DNA Sequence Design Based on Template Strategy

Wenbin Liu,*[†,‡] Shudong Wang,[†,‡] Lin Gao,[§] Fengyue Zhang,[†] and Jin Xu[†]

Department of Control Science and Engineering, Huazhong University of Science and Technology,
Wuhan City 430074, China, College of Information Science and Engineering, Shandong University of Science
and Technology, Taian City 271019, China, and College of Computer, Xidian University,
Xi'an City 710071, China

Received November 29, 2002

In DNA based computation and DNA nanotechnology, the design of proper DNA sequences has turned out to be an elementary problem. This paper takes a further look at the template strategy proposed in work by Frutos, A. G. et al. (*Nucleic Acids Res.* **1997**, *25*, 4748−4757). The H-measure proposed by Garzon et al. (Proceedings of the Second Annual Genetic Programming Conference, 1997; pp 472−487) is combined in this strategy to optimize the template and map sets obtained. Finally we describe a constructing method that can still produce more sequences by the results obtained in this paper.

1. INTRODUCTION

The design of codes that satisfy some combinatorial constraints has long been studied in communication systems in order to transform information reliably through a noise channel. It has been proven that the technology of error-correcting code has successfully solved this problem in electronic computers.[1] DNA computation is an attempt to harness the computational power of molecules for information processing. In DNA based computation, information is encoded in DNA sequences, and the retrieval (or recognition) of information is mainly implemented through specific hybridization between DNA sequences. Therefore, the result of DNA computing heavily depends on the biochemical reaction conditions, such as temperature, PH, sodium concentration, and base composition of those sequences. To achieve successful DNA computation, it is crucial to build a good encoding strategy.

Up to now, many computational paradigms have been proposed for solving mathematical problems through biological experiments. Because of the various requirements of those paradigms, it is impossible to establish a complete library of sequences that caters to all the requirements of those paradigms. But there are two common constraints for preventing experimental errors. One is to minimize the similarity between sequences under some distance measures (see 2.1). The other is to standardize the GC content of sequences, which helps to keep the perfect matches in the combinatorial set with a similar binding strength. It is also an easy and good indictor of the melting temperature for short double DNA strands.

Intensive efforts have focused on designing noninteracting DNA sequences through imposing combinatorial constraints on a set of DNA sequences.[2−8] Among which the template strategy originated in ref 9 is a simple and effective approach, in which the authors adopted a heuristic method to reach the largest 8-mer DNA sequences with size 108. In this paper, a random search method is employed to explore the instance of lengths 8-mer, 12-mer, 16-mer, 20-mer, and 24-mer. To deal with the frame shift distance, the H-measure proposed by Garzon et al. in ref 11 is used to optimize the template and map sets. Our method does not guarantee the optimality in terms of the number of sequences, but it is simple, amenable to analysis, and can be used to produce large amounts of relatively good sequences easily.

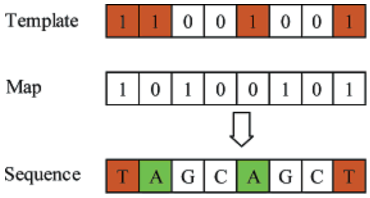

Figure 1. A 8-mer sequence produced by crossing a template strand and a map strand.

2. THE ENCODING METHOD BASED ON TEMPLATE STRATEGY

For clarity, the word "sequence" is referred to DNA sequences to be designed and the word "strand" for binary template and map strands. Without loss of generality, the length of sequences to be designed is assumed as $n = 2l$ in this paper (this allows us to keep the GC content precisely as 50%). The basic idea of the template strategy is to design DNA sequences in two steps through the template and map strands. In the first step, positions for [**AT**] and [**GC**] are determined by a template strand $t = t_1t_2\cdots t_n(t_i\in\{0,1\})$, where 1 indicates [**AT**] and 0 indicates [**GC**]. In the second step, either **A** or **T** is chosen for template positions $t_i = 1$, and either **G** or **C** for positions $t_i = 0(1 \le i \le n)$ by a map strand $m = m_1m_2\cdots m_n(m_i\in\{0,1\})$. The production rule can be described as follows: **A** is obtained from 1 × 0, **T** from 1 × 1, **C** from 0 × 0, and **G** from 0 × 1. Figure 1 shows how a DNA sequence is produced through its template strand and map strand.

