Strand Design for Bio-Molecular Computation

Arwen Brenneman and Anne E. Condon
The Department of Computer Science
University of British Columbia
2366 Main Mall, Vancouver, B.C. V6T 1Z4

March 22, 2001

Abstract

The design of DNA or RNA strands for DNA computations poses many new questions in algorithms and coding theory. DNA strand design also arises in use of molecular bar codes to manipulate and identify individual molecules in complex chemical libraries, and to attach molecules to DNA chips. We survey several formulations of the DNA strand design problem, along with results and open questions in this area.

1 Introduction

DNA molecules are now used for purposes that go far beyond their functions in nature. DNA chips - arrays of short, synthetic DNA strands - provide a means for sequencing, disease diagnosis, and gene expression analysis [5, 53, 54]. DNA tags label molecules in chemical libraries [6, 7]. Tiny instances of combinatorial problems have been solved in a wet-lab, using DNA or RNA to represent a pool of solutions to a problem instance [1]. Novel topological and rigid three-dimensional structures have been built from DNA [51, 59].

The ability to use DNA and RNA in these ways rests on solving many challenging research questions not only in chemistry but also in computer science and mathematics. In this article, we focus on one of these challenges, which draws on the fields of combinatorial algorithms and coding theory, namely strand design.

We use two simple analogies to describe intuitively what strand design means. First, consider a classical problem from coding theory: find the maximum size set of equal-length binary strings such that the Hamming distance between each pair of strings is above some threshold. Such sets of "code words" are useful in detection and correction of errors when transmitting information over a noisy channel. Now, replace binary Hamming distance by an appropriate combinatorial measure of distance between pairs of strands (which can be viewed as strings over a 4-letter alphabet). This form of the strand design problem arises in what we call classical DNA computation, as exemplified by Adleman's pioneering work [1]: A large pool of DNA strands, each representing a possible solution to a combinatorial problem, is pruned in a sequence of steps until all non-solutions are weeded out. Strand design is relevant to classical DNA computing, and also to applications involving DNA tags, because information-carrying DNA strands are subject to error-prone chemical and enzymatic operations, and it is important that one strand not be mistaken for another.

For our second analogy, think of a quilt as an assembly of individual motifs, each of different shapes and sizes. Each motif is itself woven from threads. Now, replace the quilt by an assembly of small DNA motifs, where each motif is a 2-D or 3-D molecule, "woven" from DNA strands. Winfree et al. [58, 60] showed that DNA self-assembly can in principle be used to perform general computations. Strand design is needed in order to create the individual motifs from their component DNA strands, and later to ensure that the motifs assemble as desired. Just as a slinky gets irreversibly tangled once in a child's hand, so do poorly designed DNA strands fold into an undesired shape, making this design task challenging.

We organize this survey into two main sections, one on strand design for classical computation and one on strand design for self-assembly computation. We note that, particularly for classical computation, there are almost as many formulations of the problem as there are experimental projects for building DNA tags, structures, or computers! Perhaps the most interesting questions that emerge from this survey ask how well the different formulations of the word design ultimately meet the physical and chemical requirements posed by use of the words in DNA computations.

1.1 Background information on DNA and RNA

A single-stranded DNA or RNA molecule is a sequence over four possible nucleotides, which are strung together in a strand like beads on a necklace. Each nucleotide is comprised of three parts; a phosphate group, a ribose group, and a heterocyclic base. Any strand of DNA or RNA is linked by a backbone that is formed by the alternating phosphate and ribose of each nucleotide. This alternating backbone gives each base, and cumulatively the strand, a direction from the ribose end (denoted by $5'$) to the phosphate end (denoted by $3'$). The heterocyclic base gives each nucleotide its identity in the strand. In RNA the bases comprising the nucleotides are Adenine (A), Cytosine (C), Guanine (G), and Uracil (U). DNA also has four bases, including A, C, G and replacing Uracil (U) with Thymine (T).

The heterocyclic bases in RNA and DNA form attractions via hydrogen bonding to other bases, in a process known as hybridization. DNA is best known for double helix bonding. In a double helix, successive pairs of bases in two strands are bonded, starting from the first base at the $3'$ end of one strand and at the $5'$ end of the other. A strand forms the most stable double helix with its Watson-Crick complement. The Watson-Crick complement of a DNA strand is the strand obtained by replacing each A nucleotide by a T and vice versa, each C by a G and vice versa and also switching the $5'$ and $3'$ ends. For example, the Watson-Crick complement of $5'$-$AACATG$-$3'$ is $3'$-$TTGTAC$-$5'$. In an RNA molecule, A is complementary to U instead of T. GC pairs have three hydrogen bonds, and are the most stable, whereas AT pairs have two hydrogen bonds. Other complements may form, such as the "wobble" pair GT, however their chemical bonding is generally less stable than the Watson-Crick complements. The complete list of pair bonds in a DNA or RNA molecule is known as the molecule's **secondary structure**.

Complementary bases within a single DNA or RNA strand may also bond, causing the strand to fold on itself and thus to have a secondary structure. For example, the strand $5'$-$GCAAAAGC$-$3'$ may fold so that the substrand $5'$-GC-$3'$ at the $5'$ end bonds with the substrand $3'$-CG-$5'$ at the $3'$ end. In this case, the unbonded A's between the bonded ends are said to form a hairpin loop. Single strand bonding is common in vivo, in such molecules as tRNA or types of virus molecules [56]. The geometry of these molecules, also known as their tertiary structure, is vital to their function.

Like any chemical reaction, the pair bonding of bases in sequences of RNA or DNA is influenced by environment. Temperature, salinity, and molarity all affect the ability of strands to hybridize. DNA and RNA structures melt into their individual components at higher temperatures. These properties can be exploited for manipulation and construction of sequences in the lab.

2 Word design for classical DNA computations

Short DNA strands, called oligonucleotides, can be quickly and cheaply synthesized and thus can be used to store information at the molecular level. Just as a computer memory is composed of short, equi-length words of information, in a DNA computation a long strand is typically a concatenation of short DNA (or RNA) strands which are the units of information storage and manipulation in a computation. A set of equi-length DNA strands is henceforth referred to as a DNA word set. For example, in order to represent the set of all n-bit strings, the DNA word set S might consist of $2n$ words $s_1, \bar{s}_1, s_2, \bar{s}_2, \ldots, s_n, \bar{s}_n$. The set L of strands in the computation would then be of the form $z_1 z_2 \ldots z_n$ where for $1 \leq i \leq n$, $z_i = s_i$ or $z_i = \bar{s}_i$, yielding a total of 2^n strands.

