Published online February 8, 2005

Nucleic Acids Research, 2005, Vol. 33, No. 3 ***903–911***
doi:10.1093/nar/gki235

Design of nucleic acid sequences for DNA computing based on a thermodynamic approach

Fumiaki Tanaka[1,*], Atsushi Kameda[2], Masahito Yamamoto[1,2] and Azuma Ohuchi[1,2]

[1]Graduate School of Engineering, Hokkaido University, North 13, West 8, Kita-ku, Sapporo 060-8628, Japan and [2]CREST, Japan Science and Technology Corporation, 4-1-8, Honmachi, Kawaguchi, Saitama, 332-0012, Japan

Received October 19, 2004; Revised and Accepted January 20, 2005

ABSTRACT

We have developed an algorithm for designing multiple sequences of nucleic acids that have a uniform melting temperature between the sequence and its complement and that do not hybridize non-specifically with each other based on the minimum free energy (ΔG_{min}). Sequences that satisfy these constraints can be utilized in computations, various engineering applications such as microarrays, and nano-fabrications. Our algorithm is a random generate-and-test algorithm: it generates a candidate sequence randomly and tests whether the sequence satisfies the constraints. The novelty of our algorithm is that the filtering method uses a greedy search to calculate ΔG_{min}. This effectively excludes inappropriate sequences before ΔG_{min} is calculated, thereby reducing computation time drastically when compared with an algorithm without the filtering. Experimental results *in silico* showed the superiority of the greedy search over the traditional approach based on the hamming distance. In addition, experimental results *in vitro* demonstrated that the experimental free energy (ΔG_{exp}) of 126 sequences correlated well with ΔG_{min} ($|R| = 0.90$) than with the hamming distance ($|R| = 0.80$). These results validate the rationality of a thermodynamic approach. We implemented our algorithm in a graphic user interface-based program written in Java.

INTRODUCTION

Nucleic acids are now being utilized in computations (1–3), various engineering applications such as microarrays (4–6), and nano-fabrications (7–9). In these fields of research, called 'DNA computing', nucleic acid design or sequence design is a crucial problem in engineering using nucleic acids. Nucleic acid design is deciding the base sequences (e.g. 'GCTAGCTAGGTTTA', . . . , 'ATCGTACGCTATGTGCA' in DNA) in order to satisfy the constraints based on the physicochemical properties of nucleic acids. In particular, it is essential to prevent undesired hybridization. In DNA computing, multiple sequences need to be designed that do not hybridize non-specifically with each other (10,11), while in RNA secondary structure design, a single sequence needs to be designed that folds into the desired secondary structure (12–15). For example, 40 DNA sequences of length 15 were designed to prevent undesired hybridization and used for solving a 20-variable instance of a three-satisfiability problem (2). In the field of microarrays, 69 122 DNA sequences of length 45–47 were designed to hybridize specifically to 24 502 transcripts from *Arabidopsis thaliana* (4). Furthermore, four DNA sequences of length 26 and four DNA sequences of length 48 were used to construct a periodic two-dimensional crystalline lattice (7). These sequences were carefully designed for the intended hybridization. Thus, an algorithm/program that generates multiple sequences, which do not hybridize non-specifically with each other, is useful for various applications of nucleic acid.

We applied a thermodynamic approach to the nucleic acid design for DNA computing. In particular, we focused on designing a pool P containing n sequences of length l for which (i) the duplex melting temperature (T_M) is in the range T_M^- to T_M^+ for any pairwise duplex of a sequence in P and its complement and (ii) the minimum free energy (ΔG_{min}) is greater than a threshold (ΔG^*_{min}) in any pairwise duplex of sequences in P and any concatenation of two sequences in P plus their complements except for the pairwise duplex of a sequence in P and its complement. Traditional approaches to the sequence design in DNA computing have approximated the stability between two sequences using the hamming distance (i.e. the number of base pairs) rather than ΔG_{min}. However, since the hamming distance is only an approximation of the stability, ΔG_{min} is preferable for predicting the stability. In practice, an RNA secondary structure can be adequately predicted using ΔG_{min} rather than the number of base pairs (16). However, the algorithm for calculating the ΔG_{min} for double-stranded DNA requires time complexity $O(l^3)$, where l is the length of the sequence, as is the case with secondary structure prediction for single-stranded RNA.

*To whom correspondence should be addressed. Tel: +81 11 716 2111, ext 6498; Fax: +81 11 706 7834; Email: fumiaki@dna-comp.org

Since all the combinations of three sequences must be evaluated in this design, dozens of sequences cannot be designed within a reasonable computation time. Another approach that reduces the computation time is thus needed. Andronescu *et al*. (13) overcame this drawback in the secondary structure design by hierarchically decomposing the secondary structure into smaller substructures and integrating the partial sequences. By using this method, they designed sequences that a traditional program cannot.

We have developed a random generate-and-test algorithm that generates a candidate sequence randomly and tests whether the sequence satisfies the constraints. It stores the sequence in a sequence pool if and only if it satisfies all the constraints. To reduce the difficulty in computation time, we use a greedy search to calculate $\Delta G_{\min}$. The advantage of a greedy search is that it approximates $\Delta G_{\min}$ in less time, with time complexity $O(l^2)$, than a rigorous algorithm, which calculates $\Delta G_{\min}$ with time complexity $O(l^3)$. The $\Delta G_{\min}$ approximated using the greedy search (denoted by ΔG_{gre}) correlated well with $\Delta G_{\min}$. The correlation coefficients were 0.95, 0.85 and 0.76 at 20mer, 40mer and 60mer lengths, respectively. Furthermore, since ΔG_{gre} is the upper bound for $\Delta G_{\min}$ (i.e. $\Delta G_{\min} \leqslant \Delta G_{\mathrm{gre}}$), the sequence such that $\Delta G_{\mathrm{gre}} \leqslant \Delta G^*_{\min}$ is sure to satisfy $\Delta G_{\min} \leqslant \Delta G^*_{\min}$. Therefore, using a greedy search excludes in advance most inappropriate sequences before $\Delta G_{\min}$ is calculated without excluding an appropriate sequence. With this approach, our algorithm reduces computation time.

To evaluate our algorithm, we investigated the effectiveness of the greedy search filtering. We compared the computation time of our algorithm with that of the same algorithm but without the greedy search filtering. The experimental results showed that the greedy search filtering reduces the total computation time drastically. For example, using the filtering reduced the computation time to 83% for 30 sequences with length 20 and to 87% for 20 sequences with length 15 when $T^-_{\mathrm{M}} = 69.58$, $T^+_{\mathrm{M}} = 72.58$ and $\Delta G^*_{\min} = -10.0$, and $T^-_{\mathrm{M}} = 60.49$, $T^+_{\mathrm{M}} = 63.49$ and $\Delta G^*_{\min} = -7.0$, respectively. In addition, we compared the greedy search with the traditional approach based on a hamming distance in terms of the filtering performance. We demonstrated that the greedy search can filter out inappropriate sequences better than using the hamming distance. In a laboratory experiment, we investigated the correlation coefficient between $\Delta G_{\min}$ and the experimental free energy (ΔG_{exp}) and that between the hamming distance and the ΔG_{exp} using 126 duplexes. The ΔG_{exp} correlated better with $\Delta G_{\min}$ ($|R| = 0.90$) than with the hamming distance ($|R| = 0.80$).

To implement our algorithm, we developed a computer program called DNA-SDT, a graphic user interface (GUI)-based application written in Java. This program enables users to design DNA sequences with our algorithm and can be downloaded freely from the web site (http://ses3.complex.eng.hokudai.ac.jp/~fumi95/DNA-SDT/index.html).

