Writing Information into DNA

Masanori Arita

Department of Computational Biology
Graduate School of Frontier Sciences
University of Tokyo
Kashiwanoha 5-1-5, 277-8561 Kashiwa, Japan
arita@k.u-tokyo.ac.jp

Abstract. The time is approaching when information can be written into DNA. This tutorial work surveys the methods for designing code words using DNA, and proposes a simple code that avoids unwanted hybridization in the presence of shift and concatenation of DNA words and their complements.

1 Introduction

As bio- and nano-technology advances, the demand for writing information into DNA increases. Areas of immediate application are:

- *DNA computation* which attempts to realize biological mathematics, i.e., solving mathematical problems by applying experimental methods in molecular biology [1]. Because a problem must be first encoded in DNA terms, the method of encoding is of crucial importance. Typically, a set of fixed-length oligonucleotides is used to denote logical variables or graph components.
- *DNA tag/antitag system* which designs fixed-length short oligonucleotide tags for identifying biomolecules (e.g., cDNA), used primarily for monitoring gene expressions [2,3,4].
- *DNA data storage* which advocates the use of bacterial DNA as a long-lasting high-density data storage, which can also be resistant to radiation [5].
- *DNA signature* which is important for registering a copyright of engineered bacterial and viral genomes. Steganography (an invisible signature hidden in other information) is useful for the exchange of engineered genomes among developers.

These fields are unlike conventional biotechnologies in that they attempt to *encode artificial information into DNA*. They can be referred to as 'encoding models'. Although various design strategies for DNA sequences have been proposed and some have been demonstrated, no standard code like the ASCII code exists for these models, presumably because data transfer in the form of DNA has not been a topic of research. In addition, requirements for DNA sequences differ for each encoding model.

In this tutorial work, the design of DNA words as information carriers is surveyed and a simple, general code for writing information into biopolymers is

N. Jonoska et al. (Eds.): Molecular Computing (Head Festschrift), LNCS 2950, pp. 23–35, 2004.

proposed. After this introduction, Section 2 introduces major constraints considered in the word design. In Section 3, three major design approaches are briefly overviewed and our approach is described in Section 4. Finally, Section 5 shows an exemplary construction of DNA code words using our method.

2 Requirements for a DNA Code

DNA sequences consist of four nucleotide bases (A: adenine, C: cytosine, G: guanine, and T: thymine), and are arrayed between chemically distinct terminals known as the 5'- and 3'-end. The double-helix DNA strands are formed by a sequence and its complement. The complementary strand, or *complement*, is obtained by the substitution of base A with base T, and base C with base G and vice versa, and reversing its direction. For example, the sequence `5'-AAGCGCTT-3'` is the complement of itself: $\begin{matrix}\texttt{5}' - \texttt{AAGCGCTT} - \texttt{3}' \\ \texttt{3}' - \texttt{TTCGCGAA} - \texttt{5}'\end{matrix}$. A non-complementary base in a double strand cannot form stable hydrogen bonds and is called a (base) *mismatch*. The stability of a DNA double helix depends on the number and distribution of base mismatches [6].

Now consider a set of DNA words for information interchange. Each word must be as distinct as possible so that no words will induce unwanted hybridization (*mishybridization*) regardless of their arrangement. At the same time, all words must be physico-chemically uniform (*concerted*) to guarantee an unbiased reaction in biological experiments.

In principle, there are two measures for evaluating the quality of designed DNA words: statistical thermodynamics and combinatorics. Although the thermodynamic method may yield a more accurate estimation, its computational cost is high. Therefore, since combinatorial estimations approximate the thermodynamic ones, the focus in this work is on the former method, described in terms of discrete constraints that DNA words should satisfy. In what follows, formal requirements for the DNA word set will be introduced.