Retrieval of information stored in DNA words requires success in achieving specific hybridization between a DNA code word and its Watson-Crick complement. For this reason, we want the set of DNA words to form **stable** duplexes with their complements. We also need to ensure that two distinct words are **non-interacting** - that duplexes between pairs of words, and between a word and the Watson-Crick complement of another, are relatively unstable, compared with any perfectly matched duplex formed from a DNA word and its complement.

When multiple words are concatenated to form long DNA strands for a DNA computation, non-interaction requirements become more complex. Let S be a DNA word set and let L be a set of strands formed by concatenating words in S according to certain prescribed rules. We require that each strand in L is unlikely to bond with itself to form unwanted secondary structures. In addition, for each word w in S and strand l in L, we want that duplexes formed from l and the Watson-Crick complement of w are unstable. We note that when short, single-stranded DNA "spacers" are placed between words in the long strands, the non-interaction requirements need to be generalized slightly further to allow for the fact that the spacers may have lengths different than the length of the DNA words, but we shall not consider this detail further in this survey.

DNA word design poses several constraints on the base sequences of words, in order to meet the goals of stability and non-interaction. In the next sections, we enumerate some widely used constraints.

2.1 Stability

Given fixed conditions such as temperature and concentration, a reliable measure of the relative stability of a perfectly matched DNA duplex (i.e. double helix) structure is its free energy, measured in units of kcal/mol of interaction and denoted by ΔG^o. The free energy depends on the base sequence of the duplex [8, 50]. The nearest-neighbour model predicts the free energy as the sum of terms, one per group of nearest neighbour (i.e. consecutive) bases, plus a correction factor (based on whether the duplex is formed from a self-complementary sequence or not, and whether there are any GC base pairs in the duplex). There are 10 possible nearest neighbour groups, namely

AT/TA, TA/AT, CG/GC, CG/CG, AA/TT, CC/GG, GT/CA, GA/CT, AG/TC, and TG/AC. Here, AT/TA represents the duplex $\begin{smallmatrix} 5'\text{-}AT\text{-}3' \\ || \\ 3'\text{-}TA\text{-}5' \end{smallmatrix}$ Similarly GA/CT represents $\begin{smallmatrix} 5'\text{-}GA\text{-}3' \\ || \\ 3'\text{-}CT\text{-}5' \end{smallmatrix}$ and thus GA/CT is identical to TC/AG; this and five other identities result in 10 rather than 16 possibilities. Associated with each nearest neighbour group g is a negative weight $w(g)$. The **free energy** of the duplex formed from the DNA strand $D = 5'\text{-}d_1 d_2 \ldots d_n\text{-}3'$ and its complement $3'\text{-}c_1 c_2 \ldots c_n\text{-}5'$ is given by

$$\delta G^o(D/C) = \text{correction factor} + \sum_{i=1}^{n-1} w(g_i)$$

where g_i is the nearest neighbour group $d_i d_{i+1}/c_i c_{i+1}$. The lower the free energy, the more stable the duplex.

Closely related to the free energy of a perfectly matched DNA duplex structure is its **melting temperature**. When a mixture of DNA duplexes is heated (assuming a fixed concentration of molecules), the weak hydrogen bonds in the duplex molecules tend to break and thus the strands separate. At the melting temperature, 50% of the duplexes have strands separated in equilibrium. The melting temperature can be predicted as a function of the free energy of the duplex and other parameters [57]. Since multiple hybridization reactions involving distinct DNA words may happen in a single step of a DNA computation, it is desirable that the melting temperature of the duplexes created in each reaction be in a small range [10, 19, 61] or above some threshold [3].

A third, much less accurate measure of the stability of a word is its GC content, or fraction of G's and C's. However because it is easy to measure and is amenable to combinatorial analysis, it is often used in constraining DNA words [21, 32, 49, 61]. A related simple estimate of melting temperature for short oligonucleotides is the 2-4 rule [3, 55], which is that the melting temperature of a sequence and its complement is approximately twice the number of A-T base pairs plus 4 times the number of G-C base pairs.

Stability constraints on a DNA word set are thus formulated by requiring that the free energy, melting temperature, or GC content of the set lies in in a small range.

2.2 Non-interaction

Unfortunately, to our knowledge, no general model for predicting the free energy of short duplexes with mismatches is available, although experimentally obtained thermodynamic data is available for certain duplexes (see for example [43, 21]). In design of DNA words in practice, either for use as DNA tags or DNA computations involving concatenations of multiple words, several combinatorial constraints are imposed on the words, with the goal of ensuring non-interaction. We describe word design constraints on single words, pairs of words, and on larger groups of words in the following subsections.

2.2.1 Non-interaction: constraints on single words

Several word designs are over a three rather than four base alphabet, since the absence of one base reduces the number of complementary base pairs in a long strand [4, 10, 32, 41].

When DNA words are used to tag genomic DNA, then the base sequence of each tag should not appear in the genomic DNA [53, 5]. This constraint, which we refer to as the "forbidden subwords" constraint, may arise also in the case that a restriction enzyme is used in manipulation of the DNA, in which case the corresponding restriction site should not appear as part of a DNA tag except where planned. Forbidden subwords are also motivated by the fact that certain base sequences are known to form hairpins (see for example [28]) and thus should be avoided.

2.2.2 Non-interaction: constraints on pairs of words

Each measure described in this section is defined on a pair of equi-length DNA words w = 5'-$w_1 w_2 \ldots w_n$-3' and x = 5'-$x_1 x_2 \ldots x_n$-3'. Constraints are placed on words using some or all of these measures, for example, by requiring that the mismatch distance between pairs of words is above a certain threshold.

- **Mismatch distance:** The mismatch distance of (w, x) is the number of positions i at which w_i and x_{n-i+1} are not complementary.

Most DNA word sets have the property that either (i) for every pair of words (w, x) in the set (where w may equal x), the mismatch distance of (w, x) is above some threshold, or (ii) for every pair (w, x) where w is in the set and x is the Watson-Crick complement of a distinct word in the set, the mismatch distance of (w, x) is above some threshold, or both [1, 6, 11, 12, 13, 21, 22, 23, 53, 54]). Constraint (i) is intended to inhibit interaction (formation of duplexes) between pairs of words in the set, whereas condition (ii) is intended to inhibit interaction between a word in the set and the complement of a distinct word in the set.

