table.

possess so as to either facilitate testing for secondary structure, or exhibit a very low probability for forming such a structure. We next make some straightforward, yet important, observations about the NJ algorithm that provide us with some guidelines for DNA code design.

3.2 Testing for Secondary Structure

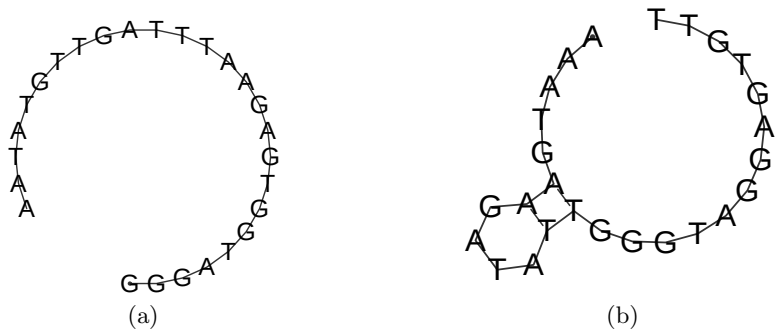
One design principle that arises out of a study of the NJ algorithm is that DNA codes should contain a cyclic structure. The key idea behind this principle is based on the observation that once the free-energy table, and consequently, the minimum free energy of a DNA sequence **q** has been computed, the corresponding computation for any cyclic shift of **q** becomes easy. This idea is summarized in the following proposition.

Proposition 1. *The overall complexity of computing the free-energy tables of a DNA codeword* $q_1 q_2 \dots q_n$ *and* all *of its cyclic shifts is* $O(n^3)$.

Sketch of Proof. It is enough to show that the free-energy table of the cyclic shift $\mathbf{q}^* = q_n q_1 \dots q_{n-1}$ can be obtained from the table of $\mathbf{q} = q_1 \dots q_n$ in $O(n^2)$ steps. The sets of subsequences contained within the positions $1, \dots, n-1$ of **q** and within the positions $2, \dots, n$ of **q*** are the same. This implies that only entries in the first row of the energy table of **q*** have to be computed. Computing each entry in the first row involves $O(n)$ operations, resulting in a total complexity of $O(n^2)$. □

The above result shows that the complexity of testing a DNA code with M length-n codewords for secondary structure is reduced from $O(Mn^3)$ to $O(Mn^2)$, if the code is cyclic. It is also worth pointing out that a cyclic code structure can also simplify the actual production of the DNA sequences that form the code.

Example 1. The minimal free energies of the sequence shown in Figure 3(a) and all its cyclic shifts lie in the range -0.24 to -0.41 kcal/mol. None of these sequences has a secondary structure. On the other hand, for the sequence in Figure 3(b), all its cyclic shifts have a secondary structure, and the minimal free



(a) (b)

Fig. 3. Secondary structures of two DNA codewords at a temperature of 37^oC.

energies are in the range -1.05 to -1.0 kcal/mol. The actual construction of these sequences is described in Example 3 in Section 5.2. Their secondary structures have been determined using the Vienna RNA/DNA secondary structure package [29], which is based on the NJ algorithm, but which uses more accurate values for the parameters $\alpha(q_i, q_j)$, as well as sophisticated prediction methods for base pairing probabilities.

3.3 Avoiding Formation of Secondary Structure

While testing DNA sequences for secondary structure is one aspect of the code design process, it is equally important to know how to design codewords that have a low tendency to form secondary structures. The obvious approach here would be to identify properties of the sequence of bases in an oligonucleotide that would encourage secondary structure formation, so that we could then try to construct codewords which do not have those properties. For example, it seems intuitively clear that if a sequence $\mathbf{q}$ has long, non-overlapping segments $\mathbf{s}_1$ and $\mathbf{s}_2$ such that $\mathbf{s}_1 = \mathbf{s}_2^{RC}$, then there is a good chance that $\mathbf{q}$ will fold to enable $\mathbf{s}_1$ to bind with $\mathbf{s}_2^R$ thus forming a stable structure. Actually, we can slightly strengthen the above condition for folding by requiring that $\mathbf{s}_1$ and $\mathbf{s}_2$ be spaced sufficiently far apart, since a DNA oligonucleotide usually does not make sharp turns, *i.e.*, does not bend over small regions. In any case, the logic is that a sequence that avoids such a scenario should not fold. Unfortunately, this is not quite true: it is not necessarily the longest regions of reverse-complementarity in a sequence that cause a secondary structure to form, as demonstrated by the example in Figure 4. The longest regions of reverse-complementarity in the sequence in the figure are actually the segments of length 7 at either end, which do not actually hybridize with each other within the secondary structure.

A subtler approach to finding properties that inhibit folding consists of identifying components of secondary structures that have a destabilizing effect on

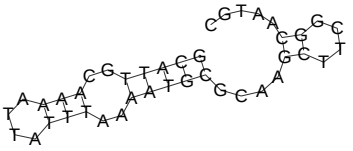


Fig. 4. Secondary structure of the sequence **CGTAA**...**TTACG**.

the structure. Since the DNA sugar-phosphate backbone is a semi-rigid structure, it is reasonable to expect that long loops (especially hairpin loops) tend to destabilize a secondary structure, unless they are held together by an even longer string of stacked base-pairs, which is an unlikely occurrence.

To identify what could induce a hairpin loop to form in a DNA sequence, we enlist the help of the free-energy tables from the NJ algorithm. As an illustrative example, consider the table and corresponding secondary structure in Figure 2. The secondary structure consists of three stacked base-pairs, and a hairpin loop involving two **A**'s. The three stacked base-pairs correspond to the three diagonal steps ($-3 \rightarrow -2$, $-2 \rightarrow -1$ and $-1 \rightarrow 0$) made in the trace-back path indicated by boldface entries in the table; the hairpin loop corresponds to the vertical segment formed by the two 0's in the trace-back path. In general, a vertical segment involving m '0' entries from the first m upper diagonals indicates the presence of a hairpin loop of length m. For sufficiently large m, such a loop would have a destabilizing effect on any nearby stacked base-pairs, leading to an unravelling of the overall structure.