2.1 Constraints on Sequences

DNA words are assumed to be of equal length. This assumption holds true in most encoding models. (Some models use oligonucleotides of different lengths for spacer- or marker sequences. As such modifications do not change the nature of the problem, their details are not discussed here.) The design problem posed by DNA words has much in common with the construction of classical error-correcting code words.

Let $x = x_1 x_2 \cdots x_n$ be a DNA word over four bases {A,C,G,T}. The *reverse* of x is denoted $x^R = x_n x_{n-1} \cdots x_1$, and the *complement* of x, obtained by replacing base A with T, and base C with G in x and vice versa, is denoted x^C. The *Hamming distance* $H(x, y)$ between two words $x = x_1 x_2 \ldots x_n$ and $y = y_1 y_2 \ldots y_n$ is the number of indices i such that $x_i \neq y_i$. For a set of DNA words S, S^{RC} is its complementation with reverse complement sequences, i.e., $\{x \mid x \in S \text{ or } (x^R)^C \in S\}$.

Hamming Constraints As in code theory, designed DNA words should keep a large Hamming distance between all word pairs. What makes the DNA code-design more complicated than the standard theory of error-correcting code is that we must consider not only $H(x, y)$ but also $H(x^C, y^R)$ to guarantee the mismatches in the hybridization with other words and their complements (Fig 1).

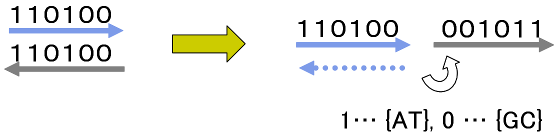

Fig. 1. Binary Pattern of Hybridization. The complementary strand has a reverse pattern of {A,T} and {G,C} bases. A reverse complement of a DNA word corresponds to its complementary strand.

Comma-Free Constraints It is desirable for the designed words to be comma-free because DNA has no fixed reading frames. By definition, a code S is *comma-free* if the overlap of any two, not necessarily different, code words $x_1x_2 \cdots x_n \in S$ and $y_1y_2 \cdots y_n \in S$, (i.e., $x_{r+1}x_{r+2} \cdots x_ny_1y_2 \cdots y_r$; $0 < r < n$) does not result in another code word in S [7,8]. The property by which any overlapping word differs from another word in at least d positions is called *comma-free with index d*. Thus, our DNA code should be comma-free with a high index. [1]

Note that comma-freeness is not replaced by introducing predefined 'spacer' words between code words. Such spacers may facilitate the decoding of words, but they do not contribute to the avoidance of mishybridization. Moreover, spacers lengthen the encoded DNA and lower its information content.

Energy Constraints In addition to the above constraints on mismatches, the melting temperatures of DNA words must be very similar to guarantee their concerted behavior *in vitro*. The most reliable estimation is the nearest neighbor approximation, where the temperature is computed from the frequency of 16 base dimers (from AA to TT) [12,6]. Arita and Kobayashi proposed its further approximation by grouping [GC] and [AT], where the temperature depends on the frequency of only 3 patterns ([GC][GC], [GC][AT] or [AT][GC], and [AT][AT]) [13]. *Dimer frequency* of a sequence x is the three tuple of integers, each describing the frequency of the above 3 patterns in this order. To integrate the terminal

[1] The idea of comma-freeness originated in the elucidation of DNA translation mechanism. Early on, DNA codons for 20 amino acids were thought to be encoded in the comma-free manner [9]. Incidentally, the number of comma-free code words of length 3 over 4 bases is at most 20. The systematic design of a comma-free code of index 1 was soon proposed [10,11].

bases, we assume as if x is cyclic in the computation of frequency. For example, `AAGCGCTT` and `TACGGCAT` exhibit close melting temperatures because they share the same dimer frequency $(3, 2, 3)$. Thus, all DNA code words should share the same dimer frequency to guarantee their concerted behavior.

Other Constraints Depending on the model used, there are constraints in terms of base mismatches. We focus on the first 2 constraints in this paper.