ALGORITHM

Definition

Let n be the number of sequences to be designed and l_i ($0 \leqslant i \leqslant n-1$) be the length of each sequence. In this paper, we formulated l_i such that $l_i = l_j = l$ ($0 \leqslant i, j \leqslant n-1$), although we can extend our algorithm easily for $l_i \neq l_j$ ($0 \leqslant i \neq j \leqslant n-1$). We define P as the pool of n sequences, with length l, to be designed. Furthermore, let $P = \{U_0, U_1, \ldots, U_{n-1}\}$ and $Q = \{V_0, V_1, \ldots, V_{n-1}\}$ such that V_i ($0 \leqslant i \leqslant n-1$) is the complement of U_i. T^-_{M} and T^+_{M} are defined as the lower and upper thresholds of T_{M} for the duplex between a sequence and its complement. Moreover, $\Delta G^*_{\min}$ is defined as the threshold of $\Delta G_{\min}$ given by the sequence designer. Thus, our algorithm designs a pool P containing n sequences of length l for which (i) the duplex T_{M} is in the range from T^-_{M} to T^+_{M} for any duplex of U_i ($0 \leqslant i \leqslant n-1$) and V_i, and (ii) $\Delta G_{\min}$ is greater than $\Delta G^*_{\min}$ for any pairwise duplex of sequences in P and any concatenation of two sequences in $P \cup Q$ except for the pairwise duplex of U_i and V_i.

Here, we describe more specifically the combination of sequences to be calculated using the $\Delta G_{\min}$. The sequence U_i ($0 \leqslant i \leqslant n-1$) is denoted by a string of bases such as $u_i^0 u_i^1 \cdots u_i^{l-1}$ (5′ to 3′ direction), and similarly, V_i ($0 \leqslant i \leqslant n-1$) is denoted as $v_i^0 v_i^1 \cdots v_i^{l-1}$ (5′ to 3′ direction). For example, if U_i = 5′-AAATTTCCCGGG-3′, then V_i = 5′-CCCGGGAAATTT-3′. Furthermore, let <X, Y> be the combination of sequences X and Y, and XY be the concatenation of sequences X and Y in that order. For example, if U_i, U_j and U_k ($0 \leqslant i, j, k \leqslant n-1$) are 5′-AAATTT-3′, 5′-CCCGGG-3′ and 5′-TCTCTC-3′, respectively, then <U_iU_j,U_k> means the combination of sequences 5′-AAATTTCCCGGG-3′ and 5′-TCTCTC-3′. In our algorithm, the following combinations are considered for the $\Delta G_{\min}$ calculation.

(i) $\langle U_i\, U_j\, U_k \rangle$ ($0 \leqslant i, j, k \leqslant n-1$)
(ii) $\langle U_i\, U_j\, V_k \rangle$ ($0 \leqslant i, j, k \leqslant n-1$), $i \neq k$
(iii) $\langle U_i\, V_j\, U_k \rangle$ ($0 \leqslant i, j, k \leqslant n-1$), $i \neq j$
(iv) $\langle U_i\, V_j\, V_k \rangle$ ($0 \leqslant i, j, k \leqslant n-1$), $(i \neq j) \wedge (i \neq k)$.

For generality, we use two sequences, $S(=s_0\, s_1 \cdots s_{N-1})$ (5′ to 3′ direction) and $T(=t_0\, t_1 \cdots t_{M-1})$ (3′ to 5′ direction), to describe the $\Delta G_{\min}$ calculation in detail. The sequences are defined to be antiparallel to each other. Note that any two sequences can be represented by S and T. In this paper, S and T represent U_i and the reverse sequence of X_jY_k ($X_j \in \{U_j, V_j\}$, $X_k \in \{U_k, V_k\}$).

The notation $s_i \cdot t_j$ represents the base pair between the i-th base in sequence S and the j-th base in sequence T, hence the structure between S and T is a set of base pairs such that each base is paired at most once. In addition, $(s_i \cdot t_j,\ s_{i'} \cdot t_{j'})$ $\{(0 \leqslant i < i' \leqslant N-1) \wedge (0 \leqslant j < j' \leqslant M-1)\}$ is defined as a structure in which base pairs $s_i \cdot t_j$ and $s_{i'} \cdot t_{j'}$ are formed and the sequences $s_{i+1}s_{i+2} \cdots s_{i'-1}$ and $t_{j+1}t_{j+2} \cdots t_{j'-1}$ do not form any base pairs. The term $(x, s_{i'} \cdot t_{j'})$ $\{x \in \{s_i, t_j\}, (0 \leqslant i < i' \leqslant N-1) \wedge (0 \leqslant j < j' \leqslant M-1)\}$ is defined as a structure in which base pair $s_{i'} \cdot t_{j'}$ is formed and sequence $s_is_{i+1} \cdots s_{i'-1}$ in the case $x = s_i$ ($t_jt_{j+1} \cdots t_{j'-1}$ in the case $x = t_j$) do not form any base pair. Similarly, $(s_i \cdot t_j, x)$ $\{x \in \{s_{i'}, t_{j'}\}, (0 \leqslant i < i' \leqslant N-1) \wedge (0 \leqslant j < j' \leqslant M-1)\}$ is defined as a structure in which base pair $s_i \cdot t_j$ is formed and the sequence $s_{i+1}s_{i+2} \cdots s_{i'}$ in the case $x = s_{i'}$ ($t_{j+1}t_{j+2} \cdots t_{j'}$ in the case $x = t_{j'}$) do not form any base pair.

Outline

Our algorithm is a random generate-and-test algorithm that generates a sequence randomly and then stores it in the pool if

and only if the sequence satisfies all the constraints. The main feature of our algorithm is that it uses a greedy search for calculating $\Delta G_{\min}$ to filter out inappropriate sequences. The advantage of a greedy search is that it approximates $\Delta G_{\min}$ in less time when compared with a well-known dynamic programming algorithm (17). Therefore, using the greedy search before the $\Delta G_{\min}$ calculation to exclude inappropriate sequences reduces the computation time. Hereafter, the approximated $\Delta G_{\min}$ using a greedy search is denoted by ΔG_{gre}.

The algorithm uses three filters:

(i) T_{M} filter: checks whether the T_{M} of candidate sequence U_c $(0 \leqslant c \leqslant n-1)$ and V_c is in the range from T_{M}^- to T_{M}^+. If not, U_c is rejected.

(ii) ΔG_{gre} filter: checks whether ΔG_{gre} is greater than the threshold, $\Delta G_{\mathrm{gre}}^*$, for all the combinations above in P, provided that the candidate sequence U_c $(0 \leqslant c \leqslant n-1)$ or V_c is included in that combination. If the ΔG_{gre} of any combination is less than or equal to $\Delta G_{\mathrm{gre}}^*$, U_c is rejected.

(iii) $\Delta G_{\min}$ filter: checks whether $\Delta G_{\min}$ is greater than the threshold, $\Delta G_{\min}^*$, for all the combinations above in P, provided that the candidate sequence U_c $(0 \leqslant c \leqslant n-1)$ or V_c is included in that combination. If the $\Delta G_{\min}$ of any combination is less than or equal to $\Delta G_{\min}^*$, U_c is rejected.

The T_{M} and $\Delta G_{\min}$ filters are necessary for satisfying the constraints of sequence design, while the ΔG_{gre} filter reduces the computation time. The use of the ΔG_{gre} filter is based on the hypothesis that it can exclude most sequences that cannot pass through the $\Delta G_{\min}$ filter. If this hypothesis is true, the ΔG_{gre} filter can exclude the inappropriate sequences in less time, resulting in reduced total computation time.