- **Mismatch distance adjusted for shifts:** Let w' be a prefix or suffix of w, and x' be a prefix or suffix of x, where w' and x' have the same length $n - i$. Define the shift-adjusted distance of (w', x') to be i plus the mismatch distance of (w', x'). Then the shift-adjusted mismatch distance of pair (w, x) is the minimum shift-adjusted distance of (w', x'), taken over all prefix/suffix pairs (w', x') [12, 21, 22].

- **Length of repeated runs:** A repeated run in a strand is a sequence of identical bases. Some designs limit the maximum length of repeated runs in words and in concatenations of words [4].

- **Subword distance:** The subword distance of (w, x) is the length of the longest strand z that is a subword of both w and x [2, 42, 61].

- Several **variations of the free energy measure**, adapted to pairs of strands (w, x) that are not perfect complements of each other, have also been considered. One, based on the so-called staggered zipper model, is the minimum free energy of all pairs (w', x') for which w' is a subword of w, x' is a subword of x, and w' and x' are complementary [3, 24, 46]. Another variation adapts the free energy formula (correction factor $+\sum_{i=1}^{n-1} w(g_i)$) so that, as before, g_i is the nearest neighbour group $w_i w_{i+1}/x_{n-i+1}x_{n-i}$ and now $w(g_i)$ is zero if w_i and x_{n-i+1} are non-complementary bases or w_{i+1} and x_{n-i} are non-complementary bases [21]. The BIND simulator of Hartemink and Gifford [26] predicts the binding specificity of pairs of oligonucleotides, based on melting temperature calculations that take shifts and mismatches into account.

2.2.3 Non-interaction: constraints on larger groups of words

In cases where long strands are concatenations of DNA words, many designs enforce constraints on fixed-length windows that can arise in any long strand. Although the number of strands may be exponential in the size of the DNA word set, the number of distinct windows is bounded, usually by the square of the size of the word set, since typically the window spans at most two words. Many designs require that windows, pairs of windows, and windows and the complements of DNA words satisfy constraints based on the melting temperature, GC content, mismatch distance, repeated bases, and subwords distance measures listed in sections 2.2.1 and 2.2.2 [4, 10, 19, 48, 54]. Since a window may span more than one word, such constraints in effect involve multiple words.

For example, Braich et al. [4] design their word set so that words have length 15 and are over a three letter alphabet (no G's). The word set has no repeated runs of length more than 4. In addition, there is a mismatch distance of at least 4 between the complement of each word and every 15-base window of the sequence library, and between every pair of 15-base windows. Finally, the subword distance between any pair of 8-base windows and between the complement of a word and any 8-base window is at most 7. Similarly, Faulhammer et al. [19] constrain windows in their RNA strand pool using the mismatch distance and subword distance constraints, as well as using a three-letter alphabet (again G's are excluded).

A related constraint on triples of words is used by Brenner and Lerner [7], in an application where DNA tags and the polymers to be tagged are chemically synthesized in an alternating parallel fashion. Thus each code word (or tag) is the concatenation of "units," one per monomeric chemical unit in the polymer. The units are designed to have the *comma free* constraint: no unit x occurs as a substring in the concatenation of two other distinct units yz.

2.3 Statistical formulation of the word design problem

So far, we have described the word design problem in terms of threshold-based constraints: words (or windows) are designed so that certain measures of each pair of words (or fixed sized set of words) exceed given thresholds. An alternative formulation of the word design problem by Garzon et al. [24] and Rose et al. [46, 47], motivated by principles of statistical mechanics, assigns a weight Z to each possible "hybridization configuration" for all word pairs (w, x). Hybridization configurations include perfectly matched duplexes and also mismatch configurations: partially mismatched duplexes formed from word pairs or from a word and the complement of another word. Associated with each hybridization configuration j is its free energy ΔG_j (see section 2.2.2 above). The statistical weight of the configuration is $\exp(-\Delta G_j/RT)$, where R is the molar gas constant and T is temperature. Let Z_e be the sum of these statistical weights over all error configurations, and let Z_c be the sum of the Z's over all configurations. In the statistical formulation of the word design problem, the task is to find a set of words for which Z_e/Z_c is small.

2.4 Results and further areas for research

The vast literature on design of codes is very relevant to DNA word design. However, there are many new questions and challenges posed by DNA word design that require further study.

The classical theory of codes provides constructions and bounds on sets of code words satisfying the mismatch distance (or Hamming distance) constraint (see the text by MacWilliams and Sloane [38] for a comprehensive treatment or [17] for a short survey). Some tables listing the best known sizes of code sets over the binary alphabet can be found, for example, in the work of Brouwer et al. [9], although even for relatively small values of the code word length and the distance threshold, the size of maximum code word sets are still unknown [36, 33]. Less is known about optimal sizes of code word sets over a 4-letter alphabet. DNA code words with fixed GC content are a natural analogue of constant weight binary codes [9, 34], in which the fraction of 1's in each code word is fixed.

The theory of comma free codes and DeBruijn sequences [14] provides results on construction of DNA word sets that, when concatenated in any order, have the property that the complement of any word does not perfectly match any window of the same length in the concatenated strand [6, 54]. To our knowledge, however, there is no theory for construction of code words that satisfy mismatch constraints adjusted for shifts when the number of mismatches is greater than 1.

Results from coding theory have been applied to obtain several bounds and constructions for

DNA word sets with the property that the mismatch distance between each word and the complement of each distinct word is at least some threshold, d [11, 13, 39]. Marathe et al. [39] also provide constructions of DNA word sets for which, in addition, the mismatch distance between each pair of words is above some threshold. They also describe dynamic programming algorithms that can calculate the total number of words of length n whose free energy value (as approximated by a formula of Breslauer et al.) falls in a given range, and output a random such word.

Ben-Dor et al. [3] consider the problem of designing a DNA word set based on constraints derived from the 2-4 rule for estimating melting temperature. They assign a weight to a strand, namely the estimate of melting temperature of the duplex formed from the strand and its complement given by the 2-4 rule (see section 2.1). They define a set of DNA words to be a $c-h$ code if (i) the weight of every word in the set is at least h and (ii) any strand that has weight at least c occurs at most once as a substrand in any word in the set. They provide strong upper and lower bounds on the optimal size of such DNA word sets.