1. *Forbidden subwords* that correspond to restriction sites, simple repeats, or other biological signal sequences, should not appear anywhere in the designed words and their concatenations. This constraint arises when the encoding model uses pre-determined sequences such as genomic DNA or restriction sites for enzymes.
2. *Any subword of length k* should not appear more than once in the designed words. This constraint is imposed to ensure the avoidance of base pair nucleation that leads to mishybridization. The number k is usually ≥ 6.
3. *A secondary structure* that impedes expected hybridization of DNA words should not arise. To find an optimal structure for these words, the minimum free energy of the strand is computed by dynamic programming [14]. However, the requirement here is that the words do not form some structure. This constraint arises when temperature control is important in the encoding models.
4. *Only three bases*, A,C, and T, may be used in the word design. This constraint serves primarily to reduce the number of mismatches by biasing the base composition, and to eliminate G-stacking energy [15]. In RNA word design, this constraint is important because in RNA, both G-C pairs and G-U pairs (equivalent to G-T in DNA) form stably.

2.2 Data Storage Style

Because there is no standard DNA code, it may seem premature to discuss methods of aligning words or their storage, i.e., their data-addressing style. However, it is worth noting that the storage style depends on the word design; the immobilization technique, like DNA chips, has been popular partly because its weaker constraint on words alleviates design problems encountered in scaling up experiments.

Surface-Based Approach In the surface-based (or solid-phase) approach, DNA words are placed on a solid support (Fig 2). This method has two advantages: (1) since one strand of the double helix is immobilized, code words can be separated (washed out) from their complements, thereby reducing the risk of unexpected aggregation of words [16]; (2) since fluorescent labeling is effective, it is easier to recognize words, e.g., for information readout.

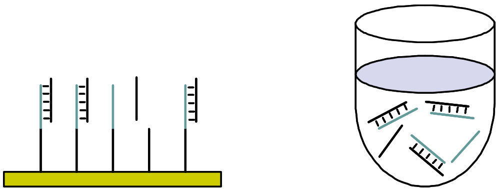

Fig. 2. The Surface-Based versus the Soluble Approach. While they are indistinguishable in solution, immobilization makes it easy to separate information words (gray) from their complements (black).

Soluble Approach Easier access to information on surfaces simultaneously limits the innate abilities of biomolecules. DNA fragments in solution acquire more flexibility as information carriers, and have been shown capable of simulating cellular automata [17]. Other advantages of the soluble approach are: (1) it opens the possibility of autonomous information processing [18]; (2) it is possible to introduce DNA words into microbes. The words can also be used for nano structure design.

Any systematic word design that avoids mishybridization should serve both approaches. Therefore, word constraints must extend to complements of code words. Our design problem is then summarized as follows.

> **P**roblem: Given two integers l and d $(l > d > 0)$, design a set S of length-l DNA words such that S^{RC} is comma-free with index d and for any two sequences $x, y \in S^{RC}$, $H(x,y) \geq d$ and $H(x^C, y^R) \geq d$. Moreover, all words in S^{RC} share the same dimer frequency.

3 Previous Works

Due to the different constraints, there is currently no standard method for designing DNA code words. In this section, three basic approaches are introduced: (1) the template-map strategy, (2) De Bruijn construction, and (3) the stochastic method.

3.1 Template-Map Strategy

This simple yet powerful construction was apparently first proposed by Condon's group [16]. Constraints on the DNA code are divided and separately assigned to two binary codes, e.g., one specifies the GC content (called *templates*), the other specifies mismatches between any word pairs (called *maps*). The product of two codes produces a quaternary code with the properties of both codes (Fig 3). Frutos et al. designed 108 words of length 8 where (1) each word has four GCs; (2) each pair of words, including reverse complements, differs in at least four bases [16]. Later, Li et al., who used the Hadamard code, generalized this construction to longer code words that have mismatches equal to or exceeding half their length [19]. They presented, as an example, the construction of 528 words of length 12 with 6 minimum mismatches.