The algorithm is defined as follows:

- Input: n, l, T_{M}^-, T_{M}^+, $\Delta G_{\mathrm{gre}}^*$, $\Delta G_{\min}^*$
- Output: pool P consisting of n sequences with length l.
- Procedure:

(i) Initialize pool P as an empty set.

(ii) Iterate the following procedure until P has n sequences.

(a) Generate candidate sequence U_c $(0 \leqslant c \leqslant n-1)$ with length l randomly, then add U_c to P.

(b) Evaluate U_c with T_{M} filter. If U_c is rejected, exclude U_c from P and return to ii(a).

(c) Evaluate U_c with ΔG_{gre} filter. If U_c is rejected, exclude U_c from P and return to ii(a).

(d) Evaluate U_c with $\Delta G_{\min}$ filter. If U_c passes, leave U_c in P; else exclude U_c from P and return to ii(a).

The order of the filters is important. Each candidate sequence should be evaluated in this order to reduce the computation time. If pool P has m sequences, the time complexities to evaluate the $(m+1)$-th candidate sequence are $O(l)$, $O(m^2l^2)$ and $O(m^2l^3)$ at the T_{M}, ΔG_{gre} and $\Delta G_{\min}$ filters, respectively. By evaluating the candidate sequences in ascending order of time complexity, the inappropriate ones can be excluded sooner.

Energy model

The $\Delta G_{\min}$ between S and T is calculated using a dynamic programming algorithm (17), while the ΔG_{gre} between S and T is calculated using a greedy search. Both $\Delta G_{\min}$ and ΔG_{gre} are calculated based on the nearest-neighbor model, which calculates the total free energy as the summation of the contributions of various elementary structures (18,19). The elementary structures considered in this paper are stacking base pairs, bulge loops, internal loops, dangling ends and free ends.

The contributions of the stacking base pairs, defined as $(s_i \cdot t_j, s_{i+1} \cdot t_{j+1})$, to the free energy are calculated using 12 parameters reported previously (19). The free energy contributions of the loop regions are sequence dependent (15,16,20). The free energies of single bulge loops, defined as $(s_i \cdot t_j, s_{i+2} \cdot t_{j+1})$ or $(s_i \cdot t_j, s_{i+1} \cdot t_{j+2})$, are calculated using 64 parameters covering all the possible combinations of bulged base and flanking base pairs (19). The free energies of the other loops, bulge loops longer than one and internal loops, are calculated using conventional parameters and equations (20,21). Bulge loops longer than one are defined as $\{(s_i \cdot t_j, s_{i+l} \cdot t_{j+1}) \wedge (l \geqslant 3)\}$ or $(s_i \cdot t_j, s_{i+1} \cdot t_{j+l}) \wedge (l \geqslant 3)$, and the internal loops are defined as $(s_i \cdot t_j, s_{i+l} \cdot t_{j+m}) \wedge (l, m \geqslant 2)$. The free energies of dangling ends, defined as $(s_0, s_1 \cdot t_0)$, $(t_0, s_0 \cdot t_1)$, $(s_{N-2} \cdot t_{M-1}, s_{N-1})$ or $(s_{N-1} \cdot t_{M-2}, t_{M-1})$, are calculated using 32 parameters covering all the possible combinations (22). The free ends are defined as the sequences $s_0 \cdots s_i$ and $t_0 \cdots t_j$ closing with $s_i \cdot t_j$ such that both $s_{i'(<i)}$ and $t_{j'(<j)}$ do not form a base pair or $s_i \cdots s_{N-1}$ and $t_j \cdots t_{M-1}$ closing with $s_i \cdot t_j$ such that both $s_{(i<)i'}$ and $t_{(j<)j'}$ do not form a base pair. The free energies of the free ends are also calculated using conventional parameters (20).

The crossing base pairs are defined as a pair of base pairs $s_i \cdot t_j$ and $s_{i'} \cdot t_{j'}$ in a structure with $\{(i < i') \wedge (j' < j)\}$ or $\{(i' < i) \wedge (j < j')\}$. We prohibit crossing base pairs because of the computation time and the lack of thermodynamic data. This constraint is equivalent to the pseudoknot-free constraint in the RNA secondary structure prediction. At this point, we do not consider intra-molecular base pairs or the interactions between loop regions.

T_{M} filter

This filter checks whether a candidate sequence paired to its complement has a T_{M} in the range T_{M}^- to T_{M}^+.

$$T_{\mathrm{M}} = \frac{\Delta H^\circ}{R \ln (C_{\mathrm{T}}/\alpha)} + \Delta S^\circ, \qquad \mathbf{1}$$

where R is the gas constant, C_{T} is the concentration, ΔH° is the enthalpy and ΔS° is the entropy. Parameter α is set to 1 for self-complementary and to 4 for non-self-complementary. Parameters ΔH° and ΔS° are calculated based on the nearest-neighbor model (18,19).

ΔG_{gre} filter

This filter checks whether ΔG_{gre} is greater than $\Delta G_{\mathrm{gre}}^*$ for all the combinations as mentioned in Algorithm.

Using a 'greedy search' reduces the computation time for calculating $\Delta G_{\min}$. The greedy search works well because a structure with $\Delta G_{\min}$ tends to include stable helices (i.e. continuous complementary regions). Therefore, the greedy search approximates $\Delta G_{\min}$ by iteratively searching for the most stable helix and fixing the helix.

The greedy search algorithm is as follows:

1. First, calculate the free energies of all helices over the structure between sequence $s_0 s_1 \cdots s_{N-1}$ and sequence $t_0 t_1 \cdots t_{M-1}$. Calculate the free energies of the helices with free ends using the following equations. Here, a helix is denoted by $s_i s_{i+1} \cdots s_k$ $(5' \rightarrow 3')$ and $t_j t_{j+1} \cdots t_{l(=j+k-i)}$ $(3' \rightarrow 5')$.
   - $\Delta G_{\mathrm{freeEnd}} = \Delta G_{\mathrm{core}} + D(s_0, s_i, t_0, t_j) + D(t_{M-1}, t_l, s_{N-1}, s_k)$
   - $\Delta G_{\mathrm{core}} = \sum_{a=i}^{k-1} eS(s_a, s_{a+1}, t_{j+a-i}, t_{j+a-i+1})$