In light of the difficulty in finding theoretical constructions of optimal codes, stochastic search methods have also been used for construction of good binary codes [16, 29]. Many experimental groups in the area of DNA computing have developed their own programs for designing word sets, each employing somewhat different design criteria. Deaton et al. [11, 13] describe genetic algorithms for finding DNA codes that satisfy several constraints, including mismatch constraints adjusted for shifts. Heuristic methods for finding good DNA encodings have also been described by Garzon et al. [24] (EDNA simulator), Zhang and Shin [64], and Hartemink et al. [27] (SCAN simulator).

Some DNA word design programs are publically available. The DNASequenceGenerator program [44, 15] designs DNA sequences that satisfy mismatch distance constraints and, in addition, have melting temperature or GC content within prescribed ranges. The program can generate DNA sequences de novo, or integrate partially specified words or existing words into the set. The PERMUTE program was used to design the sequences of Faulhammer et al. [10] for their RNA-based 10-variable computation.

Since these methods design DNA words with different constraint sets, there has been no extensive comparison of results from different methods, and thus poor understanding of which stochastic search principles are most effective in finding optimal or close-to-optimal DNA word sets.

Finally, we note that there is also a need for theoretical work aimed at understanding the effectiveness of "local" constraints, such as mismatch constraints on words or on short windows, in ensuring that long strands, formed by concatenating words, meet "global" constraints such as having no secondary structure.

3 Designing strands with structure for DNA computation

Properties of the secondary structure of DNA strands have been exploited for performing DNA computations in several ways. For example, Winfree et al. [58] proposed a method for self-assembly of DNA molecules in a programmable fashion. Sakamoto et al. [49] have examined computational models, based on hairpin loop formation, in which both an input to the computation and state transition information are encoded in a DNA strand. Yurke et al. [62] constructed a simple DNA machine from three strands that acts as a molecular tweezers, fueled by auxiliary DNA strands.

All of these uses of DNA molecules rely on strands that have structural properties that extend well beyond word stability and noninteraction described in the last section. In this section, we first describe strand design requirements for self-assembly computation. We then briefly review algorithms for prediction of secondary structure, and the models for secondary structure underlying these algorithms. Finally, we describe inverse secondary structure prediction problems which arise in design of strands for self-assembly and other forms of DNA computation that exploit the structure of strands.

3.1 Self-assembly computation

Winfree et al. [58, 60] describe a theoretical model for self-assembly of DNA molecules into lattices, so as to simulate the assembly of Wang tiles. Wang tiles are rectangular tiles with labeled edges; simple constraints are placed on the ways that tiles can be aligned, based on the edge labels. It is known that Wang tiles can be designed so that they assemble precisely in a manner that simulates the operation of a Turing machine on a given input.

In experimental work, Winfree et al. [59, 60] designed multi-stranded molecules of DNA that mimic Wang tiles, in that they self-assemble in a pre-programmed fashion (see figure 1). The molecules are shaped in such a way that four single-stranded sequences, called sticky ends, correspond to sides of a Wang tile. A sticky end can be thought of as a colouring that matches only with its complement. Hence, if one side of a tile molecule is figuratively the "blue" side, it bonds only with a "complement blue", and not with any other non-matching sides on other tiles in the experiment. The design criteria are therefore twofold: first, tiles must consistently form tile molecules of a certain structure, and secondly, tiles must be able to find one another and bond to construct a lattice of correctly matched Wang tiles.

The obstacles to overcome in the creation of these tile molecules were ones seen in vivo. "Wild" RNA secondary structures with complex geometry often shift through a series of related but non-identical folds, which would make structure formation vulnerable if subcomponents came together at inopportune times. In tile formation, a number of separate DNA strands must bond in an intricate cross-over fashion, (adding rigidity to the molecule) and the design of the strands must ensure that no unwanted interaction (bonding) occurs within or between strands.

A further consideration is the angular stability of the structural components in the formed tile molecules. Multi-stranded DNA molecules often contain branch formations [51, 52]. A branch formation, also known as a junction or a multi-loop, is a structure formed when three or more helices meet, such as the four-armed multi-loop of figures 4 and 5 in the Appendix. The angles at such junctions may not be rigid, and in any case are constrained by the twist (like the cord from a phone base to the receiver) in a double helix. Regular B-form DNA double helices have 10 base pairs per spiral twist. Other forms of DNA do exist in vivo: Z-form DNA molecules have 12 base pairs per spiral twist, due to the high incidence of stable C-G base pairs, which causes the angle of the bonding parts of the molecule to the external parts of the molecule to be comparatively small.

In response to these considerations, Winfree et al. [59] constructed tile molecules which, roughly speaking, consist of two side-by-side double helix sequences that are linked (cross over) at two junctions. One example is the DAO (double crossover, anti-parallel, odd spaced) molecule of figure 1, formed from four strands of DNA. These molecules have an odd number (3) of helix half-turns in the 16 base pairs between the two crossover points. The DAO molecule's rigidity helps ensure that the sticky ends do not become involved in pair bonding within the molecule or that two sticky ends of one molecule do not bond with two sticky ends from a second molecule.

A precise formulation of the strand design problem for tiles would require a sound model of secondary structure formation in DNA molecules. Secondary structure formation depends strongly on thermodynamic interactions between bonding pairs: intuitively, base pairs bond so as to create the lowest possible energy state. It is desired to find strands for which the free energy difference between the desired secondary structure and all other secondary structures is maximized. We note that other factors such as hydrostatic forces (the interactions of the molecules with the surrounding fluid forces), geometric forces (whether the molecule is B or Z or some other form), and base solution properties (including the molar strength, acidity, and temperature of the solution) also influence the shape of the molecules. However, prediction of the secondary structure of RNA molecules is a useful step in understanding the molecule's geometry.

To date, energy-based models of secondary structure formation pertain to single DNA molecules. We describe two prediction algorithms based on free energy models in the next section. In section 3.3 we return to the inverse secondary structure prediction problem which is relevant to strand design.