Thus, if the first m upper diagonals of the free-energy table of a DNA sequence $\mathbf{q} = q_1 q_2 \dots q_n$ contain only zero-valued entries, then a hairpin loop of size m is necessarily present in the secondary structure. Consequently, it is very likely that even if base pairing is possible, the overall structure will be unstable[7]. It is easy to verify that the first m upper diagonals in the free-energy table contain only zeros if and only if $\mathbf{q}$ and any of its first $m-1$ shifts contain no complementary base pairs at the same positions, *i.e.*, $q_i \neq \overline{q_{i+j}}$ for $1 \leq j \leq m-1$ and $1 \leq i \leq n-j$.

Relaxing the above argument a little, we see that from the stand-point of designing DNA codewords without secondary structure, it is desirable to have codewords for which the sums of the elements on each of the first few diagonals in

[7] The no sharp turn constraint implies that one can restrict its attention only to the fifth, sixth, ..., m-th upper diagonals, but for reasons of simplicity, we will consider only the previously described scenario.

their free-energy tables are either all zero or of some very small absolute value. This requirement can be rephrased in terms of requiring a DNA sequence to satisfy a "shift property", in which a sequence and its first few shifts have few or no complementary base pairs at the same positions.

In the following section, we define a shift property of a sequence more rigorously, and provide some results on the enumeration of DNA sequences satisfying certain shift properties.

4 Enumerating DNA sequences satisfying a shift property

Recall that for $q \in Q = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$, $\overline{q}$ denotes the Watson-Crick complement of q.

Definition 1. *Given a DNA sequence* $\mathbf{q} = q_1 q_2 \dots q_n$*, we define for* $0 \leq i \leq n-1$*, the* ith matching number*,* $\mu_i(\mathbf{q})$*, of* $\mathbf{q}$ *to be the number of indices* $\ell \in \{1, 2, \dots, n-i\}$ *such that* $q_\ell = \overline{q_{i+\ell}}$*.*

A shift property *of* $\mathbf{q}$ *is any sort of restriction imposed on the matching numbers* $\mu_i(\mathbf{q})$*.*

Enumerating sequences having various types of shift properties is useful because doing so yields upper bounds on the size of DNA codes whose codewords satisfy such properties. We present a few such combinatorial results here.

Given $s \geq 1$, let $g_s(n)$ denote the number of sequences, $\mathbf{q}$, of length n for which $\mu_i(\mathbf{q}) = 0$, $i = 1, ..., s$. For $n \leq s$, we take $g_s(n)$ to be $g_{n-1}(n)$.

Lemma 2. *For all* $n > 1$*,* $g_{n-1}(n) = 4(2^n - 1)$*.*

Proof. It is clear that a DNA sequence is counted by $g_{n-1}(n)$ iff it contains no pair of complementary bases. Such a sequence must be over one of the alphabets $\{\mathbf{A}, \mathbf{G}\}$, $\{\mathbf{A}, \mathbf{C}\}$, $\{\mathbf{T}, \mathbf{G}\}$ and $\{\mathbf{T}, \mathbf{C}\}$. There are $4(2^n - 1)$ such sequences, since there are 2^n sequences over each of these alphabets, of which $\mathbf{A}^n$, $\mathbf{T}^n$, $\mathbf{G}^n$ and $\mathbf{C}^n$ are each counted twice. □

Lemma 3. *For all* $n > s$*,*

$$g_s(n) = 2g_s(n-1) + g_s(n-s).$$

Proof. Let $\mathcal{G}_s(n)$ denote the set of all sequences $\mathbf{q}$ of length n for which $\mu_i(\mathbf{q}) = 0$, $i = 1, ..., s$. Thus, $|\mathcal{G}_s(n)| = g_s(n)$. Note that for any $\mathbf{q} \in \mathcal{G}_s(n)$, $\mathbf{q}_{[n-s,n]}$ cannot contain a complementary pair of bases, and hence cannot contain three distinct bases. Let $\mathcal{E}(n)$ denote the set of sequences $q_1 q_2 \dots q_n \in \mathcal{G}_s(n)$ such that $q_{n-s+1} = q_{n-s+2} = \cdots = q_n$, and let $\mathcal{U}(n) = \mathcal{G}_s(n) \setminus \mathcal{E}(n)$. We thus have $|\mathcal{E}(n)| + |\mathcal{U}(n)| = g_s(n)$. Each sequence in $\mathcal{E}(n)$ is obtained from some sequence $q_1 q_2 \dots q_{n-s+1} \in \mathcal{G}_s(n-s+1)$ by appending $s-1$ bases, $q_{n-s+2}, \dots, q_n$, all equal to q_{n-s+1}. Hence, $|\mathcal{E}(n)| = |\mathcal{G}_s(n-s+1)| = g_s(n-s+1)$, and therefore, $|\mathcal{U}(n)| = g_s(n) - g_s(n-s+1)$.

Now, observe that each sequence $q_1q_2\ldots q_n \in \mathcal{G}_s(n)$ is obtained by appending a single base, q_n, to some sequence $q_1q_2\ldots q_{n-1} \in \mathcal{G}_s(n-1)$. If $q_1q_2\ldots q_{n-1}$ is in fact in $\mathcal{E}(n-1)$, then there are three choices for q_n. Otherwise, if $q_1q_2\ldots q_{n-1} \in \mathcal{U}(n-1)$, there are only two possible choices for q_n. Hence,

$$\begin{aligned} g_s(n) &= 3\,|\mathcal{E}(n-1)| + 2\,|\mathcal{U}(n-1)| \\ &= 3\,g_s(n-s) + 2\,(g_s(n-1) - g_s(n-s)) \end{aligned}$$

This proves the claimed result. □

From Lemmas 2 and 3, we obtain the following result.