   $\Delta G_{\mathrm{freeEnd}}$ is the free energy of a helix with free ends; ΔG_{core} is that without free ends. $D(s_0, s_i, t_0, t_j)$ represents the free energy contribution of the dangling or free end between sequence $5'$-$s_0 s_1 \cdots s_i$-$3'$ and sequence $3'$-$t_0 t_1 \cdots t_j$-$5'$ closing with base pair $s_i \cdot t_j$. For $(i = 0) \wedge (j = 0)$, $D(s_0, s_i, t_0, t_j)$ is zero because there is no dangling or free end. $eS(s_a, s_{a+1}, t_{j+a-i}, t_{j+a-i+1})$ is the free energy of the stacked base pair between sequence $5'$-$s_a s_{a+1}$-$3'$ and sequence $3'$-$t_{j+a-i} t_{j+a-i+1}$-$5'$.
2. Search for a minimal value of $\Delta G_{\mathrm{freeEnd}}$ over all helices. Then, fix the base pairs in the helix where $\Delta G_{\mathrm{freeEnd}}$ is minimum. This region is freshly denoted as $s_i s_{i+1} \cdots s_k$ $(5' \rightarrow 3')$ and $t_j t_{j+1} \cdots t_{l(=j+k-i)}$ $(3' \rightarrow 5')$. Furthermore, the free energies of the regions with and without free ends are freshly denoted as $\Delta G_{\mathrm{freeEnd}}$ and ΔG_{core}, respectively.
3. Iterate the following procedure d times, where d is a parameter defined below.
   (a) If $(0 < i-1) \wedge (0 < j-1)$ holds, calculate the free energies of all helices with the loop closing with base pair $s_i \cdot t_j$ over the structure between sequence $s_0 s_1 \cdots s_{i-1}$ and the sequence $t_0 t_1 \cdots t_{j-1}$. Thus, the free energy of helix, denoted as $s_{i'} s_{i'+1} \cdots s_{k'}$ $(5' \rightarrow 3')$ and $t_{j'} t_{j'+1} \cdots t_{l'(=j'+k'-i')}$ $(3' \rightarrow 5')$, is calculated as follows:
   - $\Delta G^L_{\mathrm{freeEnd}} = \Delta G^L_{\mathrm{core}} + D(s_0, s_{i'}, t_0, t_{j'})$
   - $\Delta G^L_{\mathrm{core}} = \sum_{a=i'}^{k'-1} eS(s_a, s_{a+1}, t_{j'+a-i'}, t_{j'+a-i'+1}) + eL(s_{k'}, s_i, t_{l'}, t_j)$,

   where $\Delta G^L_{\mathrm{freeEnd}}$ is the free energy of the helix with a free end, $\Delta G^L_{\mathrm{core}}$ is that of one without a free end and $eL(s_{k'}, s_i, t_{l'}, t_j)$ is the free energy contribution of a bulge or internal loop between sequence $s_{k'} s_{k'+1} \cdots s_i$ and sequence $t_{l'} t_{l'+1} \cdots t_j$ closing with base pairs $s_{k'} \cdot t_{l'}$ and $s_i \cdot t_j$.
   (b) Search for a region where $\Delta G^L_{\mathrm{freeEnd}}$ is minimum over all helices. This region is freshly denoted as $s_{i'} s_{i'+1} \cdots s_{k'}$ $(5' \rightarrow 3')$ and $t_{j'} t_{j'+1} \cdots t_{l'(=j'+k'-i')}$ $(3' \rightarrow 5')$. Furthermore, the free energies of the regions with and without a free end are freshly denoted as $\Delta G^L_{\mathrm{freeEnd}}$ and $\Delta G^L_{\mathrm{core}}$, respectively. If $\Delta G^L_{\mathrm{freeEnd}} \leqslant D(s_0, s_i, t_0, t_j)$ holds, fix the base pairs, $s_{i'} \cdot t_{j'}, s_{i'+1} \cdot t_{j'+1}, \ldots, s_{k'} \cdot t_{l'}$, and update i, j and ΔG_{core} to i', j' and $\Delta G_{\mathrm{core}} + \Delta G^L_{\mathrm{core}}$, respectively. This means that the base pairs in the region are energetically favorable.
   (c) If $(k+1 < N-1) \wedge (l+1 < M-1)$ holds, calculate the free energies of all helices with a loop closing with base pair $s_k \cdot t_l$ over the structure between sequence $s_{k+1} s_{k+2} \cdots s_{N-1}$ and sequence $t_{l+1} t_{l+2} \cdots t_{M-1}$. Thus, the free energy of helix, denoted as $s_{i''} s_{i''+1} \cdots s_{k''}$ $(5' \rightarrow 3')$ and $t_{j''} t_{j''+1} \cdots t_{l''(=j''+k''-i'')}$ $(3' \rightarrow 5')$, is calculated as follows:
   - $\Delta G^R_{\mathrm{freeEnd}} = \Delta G^R_{\mathrm{core}} + D(t_{M-1}, t_{l''}, s_{N-1}, s_{k''})$
   - $\Delta G^R_{\mathrm{core}} = \sum_{a=i''}^{k''-1} eS(s_a, s_{a+1}, t_{j''+a-i''}, t_{j''+a-i''+1}) + eL(s_k, s_{i''}, t_l, t_{j''})$,

   where $\Delta G^R_{\mathrm{freeEnd}}$ is the free energy of the helix with a free end and $\Delta G^R_{\mathrm{core}}$ is that of one without a free end.
   (d) Search for a region where $\Delta G^R_{\mathrm{freeEnd}}$ is minimum over all helices. This region is freshly denoted as $s_{i''} s_{i''+1} \cdots s_{k''}$ $(5' \rightarrow 3')$ and $t_{j''} t_{j''+1} \cdots t_{l''(=j''+k''-i'')}$ $(3' \rightarrow 5')$. Furthermore, the free energies of the regions with and without a free end are freshly denoted as $\Delta G^R_{\mathrm{freeEnd}}$ and $\Delta G^R_{\mathrm{core}}$, respectively. If $\Delta G^R_{\mathrm{freeEnd}} \leqslant D(t_{M-1}, t_l, s_{N-1}, s_k)$ holds, fix the base pairs, $s_{i''} \cdot t_{j''}, s_{i''+1} \cdot t_{j''+1}, \ldots, s_{k''} \cdot t_{l''}$, and update k, l and ΔG_{core} to k'', l'' and $\Delta G_{\mathrm{core}} + \Delta G^R_{\mathrm{core}}$, respectively. This means that the base pairs in the region are energetically favorable.
4. Calculate ΔG_{gre} using

$$\Delta G_{\mathrm{gre}} = \Delta G_{\mathrm{core}} + D(s_0, s_i, t_0, t_j) + D(t_{M-1}, t_l, s_{N-1}, s_k) + \mathrm{init},$$

   where init is the energy penalty for forming double-stranded DNA. If $\Delta G_{\mathrm{gre}} > 0$, however, set $\Delta G_{\mathrm{gre}} = 0$. This is because the free energies are calculated relative to two non-interacting sequences, for which the free energy is defined as zero. Thus, two sequences must remain separate with zero free energy rather than form a structure with positive free energy.

In the above procedure, the number of iterations for a search, d, is defined as 'degree'. A structure with degree = 0 has at most one helix. Note that base pairing does not occur during a greedy search when more than two continuous complementary bases do not exist between two sequences. For degree = 1, there are at most three helices. Eventually, for degree = d, there are at most $(1 + 2 \cdot d)$ helices. If the length of sequences S and T are l and $2 \cdot l$, respectively, the number of iterations is at most $(l - 1)/3$. Thus, the time complexity of greedy search $O(\mathrm{degree} \cdot l^2)$ is $O(l^3)$ at worst. However, because the degree increases slowly with the sequence length, the time complexity of a greedy search can be in practice regarded as $O(l^2)$. To confirm this, we calculated the degree for 10 000 random pairs of sequences from 10mer to 100mer in steps of 10mer. As shown in Table 1, degree was at most 9 at 80mer and 90mer, while it was 33 [=(100 − 1)/3]) at 100mer in the worst case. Furthermore, degree was rarely >6 (<1%), and, for >98% of the 10 000 random pairs, degree was in the range 0–4 at any length. Therefore, degree was nearly constant regardless of the sequence length on average, indicating that the time complexity of a greedy search is $O(l^2)$ in practice.

ΔG_{min} filter

This filter checks whether the ΔG_{min} is greater than $\Delta G^*_{\mathrm{min}}$ for all the combinations as mentioned in Algorithm.