2.1. Measurements Used in the Encoding Method. In the design of DNA sequences, the Hamming distance and

* Corresponding author e-mail: wbliu@mail.hust.edu.cn, wbliu69@sohu.com.
[†] Huazhong University of Science and Technology.
[‡] Shandong University of Science and Technology.
[§] College of Computer, Xidian University.

10.1021/ci025645s CCC: $25.00 © 2003 American Chemical Society
Published on Web 08/20/2003

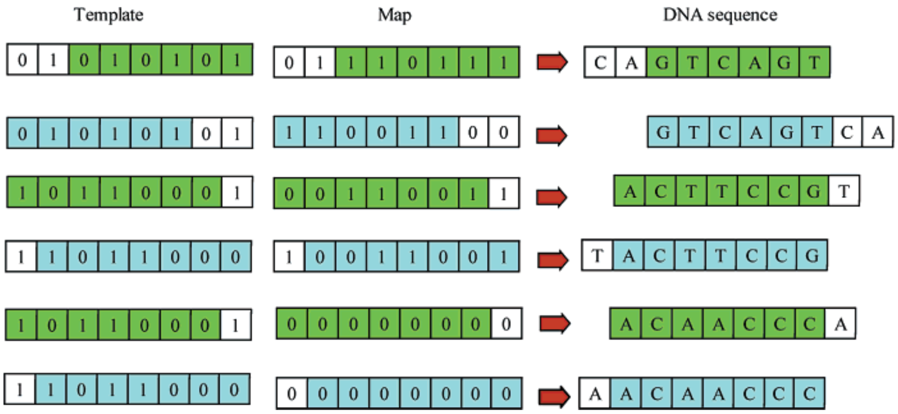

Figure 2. Examples of the frame shift instance.

its extended forms are generally used to evaluate the similarity between those sequences. The definition of those distances applied in this paper is described as follows:[10,11]

- The **Hamming distance** is defined as the number of corresponding places where two characters differ between two distinct sequences (or strands) s_i and s_j.
- The **reverse Hamming distance** between two distinct sequences s_i and s_j is the Hamming distance of sequences s_i and s_j^r, where s_j^r denotes the reverse sequence of s_j.
- The **reverse-complement Hamming distance** between two distinct sequences s_i and s_j is the Hamming distance of sequence s_i^c and s_j^r, where s_i^c denotes the complementary of sequence s_i. For binary strands, the complement of 0 is 1 and vice versa, the complement of 1 is 0.
- The **H-measure** between two sequences s_i and s_j is essentially the minimum of all Hamming distances obtained by successively shifting and lining up sequence s_i against sequence s_j. It can be formulated as

$$|s_i,s_j| = \min_{0\leq k<n} H(s_i,\rho^k(s_j))$$

where $\rho^k(\rho^{-k})$ means to shift sequence s_j to the right (left) against s_i with k positions, n is the length of the sequences, and $H(*,*)$ is the ordinary Hamming distance. This metric was first proposed by Garzon et al. in ref 11.

In this paper, we additionally require that $k \neq 0$ for $s_i = s_j$ such that the formula above can still be used to measure the maximal iterative subsequence in the sequence itself (described in 4.2). The advantage of the H-measure is that it considers the frame-shift distance between sequences (shown in Figure 2), and thus can be regarded as measuring the degree of inertia against hybridization.

2.2. Constraints for the Template Set and Map Set. Roughly speaking, the likelihood of hybridization between two sequences s_i and s_j decreases as the number of mismatches between them increases. Therefore, two properties are necessary for all sequences in a code set: (1) sequence s_i should differ from s_j in many bases and (2) the Watson−Crick complement of sequence s_i should differ from s_j in many bases. In template strategy, it is easy to see that a sequence s and its complement, s^c, can be produced as (their orientation is from 5′ to 3′)

$$t \times m \rightarrow s$$
$$t^r \times m^{rc} \rightarrow s^c$$

To hold the two properties above, the template and map set should satisfy following constraints. Detailed descriptions are recommended to the work of Frutos et al. in ref 9.