Theorem 4. *The generating function* $G_s(z) = \sum_{z=1}^{\infty} g_s(n)z^{-n}$ *is given by*

$$G_s(z) = 4 \cdot \frac{z^{s-1} + z^{z-2} + \cdots + z + 1}{z^s - 2z^{s-1} - 1}.$$

It can be shown that for $s > 1$, the polynomial $\psi_s(z) = z^s - 2z^{s-1} - 1$ in the denominator of $G_s(z)$ has a real root, ρ_s, in the interval (2,3), and $s-1$ other roots within the unit circle. It follows that $g_s(n) \sim \beta_s(\rho_s)^n$ for some constant $\beta_s > 0$. It is easily seen that ρ_s decreases as s increases, and that $\lim_{s\to\infty} \rho_s = 2$.

Theorem 5. *Given an* $s \in \{1, 2, \ldots, n-1\}$*, the number of length-n DNA sequences* $\boldsymbol{q}$ *such that* $\mu_s(\boldsymbol{q}) = m$*, is* $\binom{n-s}{m}4^s3^{n-s-m}$.

Proof. Let $B_s(n, m)$ be the set of length-n DNA sequences $\mathbf{q}$ such that $\mu_s(\mathbf{q}) = m$. A sequence $\mathbf{q} = q_1q_2\ldots q_n$ is in $B_s(n, m)$ iff the set $I = \{i : q_i = \overline{q_{i-s}}\}$ has cardinality m. So, to construct such a sequence, we first arbitrarily pick $q_1, q_2, \ldots, q_s$ and an $I \subset \{s+1, s+2, \ldots, n\}$, $|I| = m$, which can be done in $4^s\binom{n-s}{m}$ ways. The rest of $\mathbf{q}$ is constructed recursively: for $i \geq s+1$, set $q_i = \overline{q_{i-s}}$ if $i \in I$, and pick a $q_i \neq \overline{q_{i-1}}$ if $i \notin I$. Thus, there are 3 choices for each $i \geq s+1$, $i \notin I$, and hence a total of $\binom{n-s}{m}4^s3^{n-s-m}$ sequences $\mathbf{q}$ in $B_s(n, m)$.

The enumeration of DNA sequences satisfying any sort of shift property becomes considerably more difficult if we bring in the additional requirement of constant GC-content. The following result can be proved by applying the powerful Goulden-Jackson method of combinatorial enumeration [11, Section 2.8]. The result is a direct application of Theorem 2.8.6 and Lemma 2.8.10 in [11], and the details of the algebraic manipulations involved are omitted.

Theorem 6. *The number of DNA sequences* $\boldsymbol{q}$ *of length n and GC-content w, such that* $\mu_1(\boldsymbol{q}) = 0$*, is given by the coefficient of* $x^n y^w$ *in the (formal) power series expansion of*

$$\Phi(x, y) = \left(1 - \frac{2x}{1+x} - \frac{2xy}{1+xy}\right)^{-1}.$$

5 Some DNA Code Constructions

Having in previous sections described some of the code design problems in the context of DNA computing, we present some sample solutions in this section. We mainly focus on constructions of cyclic codes, since as mentioned earlier, the presence of a cyclic structure reduces the complexity of testing DNA codes for secondary structure formation, and also simplifies the DNA sequence fabrication procedure. We have seen that other properties desirable in DNA codes include large minimum Hamming distance, large minimum reverse-complement distance, constant GC-content, and the shift properties introduced in Sections 3.3 and 4. The codes presented in this section are constructed in such a way as to possess some subset of these properties. There are many such code constructions possible, so we pick some that are easy to describe and result in sufficiently large codes. Due to the restrictions imposed on the code design methods with respect to testing for secondary structure, the resulting codes are sub-optimal with respect to the codeword cardinality criteria [10].

5.1 DNA Codes from Cyclic Reversible Extended Goppa Codes

The use of reversible cyclic codes for the construction of DNA sequences was previously proposed in [1] and [23]. Here, we will follow a more general approach that allows for the construction of large families of DNA codes with a certain guaranteed minimum distance and minimum reverse-complement distance, based on extended Goppa codes over $GF(2^2)$ [28].

Recall that a code $\mathcal{C}$ is said to be reversible if $\mathbf{c} \in \mathcal{C}$ implies that $\mathbf{c}^R \in \mathcal{C}$ [16, p. 206]. It is a well-known fact that a cyclic code is reversible if and only if its generator polynomial $g(z)$ is self-reciprocal, *i.e.*, $z^{deg(g(z))}g(z^{-1}) = \pm g(z)$. Given an $[n, k, d]$ reversible cyclic code, $\mathcal{C}$, over $GF(2^2)$ with minimum distance d, consider the code $\widehat{\mathcal{C}}$ obtained by first eliminating all the self-reversible codewords (*i.e.*, codewords $\mathbf{c}$ such that $\mathbf{c}^R = \mathbf{c}$), and then choosing one half of the remaining codewords such that no codeword and its reverse are selected simultaneously. If r is the number of self-reversible codewords in $\mathcal{C}$, then $\widehat{\mathcal{C}}$ is a nonlinear code with $(4^k - r)/2$ codewords of length n, and furthermore, $d_H(\widehat{\mathcal{C}}) \geq d$ and $d_H^R(\widehat{\mathcal{C}}) \geq d$. The value of r can be determined easily, as shown below.