ΔG_{min} between S and T can be decomposed into two terms:

$$\Delta G_{\mathrm{min}} = \min_{\substack{0 \leqslant i \leqslant N-1 \\ 0 \leqslant j \leqslant M-1}} \{D(s_0, s_i, t_0, t_j) + V(s_i, t_j)\}, \qquad \mathbf{2}$$

where $V(s_i, t_j)$ represents the minimum value of the free energy between sequence $s_i s_{i+1} \cdots s_{N-1}$ ($5' \rightarrow 3'$ direction) and sequence $t_j t_{j+1} \cdots t_{M-1}$ ($3' \rightarrow 5'$ direction) closing with $s_i \cdot t_j$. Recall that $D(s_0, s_i, t_0, t_j)$ represents the free energy of the dangling or free ends between sequence $s_0 s_1 \cdots s_i$ ($5' \rightarrow 3'$ direction) and sequence $t_0 t_1 \cdots t_j$ ($3' \rightarrow 5'$ direction) closing with base pair $s_i \cdot t_j$.

Table 1. Distribution of degree for greedy search for 10 000 random pairs of sequences from 10mer to 100mer in steps of 10mer

Degree	Sequence length (mer)									
	10	20	30	40	50	60	70	80	90	100
0	9426	7941	6679	5813	5032	4544	3973	3575	3196	2931
1	560	1838	2783	3238	3638	3723	3953	3856	3861	3863
2	14	204	471	767	1040	1301	1467	1752	1959	2067
3	0	17	57	161	244	336	458	596	698	770
4	0	0	10	16	38	78	106	165	206	242
5	0	0	0	4	7	13	28	43	48	89
6	0	0	0	1	1	5	15	12	24	30
7	0	0	0	0	0	0	0	0	4	7
8	0	0	0	0	0	0	0	0	3	1
9	0	0	0	0	0	0	0	1	1	0

Furthermore, $V(s_i, t_j)$ is calculated using

$$V(s_i,t_j) = \min\{D(t_{M-1}, t_j, s_{N-1}, s_i), eS(s_i, s_{i+1}, t_j, t_{j+1}) + V(s_{i+1}, t_{j+1}), \mathrm{VBI}(s_i, t_j)\}, \quad \mathbf{3}$$

where $\mathrm{VBI}(s_i, t_j)$ represents the minimum value of the free energy forming a bulge or internal loops closing with base pair $s_i \cdot t_j$. Recall that $eS(s_i, s_{i+1}, t_j, t_{j+1})$ is the free energy of the stacked base pair between sequence $s_i s_{i+1}$ ($5' \rightarrow 3'$ direction) and sequence $t_j t_{j+1}$ ($3' \rightarrow 5'$ direction). The first term represents the case in which the unpaired end consists of sequences $t_{M-1} t_{M-2} \cdots t_j$ ($5' \rightarrow 3'$ direction) and $s_{N-1} s_{N-2} \cdots s_i$ ($3' \rightarrow 5'$ direction) closing with base pair $t_j \cdot s_i$. The second term corresponds to the case in which the stacked base pair is energetically favorable. In this case, $V(s_{i+1}, t_{j+1})$ is calculated recursively. The third term is calculated using

$$\mathrm{VBI}(s_i, t_j) = \min_{\substack{i<i', j<j' \\ i'-i+j'-j>2}} \{eL(s_i, s_{i'}, t_j, t_{j'}) + V(s_{i'}, t_{j'})\}. \quad \mathbf{4}$$

Recall that $eL(s_i, s_{i'}, t_j, t_{j'})$ represents the free energy contribution of the loops between sequence $s_i s_{i+1} \cdots s_{i'}$ and sequence $t_j t_{j+1} \cdots t_{j'}$ closing with base pairs $s_i \cdot t_j$ and $s_{i'} \cdot t_{j'}$. The $\Delta G_{\min}$ is calculated recursively using Equations 2–4 by dynamic programming. If we compute the VBI term in a straightforward manner, its time complexity is $O(l^4)$. However, this can be reduced to $O(l^3)$ using the algorithm of Lyngsø *et al*. (23).

RESULTS

Effectiveness of ΔG_{gre} filter

To evaluate the effectiveness of the ΔG_{gre} filter, we compared the computation time of our algorithm with that of the algorithm without the ΔG_{gre} filter. That algorithm checks a randomly generated sequence by using the T_{M} and $\Delta G_{\min}$ filters and then stores the sequence in the pool if and only if the sequence passes both filters. All the computational experiments described in this section were performed using Windows 2000 on a computer with an Athlon 1.4 GHz CPU and 256 MB of memory. The results are shown in Figure 1. The computation time grew exponentially because the number of three-sequence combinations increased exponentially with the number of sequences.

Figure 1 shows that using our algorithm reduced the computation time drastically. For example, our algorithm needed ~1.3 h to design 30 sequences with length 20 for $\Delta G^*_{\min} = -10.0$, while the algorithm without the ΔG_{gre} filter needed ~7.7 h (Figure 1a and b). This means that the ΔG_{gre} filter effectively excludes the sequences that cannot pass through the $\Delta G_{\min}$ filter.

A comparison of Figure 1a and b clearly shows the primacy of our algorithm for $\Delta G^*_{\min} = -10.0$ versus $\Delta G^*_{\min} = -13.0$. For instance, our algorithm reduced computation time to 83% (7.7 → 1.3 h) for 30 sequences and $\Delta G^*_{\min} = -10.0$ while it reduced computation time to 57% (9.1 → 3.9 h) for 69 sequences and $\Delta G^*_{\min} = -13.0$. This is because both algorithms can find sequences that satisfy the constraints more easily for $\Delta G^*_{\min} = -13.0$ than for $\Delta G^*_{\min} = -10.0$. A similar trend is seen in the sequences with length 15 (Figure 1c and d). Therefore, using the ΔG_{gre} filter enables sequences to be designed in less time, particularly when the threshold is high.

Filtering performance of ΔG_{gre} filter versus hamming distance

The function of the ΔG_{gre} filter is to filter out the inappropriate sequences from many sequences generated randomly; the promising sequences are then checked using the $\Delta G_{\min}$ filter. This means that the filtering performance of the ΔG_{gre} filter can be evaluated using four terms: the number of sequences with both $\Delta G_{\min}$ and ΔG_{gre} higher than the threshold (true positive: TP), that with $\Delta G_{\min}$ higher but not with ΔG_{gre} (false negative: FN), that with ΔG_{gre} higher but not with $\Delta G_{\min}$ (false positive: FP) and that with neither $\Delta G_{\min}$ nor with ΔG_{gre} higher (true negative: TN) (Figure 2). There is a trade-off between FN and FP, which are the number of sequences incorrectly excluded by the ΔG_{gre} filter and that incorrectly passing through the ΔG_{gre} filter, respectively. To evaluate the ΔG_{gre} filter, we compared the filtering performance of the ΔG_{gre} filter with that using the hamming distance (called hamming filter) with respect to FN such that FP = 0 and to FP such that FN = 0 for 10 000 random pairs of sequences with length 20. Setting FP to 0 means that the ΔG_{gre} or hamming filter certainly excludes inappropriate sequences with a $\Delta G_{\min}$ less than or equal to $\Delta G^*_{\min}$. Therefore, the lower the number in FN such that FP = 0, the better the filter. Similarly, because FN = 0 means that the ΔG_{gre} or hamming filter never excludes appropriate sequences with a $\Delta G_{\min}$ greater than $\Delta G^*_{\min}$, the filter should have fewer sequences in FP such that FN = 0.