3.2 Secondary structure formation with a single DNA strand

A molecule's secondary structure is a list of the bonding base pairs that hold together the tertiary structure. In what follows, if $5'$-$b_1 b_2 \dots b_n$-$3'$ is a DNA or RNA molecule, we represent the secondary structure as a list of pairs $(i, j), 1 \leq i \neq j \leq n$. The pattern of pair bonds can be quite complex in vivo, but simple models of secondary structure limit patterns of bond formation as follows. Given the base pairs (i, j) and (i', j'), bonds must occur in one of two ways:

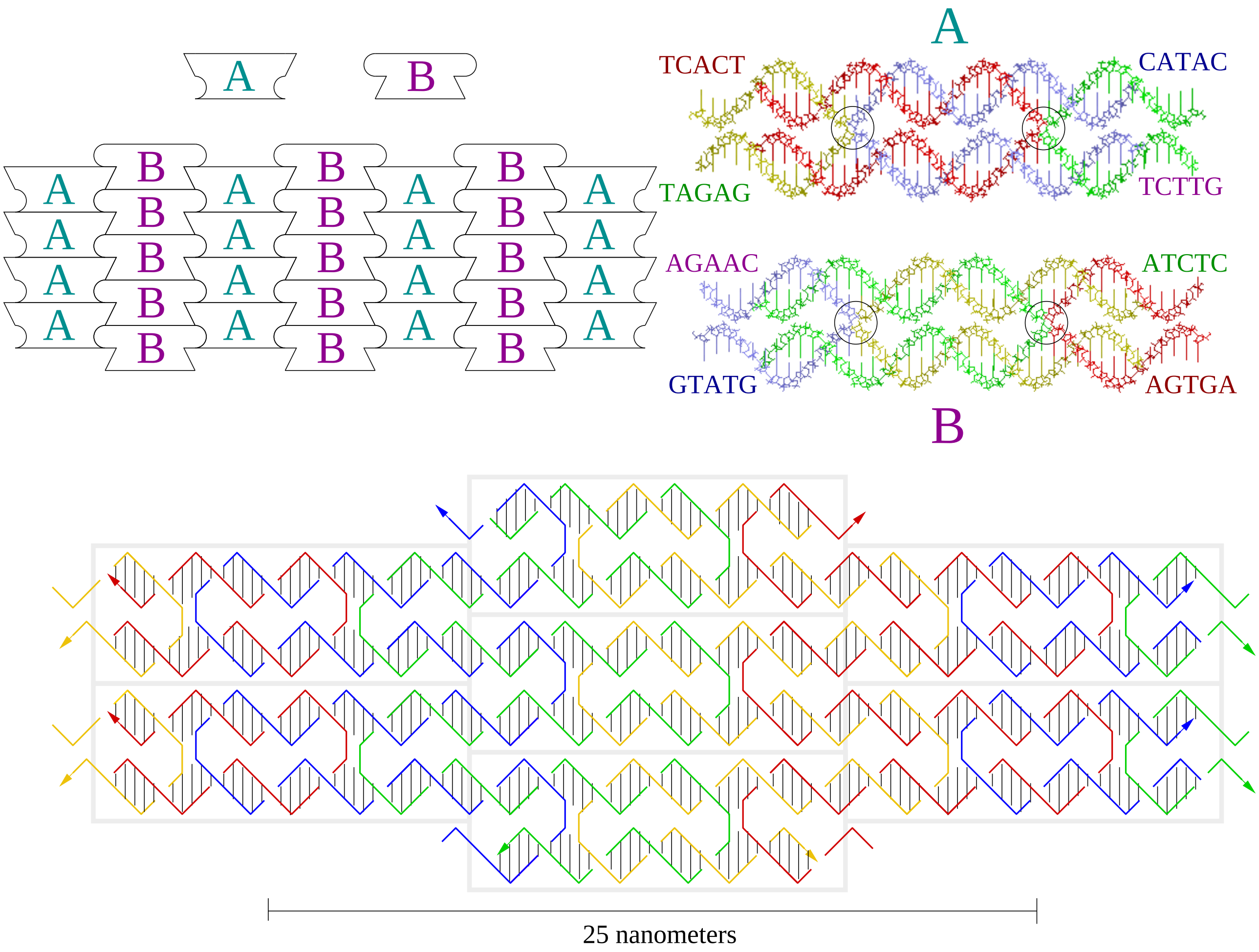

Figure 1: Tile molecules used in the work of [60, 59]. A striped two-dimensional lattice constructed from two types of tiles is shown at the top left. DAO (double, antiparallel, odd spacing) molecules representing each tile are shown at the top right. Cross-over points are circled and the base sequences of sticky ends are given. The lattice topology produced from correct assembly of the tiles is shown at the bottom. Printed with the kind permission of E. Winfree.

- **Inclusive bonding:** $i < i' < j' < j$, suggesting that (i, j) includes (i', j').
- **Precedent bonding:** $i < j < i' < j'$, suggesting that (i, j) precedes (i', j').

In many cases in vivo, there exist bonds that violate these rules, so that $i' < i < j' < j$. An example of this formation might be if two loops in a simple secondary structure, which obey the simplified rules, contain non-bonding bases that then are able to form bonds with one another. These sorts of bonds are referred to as pseudoknots.

In a pseudoknot-free secondary structure, the paired bases partition the molecule into loops. There are a number of types of loops that a secondary structure may form. The simplest loop with unpaired bases, which shows up in any non-empty secondary structure made from one strand, is the hairpin loop. This loop is created when the strand makes a "U-turn" to fold back onto itself. On the other end of the scale, the most complex of the loops is the multi-loop, which is created when a number of helix arms come together at a junction. Recall the specialized form of this loop, the branched junction, which usually describes a cruciform-shaped multi-loop with no unpaired bases. More generally, multi-loops may contain unpaired bases that separate helix arms. For uniformity, stacked pairs, namely two pairs (i, j) and $(i+1, j-1)$ (that may form part of a longer helix in the structure) are also considered to be a loop.

Associated with a secondary structure of a strand is its free energy, which is a measure of the stability of the structure. Extensive experimental observation by Freier et. al. [18] has lead to a model for predicting the energy of pseudoknot-free secondary structures. The energy of a secondary structure is the sum of the contributions of its loops. As in the model for the free energy of a perfectly matched duplex described in section 2.1, stacked pairs (which are analogous to nearest neighbour pairs) contribute a negative energy, and thus are considered to stabilize the structure. Other loops in a structure are by and large destabilizing, that is, contribute positive energy.