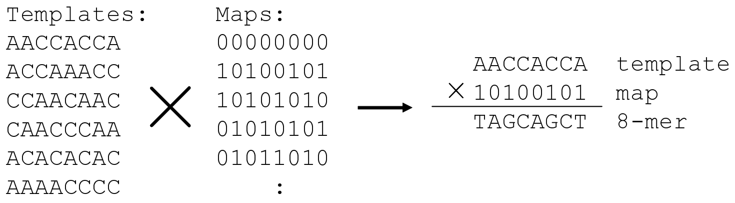

Fig. 3. Template-Map Strategy. In this figure, templates specify that the sequences contain 50% GCs and four mismatches between them and their complements. Maps are error-correcting code words and specify the choice between A and T, or G and C.

The drawback of this construction is twofold. First, the melting temperatures of the designed quaternary words may differ regardless of their uniform GC content. This property was analyzed in Li et al. and the predicted melting temperatures of the 528 words differed over 20 °C range [19]. The second problem is the comma-freeness. Although the design has been effectively demonstrated in the surface-based approach, scaling up to multiple words will be difficult due to mishybridization.

3.2 De Bruijn Construction

The longer a consecutive run of matched base pairs, the higher is the risk of mishybridization. The length-k subword constraint to avoid mishybridization is satisfied with a binary De Bruijn sequence of order k, a circular sequence of length 2^k in which each subword of length k occurs exactly once. [2] A linear time algorithm for the construction of a De Bruijn sequence is known [20]. Ben-Dor et al. showed an optimal choosing algorithm of oligonucleotide tags that satisfy the length-k subword constraint and also share similar melting temperatures [4].

One disadvantage is that the number of mismatches between words may be small, because the length-k constraint guarantees only one mismatch for each k-mer. Another disadvantage is again the comma-freeness.

3.3 Stochastic Method

The stochastic method is the most widely used approach in word design; there are as many types of design software as there are reported experiments.

[2] De Bruijn sequence can be quaternary. By using the binary version, however, we can easily satisfy the constraint that the subword does not occur in the complementary strand.

Deaton et al. used genetic algorithms to find code words of similar melting temperatures that satisfy the ‘extended’ Hamming constraint, i.e., a constraint where mismatches in the case of shift are also considered [21]. (The constraint they named the H-measure, is different from comma-freeness in that it considers mismatches between two words, not their overlaps.) Due to the complexity of the problem, they reported that genetic algorithms can be applied to code words of up to length 25 [22].

Landweber et al. used a random word-generation program to design two sets of 10 words of length 15 that satisfy the conditions (1) no more than five matches over a 20-nucleotide window in any concatenation between all 2^{10} combinations; (2) similar melting temperatures of 45 °C; (3) avoidance of secondary structures; and (4) no consecutive matches of more than 7 base pairs. [3] All of the strong constraints could be satisfied with only 3 bases [15]. Other groups that employed three-base words likewise used random word-generation for their word design [24,23].

Although no detailed analyses for such algorithms are available, the power of stochastic search is evident in the work of Tulpan et al., who could increase the number of code words designed by the template-map strategy [25]. However, they reported that the stochastic search failed to outperform the template-map strategy if searches were started from scratch. Therefore it is preferable to apply the stochastic method to enlarge already designed word sets.

4 Methods

4.1 Comma-Free Error-Correcting DNA Code

Among the different constraints on DNA code words, the most difficult to satisfy is comma-freeness; no systematic construction is known for a comma-free code of high index. The stochastic search is also not applicable because its computational cost is too high.

The comma-free property is, however, a necessary condition for the design of a general-purpose DNA code. This section presents the construction method for a comma-free error-correcting DNA code, and proposes a DNA code: 112 words of length 12 that mismatch at at least 4 positions in any mishybridization, share no more than 6 consecutive subsequences, and retain similar melting temperatures.