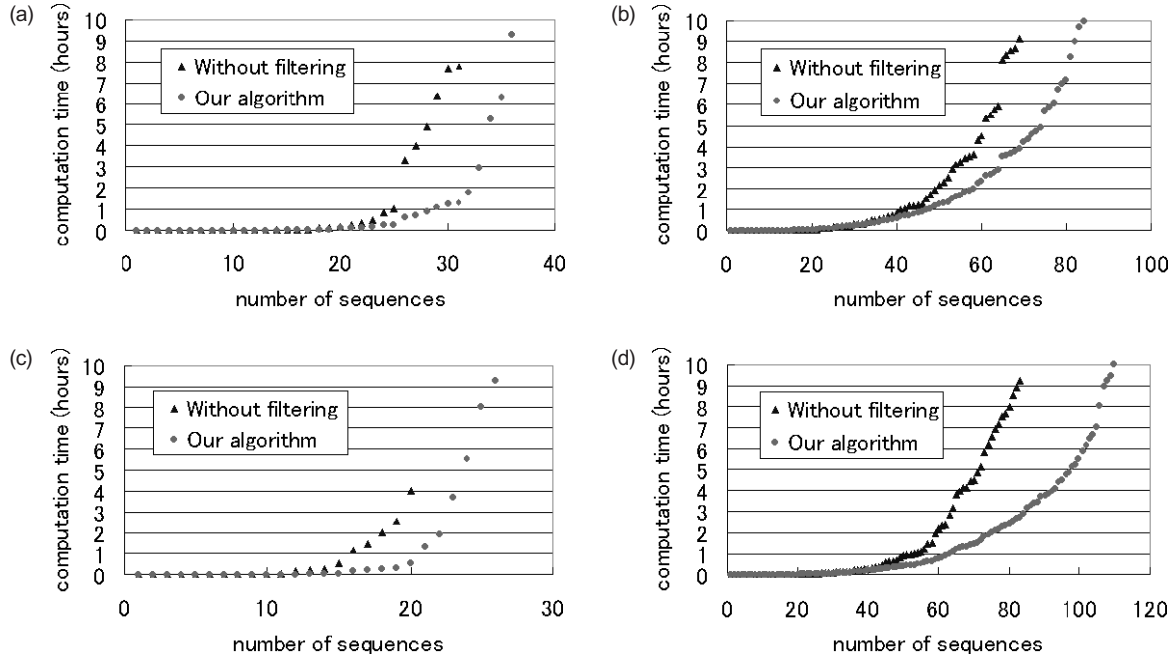

Figure 1. Number of sequences designed versus computation time for two design strategies up to 10 h. In (a) and (b), $l = 20$, $T_M^- = 69.58$ and $T_M^+ = 72.58$. In (**a**), $\Delta G^*_{gre} = \Delta G^*_{min} = -10.0$; and in (**b**), $\Delta G^*_{gre} = \Delta G^*_{min} = -13.0$. In (c) and (d), $l = 15$, $T_M^- = 60.49$ and $T_M^+ = 63.49$. In (**c**), $\Delta G^*_{gre} = \Delta G^*_{min} = -7.0$; and in (**d**), $\Delta G^*_{gre} = \Delta G^*_{min} = -11.0$.

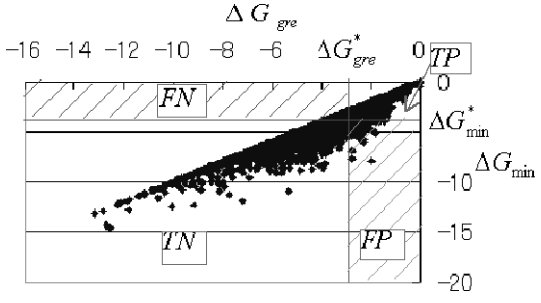

Figure 2. Plot of ΔG_{gre} versus ΔG_{min} for 10 000 random pairs of sequences with length 20; ΔG_{gre*} and ΔG_{min*} represent threshold of ΔG_{gre} and that of ΔG_{min}, respectively. Four terms were used for evaluating filtering performance: TP, $\{(\Delta G_{gre*} < \Delta G_{gre}) \wedge (\Delta G_{min*} < \Delta G_{min})\}$; FN, $\{(\Delta G_{gre} \leqslant \Delta G_{gre*}) \wedge (\Delta G_{min*} < \Delta G_{min})\}$; FP, $\{(\Delta G_{gre*} < \Delta G_{gre}) \wedge (\Delta G_{min} \leqslant \Delta G_{min*})\}$; and TN, $\{(\Delta G_{gre} \leqslant \Delta G_{gre*}) \wedge (\Delta G_{min} \leqslant \Delta G_{min*})\}$.

As shown in Figure 3, for FN = 0, the number of sequences classified into FP by the ΔG_{gre} filter was much smaller than that by the hamming filter. Because most sequences have a ΔG_{min} of more than -10.0 kcal/mol (Figure 2), the FP converged to zero for both the hamming filter and ΔG_{gre} filter for less than -10.0 kcal/mol. Similarly, for FP = 0, the number of sequences classified into FN by the ΔG_{gre} filter was smaller than that by the hamming filter. The reason there is little difference for greater than -5.0 kcal/mol is that a small number of sequences with the ΔG_{gre} actually had a ΔG_{min} such that $\Delta G_{min} \ll \Delta G_{gre}$. For example, sequences GGTCACCCTGGGCTACCGGA (5′ → 3′ direction) and TTAAGGTCGCGTGCTATCTT (3′ → 5′ direction) had a ΔG_{min} of -7.92 kcal/mol while the ΔG_{gre} was -1.96 kcal/mol. Because such sequences are rare, however, the primacy of the ΔG_{gre} filter over the hamming filter is clear for less than -5.0 kcal/mol.

Another advantage of the ΔG_{gre} filter is that it guarantees having the threshold where the number of FN is zero because the ΔG_{gre} is the upper bound for the ΔG_{min} (i.e. $\Delta G_{min} \leqslant \Delta G_{gre}$). For example, a pair of sequences having a ΔG_{gre} less than or equal to -5.0 kcal/mol is guaranteed to have a ΔG_{min} of at most -5.0 kcal/mol. Thus, the sequences with a ΔG_{gre} of more than -5.0 kcal/mol include all sequences with a ΔG_{min} of more than -5.0 kcal/mol. Therefore, setting the threshold for a ΔG_{gre} filter (ΔG^*_{gre}) to that for the ΔG_{min} filter (ΔG^*_{min}) (i.e. $\Delta G^*_{gre} = \Delta G^*_{min}$) guarantees that the number of FN is zero. This is not the case for hamming filter.

Comparison between hamming distance and ΔG_{min} *in vitro*

We investigated the validity of the ΔG_{min} based approach compared with the traditional approach based on the hamming distance by using an *in vitro* experiment. First, to address the validity of the ΔG_{min} calculation, we calculated the average deviations from the experimental free energies (ΔG_{exp}). The average deviations were derived from 126 sequences (31 complementary sequences, 83 sequences with a single bulge loop and 12 sequences with free ends). The average deviations, calculated using $\frac{1}{126}\sum_{i=1}^{126} |100(\Delta G^i_{min} - \Delta G^i_{exp})\Delta/G^i_{exp}|$ (ΔG^i_{min} and ΔG^i_{exp} represent the i-th ΔG_{min} and ΔG_{exp}, respectively), were 3.2% (3.0, 2.8 and 6.5% for the complementary sequences, single bulges and free ends, respectively). These average deviations are within the limits of what can be expected for a nearest-neighbor model (18,19). Thus, we confirmed that our algorithm can predict the ΔG_{min} adequately. The 126 sequences with their ΔG_{min} and ΔG_{exp} are provided in the Supplementary Material. The number of complementary bases for each pair of sequences is also provided.