Proposition 7. *A reversible cyclic code of dimension k over $GF(q)$ contains $q^{\lceil k/2 \rceil}$ self-reversible codewords.*

Proof. If $\mathbf{a} = a_0 a_1 \ldots a_{n-1}$ is a self-reversible codeword, then the polynomial $a(z) = a_0 + a_1 z + \ldots a_{n-1} z^{n-1}$ is self-reciprocal. Let $g(z)$ be the generator polynomial for the code, so that $a(z) = i_a(z)g(z)$ for some polynomial $i_a(z)$ of degree at most $k-1$. Since $g(z)$ and $a(z)$ are self-reciprocal, so is $i_a(z)$. Hence, $i_a(z)$ is uniquely determined by the coefficients of its $\lceil k/2 \rceil$ least-order terms z^i, $i = 0, 1, \ldots, \lceil k/2 \rceil - 1$, and there are exactly $q^{\lceil k/2 \rceil}$ choices for these coefficients. □

The code $\widehat{\mathcal{C}}$ defined above can be thought of as a DNA code by identifying $GF(2^2)$ with the DNA alphabet $Q = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$. Let $\mathcal{D}$ be the code obtained from $\widehat{\mathcal{C}}$ by means of the following simple modification: for each $\mathbf{c} \in \mathcal{C}$, replace each of the first $\lfloor n/2 \rfloor$ symbols of $\mathbf{c}$ by its Watson-Crick complement. It is clear that $\mathcal{D}$ has the same number of codewords as $\widehat{\mathcal{C}}$, and that $d_H(\mathcal{D}) \geq d$ as well. It can also readily be seen that if n is even, then $d_H^{RC}(\mathcal{D}) = d_H^R(\widehat{\mathcal{C}})$, and if n is odd, then $d_H^{RC}(\mathcal{D})$ may be one less than $d_H^R(\widehat{\mathcal{C}})$. In any case, we have $d_H^{RC}(\mathcal{D}) \geq d-1$.

We apply the above construction to a class of extended Goppa codes that are known to be reversible and cyclic. We first recall the definition of a Goppa code.

Definition 2. *[16, p. 338] Let $\mathcal{L} = \{\alpha_1, ..., \alpha_n\} \subseteq GF(q^m)$, for q a power of a prime and $m, n \in \mathbb{Z}^+$. Let $g(z)$ be a polynomial of degree $\delta < n$ over $GF(q^m)$ such that $g(z)$ has no root in $\mathcal{L}$. The Goppa code, $\Gamma(\mathcal{L})$, consists of all words $(c_1, ..., c_n), c_i \in GF(q)$ such that $\sum_{i=1}^{n} \frac{c_i}{z-\alpha_i} \equiv 0 \mod g(z)$. $\Gamma(\mathcal{L})$ is a code of length n, dimension $k \geq n - m\delta$ and minimum distance $d \geq \delta + 1$.*

The polynomial $g(z)$ in the definition above is referred to as the *Goppa polynomial*. We shall consider Goppa codes derived from Goppa polynomials of the form $g(z) = [(z-\beta_1)(z-\beta_2)]^a$, for some integer a. Two choices for the roots β_1, β_2 and the corresponding *location sets* $\mathcal{L}$ are of interest: (i) $\beta_1, \beta_2 \in GF(q^m)$, $\mathcal{L} = GF(q^m) - \{\beta_1, \beta_2\}$, $n = q^m - 2$; (ii) $\mathcal{L} = GF(q^m)$, with $\beta_1, \beta_2 \in GF(q^{2m})$, such that $\beta_2 = \beta_1^{q^m}$, $\beta_1 = \beta_2^{q^m}$, and $n = q^m$.

It was shown in [28] that for such a choice of $g(z)$ and for an ordering of the location set $\mathcal{L}$ satisfying $\alpha_i + \alpha_{n+1-i} = \beta_1 + \beta_2$, the extended Goppa codes obtained by adding an overall parity check to $\Gamma(\mathcal{L})$ in the above cases are reversible and cyclic. The extended code has the same dimension as $\Gamma(\mathcal{L})$, but the minimum distance is now at least $2a+2$. Applying the DNA code construction described earlier to such a family of extended Goppa codes over $GF(4)$, we obtain the following theorem.

Theorem 8. *For arbitrary positive integers a, m, there exist cyclic DNA codes $\mathcal{D}$ such that $d_H(\mathcal{D}) \geq 2a+2$ and $d_H^{RC}(\mathcal{D}) \geq 2a+1$, having the following parameters:*
(i) length $n = 4^m + 1$, and number of codewords $M \geq \frac{1}{2}(4^{2^{2m}-2ma} - 4^{2^{2m-1}-ma})$;
(ii) length $n = 4^m - 1$, and number of codewords $M \geq \frac{1}{2}(4^{2^{2m}-2(ma+1)} - 4^{2^{2m-1}-(ma+1)})$.

Example 2. Let $\mathcal{L} = GF(2^2)$, with $q = 2^2, m = 1$, and let $\beta_1 = \alpha$, $\beta_2 = \alpha^4$, for a primitive element α of $GF(2^4)$. We take the Goppa polynomial to be $g(z) = (z-\beta_1)(z-\beta_2)$, so that $a = 1$. The extended Goppa code over $GF(2^2)$ obtained from these parameters is a code of length 5, dimension 2 and minimum distance 4.

We list out the elements of $GF(2^2)$ as $\{0, 1, \theta, 1+\theta\}$, and make the identification $0 \leftrightarrow \mathbf{G}$, $1 \leftrightarrow \mathbf{C}$, $\theta \leftrightarrow \mathbf{T}$, $1+\theta \leftrightarrow \mathbf{A}$, The DNA code $\mathcal{D}$ constructed

as outlined in this section has $d_H(\mathcal{D}) = d_H^R(\mathcal{D}) = 4$ and $d_H^{RC}(\mathcal{D}) = 3$, and consists of the following six codewords: **CGTTC**, **CAAAT**, **CTCCA**, **GCCTT**, **GGAGA**, **ACTAA**.

5.2 DNA Codes from Generalized Hadamard Matrices

Hadamard matrices have long been used to construct constant-weight [16, Chap. 2] and constant-composition codes [26]. We continue this tradition by providing constructions of cyclic codes with constant GC-content, and good minimum Hamming and reverse-complement distance properties.