- The number of 1 and 0 in template strands should remain equal so that the GC content in sequences produced will be 50%; in this paper such a requirement is also held for the map strands, and it will be explained in 4.2 (there is no such requirement for map strands in ref 9). Thus, both the template and map set come from the binary strands with equal 01 content.
- Both the Hamming distance and the reverse Hamming distance should be larger than or equal to l for the template set. Each template set T consists of two subsets T_n (the nonpalindrome set) and T_s (the palindrome set) according to the symmetry property and the number of T_n is usually larger than that of T_s.
- There are two map subsets M_n and M_s which correspond to the subsets T_n and T_s, respectively. For subset M_n, only the Hamming distance is required to be larger than or equal to l; while for subset M_s, both the Hamming distance and the reverse Hamming distance should be larger than or equal to l. Obviously, subset M_s is substantially equal to subset T_n. So the largest number of sequences can be obtained by the template strategy is

$$N_{\max} = |T_n|(|T_s| + |M_n|)$$

where |*| expresses the code number of set *.

2.3. The Encoding Problem. In the template strategy, the encoding problem is thus decomposed into two subproblems—to find as large as possible a template set T and a map set M that satisfy the correspondent constraints above so that they can produce the largest set of possible codes available for DNA computing. Because both the set T and M are actually binary strands, the two subproblems are then transformed to search the maximum-sized subsets over the binary hypercube space, in which each element holds some distance property with other elements. A n dimensional hypercube is an undirected graph with 2^n vertices, each of which is represented by a unique binary strand, and any two vertices are adjacent if and only if the Hamming distance between them is one. As described in ref 9, the encoding problem is in fact an instance of the well-known NP-hard problems, named as the independent set problem. It has been proved that there is no efficient algorithm for them.[12] Due to the constraint on GC content, the solution space for template set T and map set M decreases from 2^n to $(2l)!/l!(l)!$. Table 1 shows the solution space for lengths 8, 12, 16, 20, and 24.

Table 1. Possible Solution Space for $n = 8, 12, 16, 20, 24'$

length	8	12	16	20	24
number	70	924	12870	184756	2704156
time(s)	0.03	0.45	7	76	960

Clearly, the solution space increases dramatically with the code length n. In ref 9, Frutos et al. solved this problem through a heuristic inference process at length 8. But as n increases, their method will not work. In this paper, a random search method is used to search the approximate optimal solution for lengths 8, 12, 16, 20, and 24.

3. THE RANDOM SEARCH METHOD

The random search algorithm for the template set T can be described as the following:

Step 1: Generate all binary strands with an equal number of 0's and 1's.

Step 2: Partition those binary strands into two groups: nonpalindrome group and palindrome group.

Step 3: In the nonpalindrome group, randomly select a binary strand and remove those conflicting with it.

Step 4: Repeat step 3 until no strand remains and then a set of T_n is obtained.

Step 5: In the palindrome group, randomly select a binary strand and remove all those strands conflicting with T_n.

Step 6: Repeat step 5 until no strand remains and then a set of T_s is obtained.

For the map set M_n, only steps 1, 3, and 4 are need. To approach the approximate largest sets, the above algorithm is performed as many times as possible at each instance. In Table 1, the average time consumed by the algorithm is presented (it is run in a Pentium IV, 2.4 GHz computer). As the length increases, the run time increases dramatically. From its definition, to calculate the H-measure between two strands, we have to perform $2n - 1$ compares. If the H-measure is used directly in the algorithm above, the time consumed will be greatly increased. For example, the average time consumed will be at least 12.4 h as n is 24. Therefore it is unreasonable to use this metric directly. In this paper, it is used to optimize the template sets and map sets obtained in the following section.

4. RESULTS AND DISCUSSIONS

4.1. Results of the Algorithm. In this paper, the submap set M_n is actually the special Hamming code $(4k,8k,2k)(k = 1,2,3\cdots)$, which respectively means the code length, the maximal code number that can be attained, and the Hamming distance between codes.[1,10] Apart from the strands of all 0's and all 1's, the largest size of M_n then becomes $8k - 2$. Our algorithm also reaches the largest subset M_n for code length n is 8, 12, 16, 20, and 24. That is the reason these instances are considered in this paper. Considering the template set T, the largest template sets obtained for lengths 8, 12, 16, 20, and 24 are 8(2), 8, 16(2), 11, and 15(1), respectively (values in parentheses refers to the number of T_s). Our results show that the largest set for template T and map set M is not unique; there are many possible sets available for further selection.