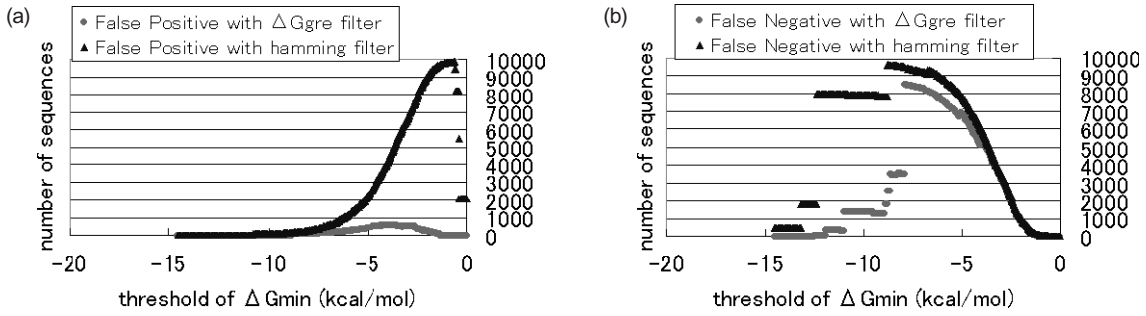

Figure 3. Threshold of ΔG_{min} versus number of sequences classified as FP and FN. (**a**) FP such that FN is zero and (**b**) FN such that FP is zero.

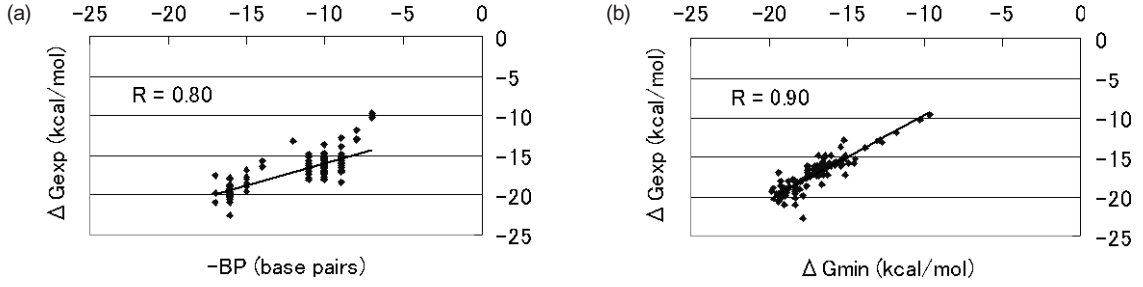

Figure 4. (**a**) −BP versus experimental free energy. BP represents the number of complementary bases. (**b**) Predicted minimum versus experimental free energy. In (a) and (b), values and lines are correlation coefficients and regression lines, respectively.

Then, to compare the hamming distance with ΔG_{min}, we investigated the correlation coefficient with ΔG_{exp}. To fairly compare different length sequences, we used the number of complementary bases but not the hamming distance. For example, although sequences 5′-GGG-3′ and 3′-CCC-5′ with three complementary bases and sequences 5′-GGGGG-3′ and 3′-CCCCC-5′ with five complementary bases both have zero hamming distance, the stability of the latter must be higher than that of the former. Thus, the number of complementary bases is appropriate for comparing the stability of sequences with different lengths. Note that the number of complementary bases is equivalent to the hamming distance for sequences with the same length. The results are shown in Figure 4a and b. In Figure 4a, BP represents the number of complementary bases. The correlation coefficient, $|R|$, between −BP and ΔG_{exp} was 0.80, while that between ΔG_{min} and ΔG_{exp} was 0.90, indicating that the ΔG_{min} is better than the number of complementary bases (i.e. hamming distance) as a predictor of stability.

Bozdech *et al*. (5) compared the number of complementary bases with the binding energy (the calculation method and nearest-neighbor parameters differed from ours) by using 70mer oligonucleotides on microarray. They found that $|R|$ between the binding energy and relative intensity of hybridization (= intensity of fluorescence) was 0.91, while it was 0.72 between the number of complementary bases and the relative intensity of hybridization. This is consistent with our findings, especially with respect to the correlation between the predicted energy (binding energy in theirs, ΔG_{min} in ours) and the stability derived from the experimental results (their $|R|$ was 0.91, while ours was 0.90). With respect to the correlation between the number of complementary bases and the experimental stability, their $|R|$ was 0.72, while ours was 0.80. This is because our data were derived from only sequences with a simple structure, i.e. with at most one loop (single bulge loop). In general, the more loops, the higher the discrepancy between the number of complementary bases and the experimental stability. Therefore, the correlation between the number of complementary bases and the experimental stability will be close to theirs with respect to more complex sequences, i.e. with more than two loops. These results demonstrate the ability of our program to approximate the experimental stability of double-stranded DNA based on the ΔG_{min} calculation.

DISCUSSION

In the previous section, we showed the effectiveness of the ΔG_{gre} filter for filtering out the sequences that cannot pass through the ΔG_{min} filter. To further address the question how the ΔG_{gre} is close to the ΔG_{min}, we investigated the average prediction error of ΔG_{gre}, calculated using $\sum_{i=1}^{10\,000}(G^i_{gre}-G^i_{min})/10\,000$. Figure 5 shows that the average prediction error can be restricted to < 1.0 kcal/mol for sequences shorter than 50mer, which are frequently used in DNA computing.

Figure 6 shows the number of sequences such that ΔG_{min} is equal to ΔG_{gre}. The number decreased as the sequences became longer. For example, $\Delta G_{min} = \Delta G_{gre}$ in 9250 pairs from 10 000 pairs of sequences at 10mer, while $\Delta G_{min} = \Delta G_{gre}$ in 157 pairs from 10 000 pairs at 100mer. Therefore, ΔG_{gre} is a good predictor of ΔG_{min} for short sequences, while it is only an approximation for long sequences.

For comparison with the hamming distance, the correlation coefficient ($|R|$) was also calculated. The correlation between

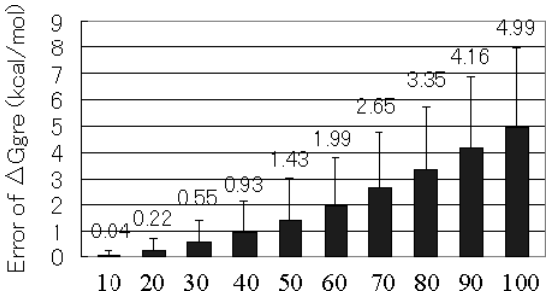

Figure 5. Average prediction error of ΔG_{gre}, calculated using $\sum_{i=1}^{10\,000}(\Delta G^{i}_{gre}-\Delta G^{i}_{min})/10\,000$.

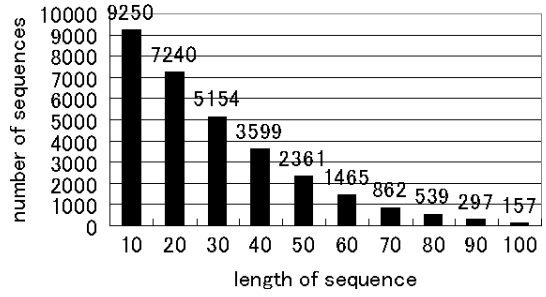

Figure 6. Number of sequences such that $\Delta G_{min} = \Delta G_{gre}$.

ΔG_{min} and ΔG_{gre} decreased almost linearly from 0.99 at 10mer to 0.63 at 100mer. Because long sequences tend to have more helices, the discrepancy increased as the sequences became longer. The correlation coefficient between ΔG_{min} and the hamming distance, which decreased almost linearly from 0.46 at 10mer to 0.12 at 100mer, was much less than that between ΔG_{min} and ΔG_{gre}.