A *generalized Hadamard matrix* $H \equiv H(n, \mathbb{C}_m)$ is an $n \times n$ square matrix with entries taken from the set of mth roots of unity, $\mathbb{C}_m = \{e^{-2\pi i\, \ell/m}, \ell = 0, ..., m-1\}$, that satisfies $HH^* = nI$. Here, I denotes the identity matrix of order n, while * stands for complex-conjugation. We will only concern ourselves with the case $m = p$ for some prime p. A necessary condition for the existence of generalized Hadamard matrices $H(n, \mathbb{C}_p)$ is that $p|n$. The *exponent matrix*, $E(n, \mathbb{Z}_p)$, of $H(n, \mathbb{C}_p)$ is the $n \times n$ matrix with entries in $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$, obtained by replacing each entry $(e^{-2\pi i})^{\ell}$ in $H(n, \mathbb{C}_p)$ by the exponent ℓ.

A generalized Hadamard matrix H is said to be in *standard form* if its first row and column consist of ones only. The $(n-1) \times (n-1)$ square matrix formed by the remaining entries of H is called the *core* of H, and the corresponding submatrix of the exponent matrix E is called the core of E. Clearly, the first row and column of the exponent matrix of a generalized Hadamard matrix in standard form consist of zeros only. It can readily be shown (see *e.g.,* [13]) that the rows of such an exponent matrix must satisfy the following two properties: (i) in each of the nonzero rows of the exponent matrix, each element of $\mathbb{Z}_p$ appears a constant number, n/p, of times; and (ii) the Hamming distance between any two rows is $n(p-1)/p$. We will only consider generalized Hadamard matrices that are in standard form.

Several constructions of generalized Hadamard matrices are known (see [13] and the references therein). A particularly nice general construction is given by the following result from [13].

Theorem 9. [13, Theorem II] *Let $N = p^k - 1$ for p prime and $k \in \mathbb{Z}^+$. Let $g(x) = c_0 + c_1 x + c_2 x^2 + ... + c_{N-k} x^{N-k}$ be a monic polynomial over $\mathbb{Z}_p$, of degree $N-k$, such that $g(x)h(x) = x^N - 1$ over $\mathbb{Z}_p$, for some monic irreducible polynomial $h(x) \in \mathbb{Z}_p[x]$. Suppose that the vector $(0, c_0, c_1, \ldots, c_{N-k}, c_{N-k+1}, \ldots, c_{N-1})$, with $c_i = 0$ for $N-k < i < N$, has the property that it contains each element of $\mathbb{Z}_p$ the same number of times. Then the N cyclic shifts of the vector $\mathbf{g} = (c_0, c_1, \ldots, c_{N-1})$ form the core of the exponent matrix of some Hadamard matrix $H(p^k, \mathbb{C}_p)$.*

Thus, the core of $E \equiv E(p^k, \mathbb{Z}_p)$ (and hence, $H(p^k, \mathbb{C}_p)$) guaranteed by the above theorem is a circulant matrix consisting of all the $N = p^k - 1$ cyclic shifts of its first row. We refer to such a core as a *cyclic core*. Each element of $\mathbb{Z}_p$ appears in each row of E exactly $(N+1)/p = p^{k-1}$ times, and the Hamming

distance between any two rows is exactly $(N+1)(p-1)/p = (p-1)p^{k-1}$. Thus, the N rows of the core of E form a constant-composition code consisting of the N cyclic shifts of some word of length N over the alphabet $\mathbb{Z}_p$, with the Hamming distance between any two codewords being $(p-1)p^{k-1}$.

DNA codes with constant GC-content can obviously be constructed from constant-composition codes over $\mathbb{Z}_p$ by mapping the symbols of $\mathbb{Z}_p$ to the symbols of the DNA alphabet, $Q = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$. For example, using the cyclic constant-composition code of length $3^k - 1$ over $\mathbb{Z}_3$ guaranteed by Theorem 9, and using the mapping that takes 0 to $\mathbf{A}$, 1 to $\mathbf{T}$ and 2 to $\mathbf{G}$, we obtain a DNA code $\mathcal{D}$ with $3^k - 1$ codewords and a GC-content of 3^{k-1}. Clearly, $d_H(\mathcal{D}) = 2 \cdot 3^{k-1}$, and in fact, since $\overline{\mathbf{G}} = \mathbf{C}$ and no codeword in $\mathcal{D}$ contains the symbol $\mathbf{C}$, we also have $d_H^{RC}(\mathcal{D}) \geq 3^{k-1}$. We summarize this in the following corollary to Theorem 9.

Corollary 10. *For any $k \in \mathbb{Z}^+$, there exist DNA codes $\mathcal{D}$ with $3^k - 1$ codewords of length $3^k - 1$, with constant GC-content equal to 3^{k-1}, $d_H(\mathcal{D}) = 2 \cdot 3^{k-1}$, $d_H^{RC}(\mathcal{D}) \geq 3^{k-1}$. and in which each codeword is a cyclic shift of a fixed generator codeword $\mathbf{g}$.*

Example 3. Each of the following vectors generates a cyclic core of a Hadamard matrix [13]:

$$\mathbf{g}^{(1)} = (22201221202001110211210200),$$
$$\mathbf{g}^{(2)} = (20212210222001012112011100).$$

DNA codes can be obtained from such generators by mapping $\{0, 1, 2\}$ onto $\{\mathbf{A}, \mathbf{T}, \mathbf{G}\}$. Although all such mappings yield codes with (essentially) the same parameters, the actual choice of mapping has a strong influence on the secondary structure of the codewords. For example, the codeword in Figure 3(a) was obtained from $\mathbf{g}^{(1)}$ via the mapping $0 \to \mathbf{A},\ 1 \to \mathbf{T},\ 2 \to \mathbf{G}$, while the codeword in Figure 3(b) was obtained from the same generator $\mathbf{g}^{(1)}$ via the mapping $0 \to \mathbf{G},\ 1 \to \mathbf{T},\ 2 \to \mathbf{A}$.