Basic Design For this design, we employed the method of Arita and Kobayashi [13]. It can systematically generate a set of words of length ℓ such that any of its members will have approximately $\ell/3$ mismatches with other words, their complements, and overlaps of their concatenations. They constructed sequences as a product of two types of binary words as in the template-map strategy, except that they used a single binary word, denoted T, as the template. Template T is

[3] The fourth condition is unnecessary if the first one is satisfied; presented here are all conditions considered in the original paper.

chosen so that its alignment with subsequent patterns always contains equal to or more than d mismatches.

$$T^R \quad TT^R \quad T^RT \quad TT \quad T^RT^R \tag{1}$$

The template specifies the GC positions of the designed words: [GC] corresponds to either 1's or 0's in the template. Since the pattern T^R specifies the AT/GC pattern of reverse complements, the mismatches between T and T^R guarantee the base mismatches between forward strands and reverse strands of designed DNAs. Other patterns from TT to T^RT^R are responsible for shifted patterns.

For the map words, any binary error-correcting code of minimum distance d or greater is used. Then, any pair of words in the resulting quaternary code induces at least d mismatches without block shift because of the error-correcting code, and with block shift or reversal because of the chosen template.

Comma-freeness is not the only advantage of their method. Because a single template is used to specify GC positions for all words, the GC arrangement of resulting code words is uniform, resulting in similar melting temperatures for all words in the nearest neighbor approximation [13].

Other Constraints In this subsection, methods to satisfy other practical constraints are introduced.

Forbidden subword
Since the error-correcting property of the map words is invariant under exchanging and 0-1 flipping columns of all words, this constraint can be easily satisfied.

Length-k subword
For the DNA words to satisfy this constraint, two additional conditions are necessary: First, the template should not share any length-k subword with patterns in (1). Second, the map words should not share any length-k subword among them.

The first condition can be imposed when the template is selected. To satisfy the second condition, the obvious transformation from word selection to the Max Clique Problem is used: the nodes correspond to the words, and the edges are linked only when two words do not share any length-k subword (without block shift). Note that the clique size is upper bounded by 2^k.

Secondary structure
Since all words are derived from the same template, in the absence of shifts, the number of mismatches can be the minimum distance of the error-correcting code words. Hybridization is therefore more likely to proceed without shifts. To avoid secondary structures, the minimum distance of the error-correcting code words is kept sufficiently large and base mismatches are as much distributed as possible. The latter constraint is already achieved by imposing the length-k subword constraint.

5 Results

5.1 DNA Code for the English Alphabet

Consider the design for the English alphabet using DNA. For each letter, one DNA word is required. One short error-correcting code is the nonlinear (12,144,4) code [26]. [4] Using a Max Clique Problem solver [5], 32, 56, and 104 words could be chosen that satisfied the length-6, -7, -8 subword constraint, respectively.

There are 74 template words of length 12 and of minimum distance 4; they are shown in the Appendix. Since 128 words cannot be derived from a single template under the subword constraint, two words, say S and T, were selected from the 74 templates such that both S and T induce more than 3 mismatches with any concatenation of 4 words T, S, T^R, and S^R (16 patterns), and each chosen word shares no more than length-5, -6, or -7 subword with the other and with their concatenations. Under the length-6 subword constraint, no template pair could satisfy all constraints. Under the length-7, and -8 subword constraints, 8 pairs were selected. (See the Appendix.) All pairs had the common dimer frequency. Under this condition, DNA words derived from these templates can be shown to share close melting temperatures.

Thus, we found 2 templates could be used simultaneously in the design of length-12 words. There were 8 candidate pairs. By combining one of 8 pairs with the 56 words in the Appendix, 112 words were obtained that satisfied the following conditions:

- They mismatched in at least 4 positions between any pair of words and their complements.
- The 4 mismatch was guaranteed under any shift and concatenation with themselves and their complements (comma-freeness of index 4).
- None shared a subword of length 7 or longer under any shift and concatenation.
- All words had close melting temperatures in the nearest neighbor approximation.
- Because all words were derived from only two templates, the occurrence of specific subsequences could be easily located. In addition, the avoidance of specific subsequences was also easy.