Zuker et al. [65] have used this energy model to predict the minimum energy secondary structure of an RNA strand using a dynamic programming method. The energy minimization algorithm defines two different functions, W and V, where $W(i, j)$ is the minimum folding energy, taken over all possible foldings between the bases in positions i and j. W satisfies the following identity [63] (base cases excluded):

$$W(i,j) = \min[W(i+1,j), W(i,j-1), V(i,j), \min_{k:i \le k \le j}\{W(i,k) + W(k+1,j)\}].$$

$V(i, j)$ is the minimum folding energy taken over all possible foldings between the bases in positions i and j in which (i, j) is paired. $V(i, j)$ also satisfies an inductive identity that has terms for each type of loop (omitted here). Based on these and other identities, Zuker's algorithm predicts the optimal secondary structure of a strand of length n in $O(n^3)$ time. An online implementation of the algorithm is available [66].

McCaskill [40] proposed another approach to RNA or DNA secondary structure, which we call the partition function algorithm. In vivo, some RNA molecules do not stabilize to one single structure, but rather cascade through a series of related low energy structures. A model based on the so-called partition function for RNA secondary structures (analogous to the partition function from statistical mechanics) associates a probability with each possible (pseudoknot-free) secondary structure of an RNA molecule. Thus, associated with each possible base pairing of the molecule is a weight, defined to be the sum of the probabilities of the structures in which it occurs. The partition function algorithm uses a clever dynamic programming approach to calculate the weight of each possible base pair. The output of the algorithm is not a single definitive structure, but rather the set of base pair weights, which gives the user an idea of the structures likely to be formed by the input sequence, An implementation of the partition function algorithm by Hofacker et al. [30] is available in a suite of programs known as the Vienna Package [31].

We note that the pseudoknot structure has been shown to be quite important in a variety of cellular functions, and algorithms that can predict pseudoknot secondary structures would be useful. At this time, there are not general, experimentally derived energy models for pseudoknot energies. This makes the development of prediction algorithms difficult, although some work has been done on theoretical models. Lyngsøand Pedersen [37] and Rivas and Eddy [45] have each developed models of algorithms to predict certain classes of pseudoknots. However, initial investigation into a most general model for structure prediction, with a highly abstracted energy function, suggest that the problem of structure prediction with pseudoknots may be NP-complete [37].

3.3 Inverse secondary structure prediction

The simplest form of the inverse secondary structure prediction problem, which is already very relevant to strand design, is as follows: given a secondary structure S (that is, a list of pase pair indices) for an RNA (or DNA) strand of length n, find a strand whose minimum free energy structure is S (according to the loop-based free energy model of the type described in section 3.2).

It is an open question whether a polynomial time algorithm exists for inverse secondary structure prediction. Two heuristic algorithms have been implemented by Hofacker et al. [30]. Both are stochastic local search methods, starting from a randomly generated sequence of the requested length or from a sequence input by the user. One algorithm, which we call the inverse-MFE algorithm, calls a free energy based structure prediction algorithm similar to that of Zuker [65] at each step in the search, and modifies the current sequence based on differences between the structure output by the Zuker algorithm and the desired structure S. The second algorithm, which we call the inverse-partition-function algorithm, modifies the current sequence based on the base pair probabilities output by the partition function algorithm of McCaskill [40]. Our own experimentation with these two algorithms suggests that the inverse-partition-function algorithm may have a greater likelihood of finding a sequence that will fold into our desired structure when given a reasonable starting sequence (See the Appendix for example trials of the MFE and Partition

Function.) With a running time of $O(n^6)$, both algorithms become costly to run as the length of the sequence grows large.

A generalization of the inverse structure prediction problem arises when the desired structure is composed of more than one strand. Researchers have been using a sequence symmetry based approach, which in the terminology of section 2 is a subword distance maximisation approach. Seeman [51] has designed a user-directed strand design algorithm for DNA strands, and describes use of the algorithm for design of branched junctions and other unstable structures. Roughly, a semi-automatic sequence assignment then takes place by alternating the following steps. The user suggests an assignment of bases to a short, fixed-length subsequence – called a chunk – of the sequences to be designed. The algorithm then checks that the sequences of base pairs assigned so far, including the new chunk, satisfies the subword distance constraint and if not, gives the user the option of reassigning the chunk. Sequence symmetry minimization methods have also been successfully used by LaBean et al. [35], Winfree et al. [59], and Sakamoto et al. [49] to create sequences that fold into desired structures. In the strands comprising the DAO molecules of figure 1, there are no 6-base subsequences (subwords) complementary to other 6-base subsequences except as required by the design.

3.4 Further areas for research

Development of efficient algorithms for secondary structure prediction in the presence of pseudoknots is currently an active research area [37, 45]. Extension of energy-based secondary structure models and secondary structure prediction algorithms to multiple strands would also be useful in formulating the strand design problem for multiple strands.

To date, approaches to inverse secondary structure prediction, even for single-stranded structures, are heuristic in nature. These methods, particularly the inverse-partition-function algorithm described in section 3.2, work well in practice, suggesting that it may be possible to efficiently solve the inverse structure prediction algorithm for "reasonable" structures with relatively long stabilizing helices. In particular, it would be interesting to know on which structures the sequence symmetry based methods are successful.

Generalizations of the simplistic inverse structure design problem described at the start of section 3.3 more faithfully capture the strand design problem. For example, it is desired to have algorithms for inverse structure design that produce a strand (or strands) with *maximum* free energy difference between the desired conformation and undesired conformations.

References

[1] L.M. Adleman, "Molecular computation of solutions to combinatorial problems," Science, Vol 266, Issue 11, November 1994, pages 1021-1024.

[2] E. B. Baum, "DNA sequences useful for computation," Proc. DNA Based Computers II, DIMACS Workshop June 10-12, 1996, L. F. Landweber and E. B. Baum, Editors, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 44, 1999, pages 235-242.

[3] A. Ben-Dor, R. Karp, B. Schwikowski, and Z. Yakhini, "Universal DNA tag systems: a combinatorial design scheme," Proc. RECOMB 2000, ACM, pages 65-75.

[4] R.S. Braich, C. Johnson, P.W.K. Rothemund, D. Hwang, N. Chelyapov, and L.M. Adleman, "Solution of a satisfiability problem on a gel-based DNA computer," Preliminary Proc. Sixth International Meeting on DNA Based Computers, Leiden, The Netherlands, June, 2000.

[5] S. Brenner, M. Johnson, J. Bridgham, G. Golda, D.H. Lloyd, D. Johnson, S. Luo, S. McCurdy, M. Foy, M, Ewan, R. Roth, D. George, S. Eletr, G. Albrecht, E. Vermaas, S.R. Williams, K. Moon, T. Burcham, M. Pallas, R.B. DuBridge, J. Kirchner, K. Fearon, J. Mao, and K. Corcoran, "Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays," Nature Biotechnology, Vol. 18, June 2000, pages 630-634.