IMPLEMENTATION

We implemented our algorithm in a program called 'DNA Sequence Design Tool (DNA-SDT)'. DNA-SDT can be downloaded freely from the web site (http://ses3.complex.eng.hokudai.ac.jp/~fumi95/DNA-SDT/index.html). It has two functions: sequence design and structure prediction.

In sequence design, the program solves the problem of designing a pool P containing n sequences of length l for which (i) the duplex T_M is in the range T_M^- to T_M^+ for any duplex of U_i $(0 \leqslant i \leqslant n-1)$ and V_i and (ii) ΔG_{min} is greater than ΔG^*_{min} in any pairwise duplex of sequences in P and any concatenation of two sequences in $P \cup Q$ except for the pairwise duplex of U_i and V_i. This problem is solved using our algorithm with the threshold for a ΔG_{gre} filter, ΔG^*_{gre}. The parameters, n, l, T_M^-, T_M^+, ΔG^*_{gre} and ΔG^*_{min}, are user-defined. DNA-SDT has a table of average T_M values at length l $(8 \leqslant l \leqslant 50)$ $[T_M^{ave}(l)]$ calculated from 10 000 sequences generated randomly. Parameters T_M^- and T_M^+ are set to $[T_M^{ave}(l)-1.5]$ and $[T_M^{ave}(l)+1.5]$, respectively, by default. The T_M is calculated at 1 μM.

In structure prediction, the program calculates the ΔG_{min} between two sequences, then displays the structure with the ΔG_{min} using a traceback algorithm. Users can thus determine the structure of any combination of sequences from the pool designed by the program. Of course, for any given combination of sequences, users can also determine the structure with the ΔG_{min}.

The program is a GUI-based application written in Java, hence it can be executed on any computer that has the Java Runtime Environment (JRE) installed. Although older versions of JRE supposedly run without problem, the latest version is preferable. We tested the program with JRE 1.4.2 on a Windows 2000 workstation, JRE 1.4.1 on a Windows XP workstation and JRE 1.4.2 on a Turbolinux Workstation 7.0.

The program interface consists of two screens: one for structure prediction and one for sequence design. The left-hand side of the window is the screen for structure prediction, which outputs the structure with the ΔG_{min} from the two sequences input in the text box. The sequences in the text box can be converted into reverse or complementary sequences. The right-hand side is the screen for sequence design, which outputs the sequences, GC content and T_M based on the constraints input in the text boxes.

To avoid impossible operations during design, the procedure for sequence design is executed as a separate thread. Therefore, the structure-prediction screen can be operated freely when sequences are being designed. Furthermore, the design procedure can be monitored and interrupted anytime, and the interim results can be output. However, because the button for running the sequence design procedure is locked during design, users cannot design another pool of sequences in parallel.

The number of sequences can be set up to 100. If more than 100 sequences need to be designed, the user can choose 'Unlimited'. In this case, the design procedure iterates until the user interrupts it. The sequence length is restricted at 8mer to 50mer because of the computation time. Although the thresholds for ΔG_{min} and ΔG_{gre} can be set freely, using irrelevant values resulting in zero sequences. We recommend setting these thresholds such that $\Delta G^*_{min} = \Delta G^*_{gre}(< 0)$ to guarantee zero false negatives (see Results).

SUPPLEMENTARY MATERIAL

Supplementary Material is available at NAR Online.

ACKNOWLEDGEMENT

Funding to pay the Open Access publication charges for this article was provided by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research on Priority Areas, 2002–2005, 14085201.

REFERENCES

1. Adleman,L.M. (1994) Molecular computation of solutions to combinatorial problems. *Science*, **266**, 1021–1024.
2. Braich,R.S., Chelyapov,N., Johnson,C., Rothemund,P.W. and Adleman,L. (2002) Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, **296**, 499–502.
3. Faulhammer,D., Cukras,A.R., Lipton,R.J. and Landweber,L.F. (2000) Molecular computation: RNA solutions to chess problems. *Proc. Natl Acad. Sci. USA*, **97**, 1385–1389.
4. Rouillard,J.M., Zuker,M. and Gulari,E. (2003) OligoArray 2.0: design of oligonucleotide probes for DNA microarrays using a thermodynamic approach. *Nucleic Acids Res.*, **31**, 3057–3062.

5. Bozdech,Z., Zhu,J., Joachimiak,M.P., Cohen,F.E., Pulliam,B. and DeRisi,J.L. (2003) Expression profiling of the schizont and trophozoite stages of *Plasmodium falciparum* with a long-oligonucleotide microarray. *Genome Biol*., **4**, R9.
6. Kane,M.D., Jatkoe,T.A., Stumpf,C.R., Lu,J., Thomas,J.D. and Madore,S.J. (2000) Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays. *Nucleic Acids Res*., **28**, 4552–4557.
7. Winfree,E., Liu,F., Wenzler,L.A. and Seeman,N.C. (1998) Design and self-assembly of two-dimensional DNA crystals. *Nature*, **394**, 539–544.
8. Shih,W.M., Quispe,J.D. and Joyce,G.F. (2004) A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, **427**, 618–621.
9. Yan,H., Zhang,X., Shen,Z. and Seeman,N.C. (2002) A robust DNA mechanical device controlled by hybridization topology. *Nature*, **415**, 62–65.
10. Arita,M. and Kobayashi,S. (2002) DNA sequence design using templates. *New Gen. Comput*., **20**, 263–277.
11. Arita,M., Nishikawa,A., Hagiya,M., Komiya,K., Gouzu,H. and Sakamoto,K. (2000) Improving sequence design for DNA computing. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, July 8–12, Las Vegas, NV. Morgan Kaufmann, pp. 875–882.
12. Dirks,R.M., Lin,M., Winfree,E. and Pierce,N.A. (2004) Paradigms for computational nucleic acid design. *Nucleic Acids Res*., **32**, 1392–1403.
13. Andronescu,M., Fejes,A.P., Hutter,F., Hoos,H.H. and Condon,A. (2004) A new algorithm for RNA secondary structure design. *J. Mol. Biol*., **336**, 607–624.
14. Andronescu,M., Aguirre-Hernandez,R., Condon,A. and Hoos,H.H. (2003) RNAsoft: a suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Res*., **31**, 3416–3422.
15. Hofacker,I.L. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res*., **31**, 3429–3431.
16. Mathews,D.H., Sabina,J., Zuker,M. and Turner,D.H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol*., **288**, 911–940.
17. Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*., **9**, 133–148.
18. SantaLucia,J.,Jr (1998) A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl Acad. Sci. USA*, **95**, 1460–1465.
19. Tanaka,F., Kameda,A., Yamamoto,M. and Ohuchi,A. (2004) Thermodynamic parameters based on a nearest-neighbor model for DNA sequences with a single-bulge loop. *Biochemistry*, **43**, 7143–7150.
20. Zuker,M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res*., **31**, 3406–3415.
21. Peritz,A.E., Kierzek,R., Sugimoto,N. and Turner,D.H. (1991) Thermodynamic study of internal loops in oligoribonucleotides: symmetric loops are more stable than asymmetric loops. *Biochemistry*, **30**, 6428–6436.
22. Bommarito,S., Peyret,N. and SantaLucia,J.,Jr (2000) Thermodynamic parameters for DNA sequences with dangling ends. *Nucleic Acids Res*., **28**, 1929–1934.
23. Lyngsø,R.B., Zuker,M. and Pedersen,C.N.S. (1999) Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, **15**, 440–445.