We consider that the 112 words serve as the standard code for the English alphabet. The number of words, 112, falls short of the 128 ASCII characters. However, some characters are usually unused. For example, HTML characters from &#14 to &#31 are not used. Therefore, the 112 words suffice for the DNA ASCII code. This is preferable to loosening the constraints to obtain 128 words.

[4] The notation (12,144,4) reads ‘a length-12 code of 144 words with the minimum distance 4’ (one error-correcting).

[5] http://rtm.science.unitn.it/intertools/

5.2 Discussion

The current status of information-encoding models was reviewed and the necessity and difficulty of constructing comma-free DNA code words was discussed. The proposed design method can provide 112 DNA words of length 12 and comma-free index 4. This result is superior to the current standard because it is the only work that considers arbitrary concatenation among words including their complementary strands.

In analyzing the encoding models, error and efficiency must be clearly distinguished. Error refers to the impairment of encoded information due to experimental missteps such as unexpected polymerization or excision. Efficiency refers to the processing speed, not the accuracy, of experiments.

Viewed in this light, the proposed DNA code effectively minimizes errors: First, the unexpected polymerization does not occur because all words satisfy the length-7 subword constraint. [6] Second, the site of possible excision under the application of enzymes is easily identified. Lastly, all words have uniform physico-chemical properties and their interaction is expected to be in concert. The efficiency, on the other hand, remains to be improved. It can be argued that 4 mismatches for words of length 12 are insufficient for avoiding unexpected secondary structures. Indispensable laboratory experiments are underway and confirmation of the applicability of the code presented here to any of the encoding models is awaited.

Regarding code size, it is likely that the number of words can be increased by a stochastic search.

Without systematic construction, however, the resulting code loses one good property, i.e., the easy location of specific subsequences under any concatenation.

The error-correcting comma-free property of the current DNA words opens a way to new biotechnologies. Important challenges include: 1. The design of a comma-free quaternary code of high indices; 2. Analysis of the distribution of mismatches in error-correcting code words; and 3. The development of criteria to avoid the formation of secondary structures.

Also important is the development of an experimental means to realize 'DNA signature'. Its presence may forestall and resolve lawsuits on the copyright of engineered genomes. Currently when a DNA message is introduced into a genome, no convenient method exists for the detection of its presence unless the message sequence is known. In the future, it should be possible to include English messages, not ACGTs, on the input window of DNA synthesizers.

References

1. L.M. Adleman, "Molecular Computation of Solutions to Combinatorial Problems," *Science* **266**(5187), 1021–1024 (1994).
2. S. Brenner and R.A. Lerner, "Encoded Combinatorial Chemistry," *Proc. Nation. Acad. Sci. USA* **89**(12), 5381–5383 (1992).

[6] The minimum length for primers to initiate polymerization is usually considered to be 8.