4.2. Optimization of the Template and Map Set. The drawback of the template strategy is that it does not take into account the so-called frame shift distance, in which DNA sequences satisfying the three Hamming distances in 2.1 can still share a high similarity through shifting some positions. Further experimental and theoretical research has shown that such an instance must be dealt with decisively in order to attain a reliable computation. As shown in Figure 2, there are three cases that can lead to DNA sequences with low H-measure. The first is one template strand and two different map strands; the second is two template strands crossing two map strands; the third is two template strands crossing one map strand.

Generally speaking, two conditions are necessary to produce DNA sequences with low H-measure in the template strategy: (1) Both the two template strands share a long common substrand $t' = t_1t_2\cdots t_k$, and the two map strands share another long common substrand $m' = m_1m_2\cdots m_k$ $(l < k < 2l)$. (2) Both the two substrands t_i' and m_i' $(i = 1,2)$ are located in the same position of the template and the map strands.

In fact, the more long common substrands appear in the template set and the map set, the more possibility of producing low H-measure DNA sequences. Moreover, strands such as "01010101", "00000000", and "11111111" are very deleterious either in the template set or the map set. This is because there exist many iterative substrands in those strands themselves. For example, there are two substrands "0000000", three substrands "000000", and four substrands "00000" in strand "00000000". As the length of the strand increases, this situation becomes more serious. Particularly, if all 0's and all 1's strands are in the map set, low H-measure sequences are surely produced no matter where the long common substrands locate in the template set (see the third case in Figure 2). That is the reason we exclude all 0's and all 1's strands from the map set in 2.2. In addition, if there are more 0's than 1's in a map strand, the possibility of iterative substrands will also increase. From this observation, it is reasonable to keep the map set with equal 01 content.

To give some intuition, the five largest template sets and the two largest map sets at length 8 are presented in Table 2. For each set, its average H-measure and the total number of H-measures less than 3 are also calculated. Ten sets of DNA sequences with the total number of 96 can be produced from these template and map sets. Figure 3 shows the total number of H-measures less than 3 for each set of DNA sequences. It is easy to come to the following two conclusions: (1) the H-measure properties of template and map sets have an influence on that of the set of DNA codes produced; (2) when the template sets have a higher H-measure value, the map sets seem to have less influence upon the properties of DNA sequences produced; while as the H-measure value of the template sets decreases, the influence of map sets increases dramatically. Additionally, the influence of map strands of "00000000" and "11111111", which are added in map set $M1$, is also presented in the third column of each group in Figure 3. Obviously, their existence deteriorates the H-measure property significantly.

Generally, the size of the template set T and the map M set is relatively small, so it is practical for us to optimize the quality of these sets by the H-measure. Figure 3 indicates that this metric can greatly improve the quality of sequences produced by the template strategy. And moreover, because the properties of the DNA sequences produced depend

Table 2. Largest Template Sets and Map Sets with Length 8

	T1	T2	T3	T4	T5	M1	M2
Tn	11010100	11000101	11001100	01010011	01001110	10000111	00001111
	01000111	11001010	01010101	11010100	01010101	10011001	00110011
	10010011	00001111	11010010	11100001	01101001	10101010	00111100
	01110001	01011100	11100001	00110101	00011011	10110100	01010101
	01101100	01101001	10110100	01001101	00100111	11001100	01011010
	00011101	10101100	01111000	01111000	10001101	11010010	01100110
						11100001	01101001
Ts	01011010	01100110	01100110	01100110	00111100	00011110	10010110
	10100101	10011001	10011001	10011001	11000011	00101101	10011001
						00110011	10100101
						01001011	10101010
						01010101	11000011
						01100110	11001100
						01111000	11110000
Average H-measure	3. 6	3. 5	3. 4	3. 3	3. 2	3. 42	3. 4
Number(H<3)	2	6	8	8	7	23	24