[6] S. Brenner, "Methods for sorting polynucleotides using oligonucleotide tags," U.S. Patent Number 5,604,097, Feb 18, 1997.

[7] S. Brenner and R. A. Lerner, "Encoded combinatorial chemistry," Proc. Natl. Acad. Sci. USA, Vol 89, pages 5381-5383, June 1992.

[8] K. Breslauer,R. Frank, H. Blocker, L. Marky, "Predicting DNA duplex stability from the base sequence", Proc. Natl. Acad. Sci. USA 83 (1986), pages 3746-3750.

[9] A.E. Brouwer, J.B. Shearer, N.J.A. Sloane, W.D. Smith, "A new table of constant weight codes", IEEE Transactions on Information Theory, Vol. 36, No. 6, November 1990, pages 1334-1380.

[10] A. R. Cukras, D. Faulhammer, R. J. Lipton, and L. F. Landweber, "Chess games: a model for RNA based computation," Preliminary Proc. Fourth International DIMACS Meeting on DNA Based Computers, U. Pennsylvania, 1998, pages 27-37.

[11] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr., "Good encodings for DNA-based solutions to combinatorial problems," Proc. DNA Based Computers II, DIMACS Workshop June 10-12, 1996, L. F. Landweber and E. B. Baum, Editors, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 44, 1999, pages 247-258.

[12] M. Garzon, R. Deaton, P. Neathery, D. R. Franceschetti, and S. E. Stevens, Jr., "On the encoding problem for DNA computing," Preliminary Proc. 3rd DIMACS Workshop on DNA Based Computers, June 23-25, 1997, pages 230- 237.

[13] R. Deaton, M. Garzon, R. C. Murphy, J. A. Rose, D. R. Franceschetti, and S. E. Stevens, Jr., "Genetic search of reliable encodings for DNA-based computation," Koza, John R., Goldberg, David E., Fogel, David B., and Riolo, Rick L. (editors), Proceedings of the First Annual Conference on Genetic Programming 1996.

[14] W. L. Eastman, "On the construction of comma-free codes," IEEE Transactions on Information Theory, Vol. 11, 1965, pages 263–267.

[15] U. Feldkamp, W. Banzhaf, H. Rauhe, "A DNA sequence compiler," Poster presented at the 6th International Meeting on DNA Based Computers, Leiden, June, 2000. See also http://ls11-www.cs.uni-dortmund.de/molcomp/Publications/publications.html (visited November 11, 2000).

[16] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," IEEE Transactions on Information Theory, Vol. IT-33, No. 1, January 1987.

[17] T. Ericson, "Bounds on the size of a code", Topics in Coding Theory, Springer-Verlag, 1989.

[18] Freier, S.M., Kierzek, R., Jaeger, J.A., Sugimoto, N., Caruthers, M.H., Neilson, T., and Turner, D.H., "Improved free-energy parameters for predictions of RNA duplex stability," Proc. Natl. Acad. Sci. USA, 83, 1986, pages 9373-9377.

[19] Faulhammer, D., Cukras, A. R., Lipton, R.J., and L. F. Landweber, "Molecular computation: RNA solutions to chess problems," Proc. Natl. Acad. Sci. USA, 97: 1385-1389.

[20] The PERMUTE program. Available at the PNAS web site http://www.pnas.org/cgi/content/full/97/4/1385/DC1. Visited November 22, 2000.

[21] A. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Smith, and R. M. Corn, "Demonstration of a word design strategy for DNA computing on surfaces," Nucleic Acids Research, Vol. 25, No. 23, December 1997, pages 4748-4757.

[22] M. Garzon, R. Deaton, P. Neathery, D.R. Franceschetti, and R.C. Murphy, "A new metric for DNA computing," Proc. 2nd Genetic Programming Conference, Morgan Kaufman, 1997, pages 472-478.

[23] M. Garzon, R. Deaton, L.F. Nino, S.E. Stevens Jr., and M. Wittner, "Encoding genomes for DNA computing," Proc. 3rd Genetic Programming Conference, Madison, WI, 1998.

[24] M. Garzon, R. J. Deaton, J. A. Rose, and D. R. Franceschetti, "Soft molecular computing," Preliminary Proc. Fifth International Meeting on DNA Based Computers, June 14-15, MIT, 1999, pages 89-98.

[25] S. W. Golomb, B. Gordon, and L. R. Welch, "Comma-free codes," Can. J. Math., Vol 10, 1958, pages 202–209.

[26] A.J. Hartemink and D. K. Gifford, "Thermodynamic simulation of deoxyoligonucleotide hybridization," Prel. Proc. 3rd DIMACS Workshop on DNA Based Computers, June 23-27, 1997, U. Pennsylvania, pages 15-25.

[27] A.J. Hartemink, D.K. Gifford, and J. Khodor, "Automated constraint-based nucleotide sequence selection for DNA computation," 4th Annual DIMACS Workshop on DNA-Based Computers, Philadelphia, Pennsylvania, June 1998

[28] I. Hirao, Y. Nishimura, Y. Tagawa, K. Watanabe, and K. Miura, "Extraordinarily stable mini-hairpins: electrophoretical and thermal properties of the various sequence variants of d(GCGAAAGC) and their effect on DNA sequencing," Nucleic Acids Research, Vol. 20, No. 15, 3891-3896, 1992.

[29] I.S. Honkala, and P.R.J. Ostergard, "Code design," In Local Search In Combinatorial Optimization (E. Aarts and J.K. Lenstra, eds.), Wiley-Interscience Series in Discrete Mathematics and Optimization, 1997.

[30] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoffer, M. Tacker, and P. Schuster, "Fast folding and comparison of RNA secondary structures." Monatshefte f. Chemie, Volume 125, 1994. pages 167-188.

[31] I. Hofacker, "RNA folding packages". 3 October, 2000. http://www.tbi.univie.ac.at/~ivo/RNA (3 October 2000).

[32] K. Komiya, K. Sakamoto, H. Gouzu, S. Yokoyama, M. Arita, A. Nishikawa, and M. Hagiya, "Successive state transitions with I/O interface by molecules," Preliminary Proc. 6th Intl. Meeting on DNA Based Computers, June, 2000, pages 21-30.