5.3 Code Constructions via a Binary Mapping

The problem of constructing DNA codes with some of the properties desirable for DNA computing can be made into a binary code design problem by mapping the DNA alphabet onto the set of length-two binary words as follows:

$$\mathbf{A} \to 00,\ \ \mathbf{T} \to 01,\ \ \mathbf{C} \to 10,\ \ \mathbf{G} \to 11. \tag{2}$$

The mapping is chosen so that the first bit of the binary image of a base uniquely determines the complementary pair to which it belongs.

Let $\mathbf{q}$ be a DNA sequence. The sequence $b(\mathbf{q})$ obtained by applying coordinatewise to $\mathbf{q}$ the mapping given in (2), will be called the *binary image* of $\mathbf{q}$. If $b(\mathbf{q}) = b_0 b_1 b_2 \ldots b_{2n-1}$, then the subsequence $e(\mathbf{q}) = b_0 b_2 \ldots b_{2n-2}$ will be referred to as the *even subsequence* of $b(\mathbf{q})$, and $o(\mathbf{q}) = b_1 b_3 \ldots b_{2n-1}$ will be called the *odd subsequence* of $b(\mathbf{q})$. Thus, for example, for $\mathbf{q} = \mathbf{ACGTCC}$, we have

$b(\mathbf{q}) = 001011011010$, $e(\mathbf{q}) = 011011$ and $o(\mathbf{q}) = 001100$. Given a DNA code $\mathcal{C}$, we define its *even component* $\mathcal{E}(\mathcal{C}) = \{e(\mathbf{p}) : \mathbf{p} \in \mathcal{C}\}$, and its *odd component* $\mathcal{O}(\mathcal{C}) = \{o(\mathbf{p}) : \mathbf{p} \in \mathcal{C}\}$.

It is clear from the choice of the binary mapping that the GC-content of a DNA sequence $\mathbf{q}$ is equal to the Hamming weight of the binary sequence $e(\mathbf{q})$. Consequently, a DNA code $\mathcal{C}$ is a constant GC-content code if and only if its even component, $\mathcal{E}(\mathcal{C})$, is a constant-weight code. Other properties of a DNA code can also be expressed in terms of properties of its even and code components (for example, see Lemma 11 below). Thus if we have binary codes $\mathcal{B}_1$ and $\mathcal{B}_2$ with suitable properties, then we can construct a good DNA code, whose binary image is equivalent to $\mathcal{B}_1 \times \mathcal{B}_2$, that has $\mathcal{B}_1$ and $\mathcal{B}_2$ as its even and odd components. We present two such constructions here.

Construction B1 Let $\mathcal{B}$ be a binary code consisting of M codewords of length n and minimum distance $d_{\min}$, such that $\mathbf{c} \in \mathcal{B}$ implies that $\overline{\mathbf{c}} \in \mathcal{B}$. For $w > 0$, consider the constant-weight subcode $\mathcal{B}_w = \{\mathbf{u} \in \mathcal{B} : w_H(\mathbf{u}) = w\}$, where $w_H(\cdot)$ denotes Hamming weight. Choose $w > 0$ such that $n \geq 2w + \lceil d_{\min}/2 \rceil$, and consider a DNA code, $\mathcal{C}_w$, with the following choice for its even and odd components:

$$\mathcal{E}_w = \{\mathbf{a}\overline{\mathbf{b}} : \ \mathbf{a}, \mathbf{b} \in \mathcal{B}_w\}, \quad \mathcal{O} = \{\mathbf{a}\mathbf{b}^{RC} : \ \mathbf{a}, \mathbf{b} \in \mathcal{B}, \ \mathbf{a} <_{\text{lex}} \mathbf{b}\},$$

where $<_{\text{lex}}$ denotes lexicographic ordering. The $\mathbf{a} <_{\text{lex}} \mathbf{b}$ in the definition of $\mathcal{O}$ ensures that if $\mathbf{a}\mathbf{b}^{RC} \in \mathcal{O}$, then $\mathbf{b}\mathbf{a}^{RC} \notin \mathcal{O}$, so that distinct codewords in $\mathcal{O}$ cannot be reverse-complements of each other.

The code $\mathcal{E}_w$ has $|\mathcal{B}_w|^2$ codewords of length $2n$ and constant weight n. Furthermore, $d_H(\mathcal{E}_w) \geq d_{\min}$ and $d_H^R(\mathcal{E}_w) \geq d_{\min}$, the first of these inequalities following from the fact that $\mathcal{B}_w$ is a subset of codewords in $\mathcal{B}$. To prove the second inequality, note that for any two distinct codewords $\mathbf{a}\overline{\mathbf{b}}$ and $\mathbf{c}\overline{\mathbf{d}}$, we have

$$d_H(\mathbf{a}\overline{\mathbf{b}}, \mathbf{d}^{RC}\mathbf{c}^R) = d_H(\mathbf{a}, \mathbf{d}^{RC}) + d_H(\overline{\mathbf{b}}, \mathbf{c}^R) = d_H(\mathbf{a}, \mathbf{d}^{RC}) + d_H(\mathbf{c}, \mathbf{b}^{RC}).$$

Since $\mathbf{b}$ and $\mathbf{d}$ both have weight w, it follows that $\mathbf{b}^{RC}$ and $\mathbf{d}^{RC}$ have weight $n - w$. Due to the constraint on the weight w, we have $d_H(\mathbf{a}, \mathbf{d}^{RC}) \geq \lceil d_{\min}/2 \rceil$, and similarly, $d_H(\mathbf{c}, \mathbf{b}^{RC}) \geq \lceil d_{\min}/2 \rceil$. Therefore, for all $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathcal{B}_w$, we must have $d_H(\mathbf{a}\overline{\mathbf{b}}, \mathbf{d}^{RC}\mathbf{c}^R) \geq 2\lceil d_{\min}/2 \rceil \geq d_{\min}$.