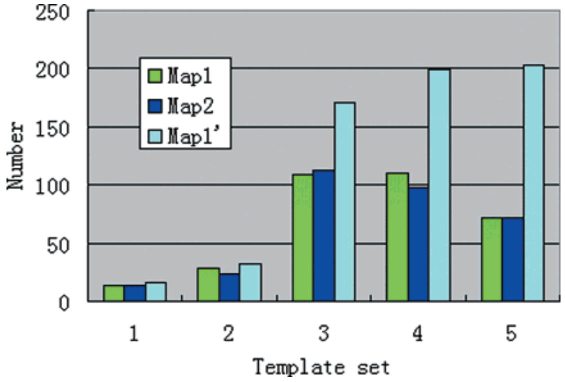

Figure 3. Number of the DNA sequences with H-measure less than 3.

heavily on that of the template set, more efforts should be paid in the optimization of template sets.

Clearly, there is a trade off between the stringency of the H-measure and the number of sequences that can be designed. To obtain as many codes as possible, the lower limit of the H-measure should be required. Unfortunately, it is still not clear what is the lowest H-measure that may be suitable for DNA computing. From the standpoint of the chemical reaction, the Gibbs free energy is the key driven force that carries out the hybridization reaction between DNA sequences. If there is enough change in the Gibbs free energy between the perfect and imperfect hybridization, the possibility of unwanted hybridization will be very small. And the change of the Gibbs free energy generally increases with the total number of hybridized base pairs. It seems that $n/3$ may be a reasonable value, and M. Arita et al. also supposed this value in ref 3.

4.3. Melting Temperature and Gibbs Free Energy. To perform a reliable computation, it is important to achieve similar Gibbs free energies and melting temperatures (t_m) for the perfectly matched hybridized DNA duplexes. The melting temperature is defined as the temperature at which half of the strands are in the double helix state and half are in the "random coil" state. From biochemistry, the nearest-neighbor model is now a simple and suitable model to calculate thermodynamic data for short DNA strands. In this paper, the melting temperature t_m and Gibbs free energy $-\Delta G°_{37}$ for the perfectly matched duplexes are calculated according to the parameters obtained in ref 13. Table 3 shows the range of the melting temperature t_m and Gibbs free energy $-\Delta G°_{37}$ of those DNA sequences produced by our algorithm in the cases of lengths 8, 12, 16, 20, and 24 (the optimized codes of the template sets T and map sets M are available in the Supporting Information of this paper). As expected, the two parameters fall in a narrow range and rise as the length of DNA sequences increases.

Table 3. Thermodynamic Parameters for the DNA Sequences[a]

n	8	12	16	20	24
t_m	39−54.5	63−71	71.5−78.7	78.6−85	82.7−89.4
$-\Delta G°_{37}$	6.1−8	11.5−14	16.5−19.7	21.9−25.8	27.1−31.2

[a] t_m: °C, $-\Delta G°_{37}$: kcal/mol.

4.4. Construction of the Template and Map Set for Larger Length. As the length of the sequences increases, both the solution space and the time consumed by the random search program will increase dramatically. As the length n increases to 28, the solution space becomes 40116600, and it seems too difficult to find the largest set at this instance. Here a construction strategy is introduced so that we can still reach a large amount of DNA codes as the length increases. Assuming that T and M is the largest template and map set obtained at length n, for each $t_i \in T$, two template strands t_it_i and $t_it_i^c$ of length $2n$ can be constructed with length $2n$; similarly, for each $m_i \in M$, two map strands m_im_i and $m_i^cm_i^c$ of length $2n$ can also be constructed. It is easy to prove that the constructed template and map sets also satisfy the constraints described in 2.2. The total number of DNA codes obtained at length $2n$ will approximately be 4 times of that of the number in length n. Finally, it should be noticed that there are two cases in which the constructed strands belong to the subset T_s: (1) if $t_i^r = t_i^c$, strand $t_i t_i^c$ will be a palindrome structure and (2) if $t_i \in T_s$, then strand $t_i t_i$ is also a palindrome structure. Generally, the less the number of T_s, the more codes we can obtain.