3. S. Brenner, S.R. Williams, E.H. Vermaas, T. Storck, K. Moon, C. McCollum, J.I. Mao, S. Luo, J.J. Kirchner, S. Eletr, R.B. DuBridge, T. Burcham and G. Albrecht, “In Vitro Cloning of Complex Mixtures of DNA on Microbeads: physical separation of differentially expressed cDNAs,” *Proc. Nation. Acad. Sci. USA* **97**(4), 1665–1670 (2000).
4. A. Ben-Dor, R. Karp, B. Schwikowski and Z. Yakhini, “Universal DNA Tag Systems: a combinatorial design scheme,” *J. Comput. Biol.* **7**(3-4), 503–519 (2000).
5. P.C. Wong, K-K. Wong and H. Foote, “Organic Data Memory Using the DNA Approach,” *Comm. of ACM*, **46**(1), 95–98 (2003).
6. H.T. Allawi and J. SantaLucia Jr., “Nearest-neighbor Thermodynamics of Internal AC Mismatches in DNA: sequence dependence and pH effects,” *Biochemistry*, **37**(26), 9435–9444 (1998).
7. S.W. Golomb, B. Gordon and L.R. Welch, “Comma-Free Codes,” *Canadian J. of Math.* **10**, 202–209 (1958).
8. B. Tang, S.W. Golomb and R.L. Graham, “A New Result on Comma-Free Codes of Even Word-Length,” *Canadian J. of Math.* **39**(3), 513–526 (1987).
9. H.F. Judson, *The Eighth Day of Creation: Makers of the Revolution in Biology,* Cold Spring Harbor Laboratory, (Original 1979; Expanded Edition 1996)
10. J.J. Stiffler, “Comma-Free Error-Correcting Codes,” *IEEE Trans. on Inform. Theor.*, **IT-11**, 107–112 (1965).
11. J.J. Stiffler, *Theory of Synchronous Communication,* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
12. K.J. Breslauer, R. Frank, H. Blocker and L.A. Marky, ”Predicting DNA Duplex Stability from the Base Sequence,” *Proc. Nation. Acad. Sci. USA* **83**(11), 3746–3750 (1986).
13. M. Arita and S. Kobayashi, “DNA Sequence Design Using Templates,” *New Generation Comput.* **20**(3), 263–277 (2002). (Available as a sample paper at http://www.ohmsha.co.jp/ngc/index.htm.)
14. M. Zuker and P. Steigler, “Optimal Computer Folding of Large RNA Sequences Using Thermodynamics and Auxiliary Information,” *Nucleic Acids Res.* **9**, 133–148 (1981).
15. D. Faulhammer, A.R. Cukras, R.J. Lipton and L.F. Landweber, “Molecular Computation: RNA Solutions to Chess Problems,” *Proc. Nation. Acad. Sci. USA* **97**(4), 1385–1389 (2000).
16. A.G. Frutos, Q. Liu, A.J. Thiel, A.M. Sanner, A.E. Condon, L.M. Smith and R.M. Corn, “Demonstration of a Word Design Strategy for DNA Computing on Surfaces,” *Nucleic Acids Res.* **25**(23), 4748–4757 (1997).
17. E. Winfree, X. Yang and N.C. Seeman, “Universal Computation Via Self-assembly of DNA: some theory and experiments,” In *DNA Based Computers II, DIMACS Series in Discr. Math. and Theor. Comput. Sci.* **44**, 191–213 (1998).
18. Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh and E. Shapiro, “Programmable and Autonomous Computing Machine Made of Biomolecules,” *Nature* **414**, 430–434 (2001).
19. M. Li, H.J. Lee, A.E. Condon and R.M. Corn, “DNA Word Design Strategy for Creating Sets of Non-interacting Oligonucleotides for DNA Microarrays,” *Langmuir* **18**(3), 805–812 (2002).
20. K. Cattell, F. Ruskey, J. Sawada and M. Serra, “Fast Algorithms to Generate Necklaces, Unlabeled Necklaces, and Irreducible Polynomials over GF(2),” *J. Algorithms*, **37**, 267–282 (2000).