The code $\mathcal{O}$ has $M(M-1)/2$ codewords of length $2n$. Clearly, $d_H(\mathcal{O}) \geq d_{\min}$, since the component codewords of $\mathcal{O}$ are taken from $\mathcal{B}$. Similarly, $d_H^{RC}(\mathcal{O}) \geq d_{\min}$, to prove which we only have to observe that for any pair of codewords $\mathbf{a}\mathbf{b}^{RC}$ and $\mathbf{c}\mathbf{d}^{RC}$, $d_H(\mathbf{a}\mathbf{b}^{RC}, \mathbf{d}\mathbf{c}^{RC}) = d_H(\mathbf{a}, \mathbf{d}) + d_H(\mathbf{c}, \mathbf{b}) \geq d_{\min}$.

Therefore, the DNA code

$$\mathcal{C} = \bigcup_{w=d_{\min}}^{w_{\max}} \mathcal{C}_w,$$

with $w_{\max} = (n - \lceil d_{\min}/2 \rceil)/2$, has $\frac{1}{2} M(M-1) \sum_{w=d_{\min}}^{w_{\max}} |A_w|^2$ codewords of length $2n$, and satisfies $d_H(\mathcal{B}) \geq d_{\min}$ and $d_H^{RC}(\mathcal{B}) \geq d_{\min}$.

The following lemma (whose simple proof we omit) records a trivial result that is useful for our next construction. For notational ease, given binary words $\mathbf{x} = (x_i)$ and $\mathbf{y} = (y_i)$, we define $\mathbf{x} \oplus \mathbf{y} = (x_i + y_i)$, the sum being taken modulo-2, and $\mathbf{x} * \mathbf{y} = (x_i y_i)$.

Lemma 11. *Let* $\mathbf{q}$ *be a length-*n *sequence over the DNA alphabet* Q*. For* $i \in \{1, 2, \ldots, n-1\}$*, defining* $\sigma_i = e(\mathbf{q}_{[1,n-i]}) \oplus e(\mathbf{q}_{[i+1,n]})$*, and* $\tau_i = o(\mathbf{q}_{[1,n-i]}) \oplus o(\mathbf{q}_{[i+1,n]})$*, we have*

$$\mu_i(\mathbf{q}) = w_H(\overline{\sigma_i} * \tau_i)$$

where $\overline{\sigma_i}$ *denotes the complement of the binary sequence* σ_i*.*

Construction B2 Let $\mathcal{C}$ be the DNA code obtained by choosing the set of non-zero codewords of a cyclic simplex code of length $n = 2^m - 1$ for both the even and odd code components. Recall that a cyclic simplex code of dimension m is a constant-weight code of length $n = 2^m - 1$ and minimum-distance 2^{m-1}, composed of the all-zeros codeword and the n distinct cyclic shifts of any non-zero codeword [12, Chapter 8]. It is clear that the DNA code $\mathcal{C}$ is a cyclic code contains $(2^m - 1)^2$ codewords of length $2^m - 1$ and GC-content 2^{m-1}.

We claim that $\mathcal{C}$ has the property that for all $i \in \{1, 2, \ldots, n-1\}$ and $\mathbf{q} \in \mathcal{C}$, $\mu_i(\mathbf{q}) \leq 2^{m-2}$. To see this, observe first that for any $\mathbf{q} \in \mathcal{C}$, σ_i and τ_i, defined as in Lemma 11, are just truncations of codewords from the simplex code. Since the simplex code is a constant-weight code, with minimum distance 2^{m-1}, each pair of codewords shares exactly 2^{m-2} positions containing 1's. This implies that for each pair of simplex codewords, there are exactly 2^{m-2} positions in which one codeword contains all 1's, while the other contains all 0's. Such positions are precisely what is counted by $w_H(\overline{\sigma}_i * \tau_i)$ in Lemma 11 which proves our claim.

Example 4. Consider the DNA code resulting from Construction B2 using the cyclic simplex code generated by the codeword 1110100. The DNA code contains 49 codewords of length 7. The minimum Hamming distance of the code is 4, and the codewords all have GC-content equal to 4. A selected subset of codewords from this code is listed below:

TGGCTCA, **TCCGTGA**, **CACGGTC**, **TAGCCTG**,
CATGGCT, **GATCCGT**, **GGGAGAA**, **GGAGAAG**.

The last two codewords consist of the bases **G** and **A** only, and clearly satisfy $\mu_i = 0$ for all $i \leq 7$. On the other hand, for the first three codewords we have $\mu_1 = 1$, while for the next three codewords we see that $\mu_1 = 2$ (meeting the upper bound claimed in the construction). Evaluation of this code using the Vienna secondary structure package [29] shows that none of the 49 codewords exhibits a secondary structure.

As a final remark, we note that the problem of constructing a DNA code that can be efficiently tested for secondary structure using the NJ algorithm can also be reformulated in terms of specifications for the even code component. If the

even component code is cyclic, and each codeword in the even component is combined with codewords from the odd component, then "approximate" testing can be performed in the following manner. The codeword from the even component code $x_1, \ldots, x_n$, $x_i \in \{0, 1\}$ is tested by the NJ algorithm following the steps outlined in Section 3.1, except that the pairing energies are found according to

$$\alpha(x_i, x_j) = \begin{cases} -1 & \text{if } x_i\, x_j \in \{00, 11\} \\ 0 & \text{if } x_i\, x_j \in \{01, 10\}. \end{cases}$$

The result of the NJ algorithm for the even component codeword represents the worst-case scenario for DNA sequence folding. If the free energy of some even component codeword, $\mathbf{b}$, exceeds a certain threshold (which can be determined by a combination of probabilistic and experimental results), all the DNA sequences $\mathbf{q}$ such that $e(\mathbf{q}) = \mathbf{b}$ are subjected to an additional test by the algorithm. If the free energy of $\mathbf{b}$ is below a given threshold, then one can be reasonably sure that none of the DNA sequences $\mathbf{q}$ that have $\mathbf{b}$ as their even subsequence will form a secondary structure.