5. CONCLUSIONS

Encoding is the first and most important step in DNA based computing. The properties of the DNA sequences have a great influence on the final result of DNA computing. To implement a more reliable computation, various strategies in the encoding method, such as the minimal common substrand strategy,[5] Genetic Algorithm,[2,6,8] template strategy,[9] etc., have been proposed. In this paper, we study on the template strategy combined with the H-measure. Our results show that the H-measure can be used to optimize the template and map sets very easily, and, more important, it improves the quality of DNA sequences produced by the template strategy. As the code length increases, more codes can still be obtained from results in short instances through a constructing strategy.

Compared with method presented in ref 14, where the authors actually use two map sets to produce sequences satisfying the Hamming distance, the advantage of our method is that it can practically reduce or even eliminate sequences with low H-measure dramatically and increase robustness of those sequences produced. The Genetic Algorithm (GA) is also a useful method to solve combinatorial problems, but we do not think it is suitable in the encoding problem of DNA computing. First, up to now, no result shows that the GA can produce large amounts of useful codes. Second, the key problem in GA is to construct the fitness function. But it is not an easy task to unite all those constraints in one function, furthermore the relative weight value of these constraints is not clear.

The advantage of the template strategy is that it is simple and flexible, and large amounts of codes with high quality can be obtained from the template set and map set. In DNA computing, the forbidden subsequences, such as PCR primers or the recognition sites for restriction enzymes, frequently play an important role. In the template strategy, this constraint can be easily satisfied through deleting the template strands that can produce those forbidden subsequences. Finally, we should address that the validity of the codes obtained by template strategy needs further investigation through biological experiments.

ACKNOWLEDGMENT

This paper is supported by the National Scientific Foundation of China under Grants 60103021 and 60144047. The authors are gratefully acknowledged to the anonymous referees' suggestions to this paper. Our appreciation extends as well to M. Li, M. Arita, A. E. Condon, and M. Garzon for their valuable help in some problems about the encoding method.

Supporting Information Available: Relatively optimized template and map sets for lengths 12, 16, 20, and 24 (Tables 4−7). This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) MacWilliams,F. J.; Solane, N. J. A. *The theory of Error-Correcting Codes*; North-Holland: 1977.

(2) Arita, M. et al. Improving Sequence Design for DNA computing. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2000), Las Vegas, 2000; pp 875−882.

(3) Arita, M. et al. DNA sequence Design using Template. *New Generation Comput.* **2002**, *20*(3), 263−277.

(4) Arita, M. et al. The power of Sequence Design in DNA computing. 4th International Conference on Computational Intelligence and Multimedia Applications, 2001; pp 163−166.

(5) Baum, E. B. DNA Sequences Useful for Computation. Proceedings of 2nd DIMACS Workshop on DNA Based Computers, 1996; pp 235−242.

(6) Deaton, R. et al. Genetic Search of Reliable Encodings for DNA-Based Computation. 1st Genetic Programming Conference, Stanford University, 1996; pp 9−15.

(7) Deaton, R. et al. Good Encodings for DNA-Based Solutions to Combinatorial Problems. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1999; Vol. 44, pp 247−258.

(8) Deaton, R. et al. A DNA Based Implementation of an Evolutionary Computation. Proceedings IEEE Conference on Evolutionary Computation, Indiana, 1997; pp 267−271.

(9) Frutos, A. G. et al. Demonstration of a Word Design Strategy for DNA Computing on Surface. *Nucleic Acids Res.* **1997**, *25*, 4748−4757.

(10) Marathe, A.; Condon, A. E.; Corn, R. M. On Combinatorial DNA Code Design. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1999; Vol. 44, pp 75−87.

(11) Garzon, M. et al. A new metric for DNA Computing. Proceedings of the 2nd Annual Genetic Programming Conference, 1997; pp 472−487.

(12) Garey, M. R.; Johnson, D. S. *Computers and intractability: A Guide to the theory of NP-Completeness*; W. H. Freeman and Company: New York, 1979.

(13) SantaLucia, J. et al. Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability. *Biochemistry* **1996**, *35*, 3555−3562

(14) Li, M. et al. DNA Word Design Strategy for Creating Sets of Noninteracting Oligonucleotides for DNA Microarrays. *Langmuir* **2002**, *18*, 805−812.

CI025645S