21. R. Deaton, R.C. Murphy, M. Garzon, D.R. Franceschetti and S.E. Stevens Jr., "Good Encodings for DNA-based Solution to Combinatorial Problems," In *DNA Based Computers II, DIMACS Series in Discr. Math. and Theor. Comput. Sci.* **44**, 247–258 (1998).
22. M. Garzon, P. Neathery, R. Deaton, D.R. Franceschetti, and S.E. Stevens Jr., "Encoding Genomes for DNA Computing," In *Proc. 3rd Annual Genet. Program. Conf.*, Morgan Kaufmann 684–690 (1998).
23. R.S. Braich, N. Chelyapov, C. Johnson, R.W. Rothemund and L. Adleman, "Solution of a 20-Variable 3-SAT Problem on a DNA Computer," *Science* **296**(5567), 499–502 (2002).
24. K. Komiya, K. Sakamoto, H. Gouzu, S. Yokoyama, M. Arita, A. Nishikawa and M. Hagiya, "Successive State Transitions with I/O Interface by Molecules," In *DNA Computing: 6th Intern. Workshop on DNA-Based Computers* (Leiden, The Netherlands), LNCS **2054**, 17–26 (2001).
25. D.C. Tulpan, H. Hoos and A. Condon, "Stochastic Local Search ALgorithms for DNA Word Design," In *Proc. 8th Intern. Meeting on DNA-Based Computers* (Sapporo, Japan), 311–323 (2002).
26. F.J. MacWilliams and N.J.A. Sloane, "The Theory of Error-Correcting Codes," New York, North-Holland, 2nd reprint (1983).

Appendix

110010100000 110001010000† 110000001010 110000000101 101100100000† 101001001000†
101000010001 101000000110† 100101000100† 100100011000 100100000011 100011000010
100010010100 100010001001 100001100001† 100000110010 100000101100† 011100000010
011010000100 011000110000† 011000001001 010110001000 010100100100 010100010001
010011000001 010010010010 010001101000 010001000110 010000100011† 010000011100
001110010000† 001101000001† 001100001100 001010101000† 001010000011 001001100010
001001010100† 001000100101 001000011010† 000110100010 000110000101 000101110000†
000101001010 000100101001† 000100010110 000011100100 000011011000 000010110001†
000010001110 000001010011 000001001101† 001101011111 001110101111 001111110101
001111111010 010011011111† 010110110111† 010111101110† 010111111001 011010111011†
011011100111 011011111100 011100111101† 011101101011 011101110110 011110011110†
011111001101 011111010011† 100011111101† 100101111011 100111001111† 100111110110
101001110111 101011011011 101011101110 101100111110 101101101101 101110010111
101110111001 101111011100† 101111100011 110001101111 110010111110† 110011110011†
110101010111 110101111100† 110110011101 110110101011† 110111011010† 110111100101
111001011101 111001111010 111010001111 111010110101 111011010110† 111011101001
111100011011 111100100111 111101001110† 111101110001 111110101100 111110110010†
000000000000† 111111111111† 000000111111 000011101011† 000101100111 000110011011†
000110111100 001001111001 001010011101 001010110110 001100110011† 001111000110†
010001110101† 010010101101† 010100001111† 010100111010 010111010100 011000010111
011000101110 011011001010† 011101011000† 011110100001 111111000000 111100010100†
111010011000† 111001100100 111001000011† 110110000110 110101100010 110101001001
110011001100 110000111001† 101110001010† 101101010010† 101011110000 101011000101†
101000101011 100111101000 100111010001 100100110101† 100010100111† 100001011110

(12,144,4) Code. Daggers indicate 56 words that satisfy the length-7-subword constraint.

101001100000 011001010000 101101110000 101100001000 011101101000 110011101000
001010011000 101110011000 111001011000 010110111000 001101000100 011101100100
001111010100 001110110100 111010001100 110010101100 101111000010 111001100010
010111100010 111100010010 011000001010 011010100110 100001110110 100100011110
111010010001 110110010001 100110101001 101110000101 111000100101 110101000011
110100100011

Templates of Length 12. When their reversals and 01-flips are included, the total number of words is 74.

000110011101 and 001010111100
001010111100 and 101110011000
010001100111 and 110000101011
110000101011 and 111001100010

000110011101 and 001111010100
001111010100 and 101110011000
010001100111 and 110101000011
110101000011 and 111001100010

Template Pairs Satisfying Minimum Distance 4 and Length-7-subword Constraint.