References

1. T. Abualrub and A. Ghrayeb, "On the construction of cyclic codes for DNA computing," preprint.
2. L.M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, Nov. 1994.
3. Y. Benenson, B. Gil, U. Ben-Dor, R. Adar and E. Shapiro, "An autonomous molecular computer for logical control of gene expression," *Nature*, vol. 429, pp. 423–429, May 2004.
4. D. Boneh, C. Dunworth, and R. Lipton, "Breaking DES using a molecular computer," *Technical Report CS-TR-489-95*, Department of Computer Science, Princeton University, USA, 1995.
5. R.S. Braich, N. Chelyapov, C. Johnson, P.W.K. Rothemund and L. Adleman, "Solution of a 20-variable 3-SAT problem on a DNA computer," *Science*, vol. 296, pp. 492–502, April 2002.
6. K. Breslauer, R. Frank, H. Blocker, and L. Marky, "Predicting DNA duplex stability from the base sequence," *Proc. Natl. Acad. Sci. USA*, vol. 83, pp. 3746–3750, 1986.
7. P. Clote and R. Backofen, *Computational Molecular Biology – An Introduction*, Wiley Series in Mathematical and Computational Biology, New York, 2000.
8. A. D'yachkov, P.L. Erdös, A. Macula, V. Rykov, D. Torney, C-S. Tung, P. Vilenkin and S. White, "Exordium for DNA codes," *J. Comb. Optim.*, vol. 7, no. 4, pp. 369–379, 2003.
9. A. D'yachkov, A. Macula, T. Renz, P. Vilenkin and I. Ismagilov, "New results on DNA codes," *Proc. IEEE Int. Symp. Inform. Theory (ISIT'05)*, Adelaide, Australia, pp. 283–287, Sept. 2005.
10. P. Gaborit and O.D. King, "Linear constructions for DNA codes," *Theoretical Computer Science*, vol. 334, no. 1-3, pp. 99–113, April 2005.
11. I.P. Goulden and D.M. Jackson, *Combinatorial Enumeration*, Dover, 2004.
12. J.I. Hall, Lecture notes on error-control coding, available online at `http://www.mth.msu.edu/~jhall/`.

13. I. Heng and C.H. Cooke, "Polynomial construction of complex Hadamard matrices with cyclic core," *Applied Mathematics Letters*, vol. 12, pp. 87–93, 1999.
14. O.D. King, "Bounds for DNA codes with constant GC-content," *The Electronic Journal of Combinatorics*, vol. 10, no. 1, #R33, 2003.
15. V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Dokl. Akad. Nauk SSSR*, vol. 163, no. 4, pp. 845–848, 1965 (Russian). English translation in *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
16. F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.
17. M. Mansuripur, P.K. Khulbe, S.M. Kuebler, J.W. Perry, M.S. Giridhar and N. Peyghambarian, "Information storage and retrieval using macromolecules as storage media," *University of Arizona Technical Report*, 2003.
18. A. Marathe, A. E. Condon and R. M. Corn, "On combinatorial DNA word design," *J. Comput. Biol.*, vol. 8, pp. 201–219, 2001.
19. S. Mneimneh, "Computational Biology Lecture 20: RNA secondary structures," available online at `engr.smu.edu/~saad/courses/cse8354/lectures/lecture20.pdf`.
20. O. Milenkovic, "Generalized Hamming and coset weight enumerators of isodual codes," accepted for publication in *Designs, Codes and Cryptography*.
21. O. Milenkovic and N. Kashyap, "DNA codes that avoid secondary structures," *Proc. IEEE Int. Symp. Inform. Theory (ISIT'05)*, Adelaide, Australia, pp. 288–292, Sept. 2005.
22. R. Nussinov and A.B. Jacobson, "Fast algorithms for predicting the secondary structure of single stranded RNA ," *Proc. Natl. Acad. Sci. USA*, vol. 77, no. 11, pp. 6309–6313, 1980.
23. V. Rykov, A.J. Macula, D. Torney and P. White, "DNA sequences and quaternary cyclic codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT'01)*, Washington DC, p. 248, June 2001.
24. D.D. Shoemaker, D.A. Lashkari, D. Morris, M. Mittman and R.W. David, "Quantitative phenotye analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy," *Nature Genetics*, vol. 16, pp. 450–456, Dec. 1996.
25. M.N. Stojanovic, D. Stefanovic, "A deoxyribozyme-based molecular automaton," *Nature Biotechnology* vol. 21, pp. 1069–1074, 2003.
26. M. Svanström, P.R.J. Östergard and G.T. Bogdanova, "Bounds and constructions for ternary constant-composition codes," *IEEE Trans. Inform. Theory*, vol. 48, no. 1, pp. 101–111, Jan. 2002.
27. S. Tsaftaris, A. Katsaggelos, T. Pappas and E. Papoutsakis, "DNA computing from a signal processing viewpoint," *IEEE Signal Processing Magazine*, pp. 100–106, Sept. 2004.
28. K.K. Tzeng and K.P. Zimmermann, "On extending Goppa codes to cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 712–716, Nov. 1975.
29. The Vienna RNA Secondary Structure Package, `http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi`.
30. E. Winfree, "DNA computing by self-assembly," *The Bridge*, vol. 33, no. 4, pp. 31–38, 2003. Also available online at `http://www.dna.caltech.edu/Papers/FOE_2003_final.pdf`.
31. D.H. Wood, "Applying error correcting codes to DNA computing," in *Proc. 4th Int. Meeting on DNA Based Computers*, 1998, pp. 109–110.
32. M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction," *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3406–15, 2003. Web access at `http://www.bioinfo.rpi.edu/~zukerm/rna/`.