[33] Y. Klein, S. Litsyn, A. Vardy, "Two new bounds on the size of binary codes with a minimum distance of three," Designs, Codes and Cryptography, 6, 1995, pages 219-227.

[34] Klaus-Uwe Koschnick, "Some new constant weight codes", IEEE Transactions on Information Theory, Vol. 37, No. 2, November 1991, pages 370-371.

[35] C. Mao, T.H. LaBean, J.H. Reif and N.C. Seeman, "Logical computation using algorithmic self-assembly of DNA triple crossover molecules," Nature, Vol 407, September 2000, pages 493-496.

[36] S. Litsyn, A. Vardy, "The uniqueness of the Best code," IEEE Transactions on Information Theory, Vol. 40, No. 5, November 1994, pages 1693-1698.

[37] R. Lyngsøand C. Pedersen, "Pseudoknots in RNA secondary structures," Proc. Fourth Annual International Conference on Computational Molecular Biology (RECOMB00), 2000, pages 201 - 209.

[38] F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error-Correcting Codes," North-Holland, 1977.

[39] A. Marathe, A. Condon, and R. Corn, "On combinatorial DNA word design," Proc. 5th International Meeting on DNA Based Computers, June 1999. To appear in J. Computational Biology.

[40] J.S. McCaskill, "The equilibrium partition function and base pair binding probabilities for RNA secondary structure," Biopolymers, Vol 29, 1990, pages 1105-1119.

[41] K. U. Mir, "A restricted genetic alphabet for DNA computing," Proc. 2nd Annual Workshop on DNA Based Computers, June 10-12, 1996, L. F. Landweber and E. B. Baum, Editors, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 44, 1999, pages 243-246.

[42] Q. Ooyang, P. Kaplan, S. Liu, and A. Libchaber, "DNA solution of the maximal clique problem," Science, 278, 17 October 1997, pages 446-449.

[43] N. Peyret, P.A. Seneviratne, H.T. Allawi, and J. SantaLucia, "Nearest-neighbor thermodynamics and NMR of DNA sequences with internal A center dot A, C center dot C, G center dot G, and T center dot T mismatches," Biochemistry 38: (12) 3468-3477 MAR 23 1999.

[44] Programmable DNA web site, http://ls11-www.cs.uni-dortmund.de/molcomp/Downloads/downloads.html. Visited November 11, 2000.

[45] E. Rivas and S. Eddy, "A dynamic programming algorithm for RNA structure prediction including pseudoknots," Journal of Molecular Biology, vol 285. 1999. pages 2053-2068.

[46] J. A. Rose, R. Deaton, D. R. Franceschetti, M. Garzon, and S. E. Stevens, Jr., "A statistical mechanical treatment of error in the annealing biostep of DNA computation," Special program in DNA and Molecular Computing at the Genetic and Evolutionary Computation Conference (GECCO-99), Orlando, FL., July 13-17, 1999, Morgan Kaufmann.

[47] J.A. Rose, and R.J. Deaton, "The fidelity of annealing-ligation: a theoretical analysis," Preliminary Proc. 6th Intl. Meeting on DNA Based Computers, June, 2000, pages 207-221.

[48] S. Roweis, E. Winfree, R. Burgoyne, N. V. Chelyapov, M. F. Goodman, P.W.K. Rothemund, and L. M. Adleman, "A sticker-based model for DNA computation," Proc. DNA Based Computers II, DIMACS Workshop June 10-12, 1996, L. F. Landweber and E. B. Baum, Editors, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 44, 1999, pages 1-29.

[49] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, and M. Hagiya, "Molecular computation by DNA hairpin formation," Science, Vol. 288, May 2000, pages 1223-1226.

[50] J. SantaLucia, "A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics," Proc. Natl Acad Sci USA 95: (4) 1460-1465 February 17, 1998.

[51] N.C. Seeman, "De novo design of sequences for nucleic acid structural engineering," Journal of Biomolecular Structure and Dynamics, Vol. 8, Issue 3, 1990, pages 573-581.

[52] N. C. Seeman, H. Wang, B. Liu, J. Qi, X. Li, X. Yang, F. Liu, W. Sun, Z. Shen, R. Sha, C. Mao, Y. Wang, S. Zhang, and J. Chen, "The perils of polynucleotides: the experimental gap between the design and assembly of unusual DNA structures," Proceedings of the Second Annual Workshop on DNA Based Computers, June 10-12, 1996, L. F. Landweber and E. B. Baum, Editors, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 44, 1999, pages 215-234.

[53] D. D. Shoemaker, D. A. Lashkari, D. Morris, M. Mittman, and R. W. Davis, "Quantitative phenotypic analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy," Nature Genetics, Vol 16, December 1996, pages 450–456.

[54] W. D. Smith and A. Schweitzer, "DNA computers in vitro and in vivo," NECI Technical Report, March 20, 1995.

[55] T. Strachen and A.P. Read, "Human Molecular Genetics," Bios Scientific Publishers, 1996.

[56] http://wwwchem.leidenuniv.nl/lic98/98high.htm Visited: October 20, 2000.

[57] J. G. Wetmur, "DNA probes: applications of the principles of nucleic acid hybridization," Critical Reviews in Biochemistry and Molecular Biology 26(3/4):227-259, 1991, pages 227-259.

[58] E. Winfree, X. Yang, and N. Seeman, "Universal computation via self-assembly of DNA: some theory and experiments," Proceedings of the Second Annual Workshop on DNA Based Computers, June 10-12, 1996, L. F. Landweber and E. B. Baum, Editors, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 44, 1999, pages 191-214.

[59] E. Winfree, F. Liu, L. Wenzler, and N. Seeman, "Design and self-assembly of 2D DNA crystals," Nature, Vol 394, August 1998, pages 539-544.

[60] E. Winfree, "Algorithmic self-assembly of DNA", Ph.D. thesis at California Institute of Technology, August 1998, 110 pages.

[61] H. Yoshida and A. Suyama, "Solution to 3-SAT by breadth first search," Proc. 5th Intl. Meeting on DNA Based Computers, M.I.T. 1999, pages 9-20.

[62] B. Yurke, A.J. Turberfield, A.P. Mills Jr, F.C. Simmel, and J.L. Newmann, "A DNA-fuelled molecular machine made of DNA," Nature, Vol 406, August 10 2000, pages 605-608.

[63] M. Zuker, "RNA folding form" 18 August 1998. http://www.ibc.wustl.edu/˜zuker/Bio-5495/RNAfold-html/. (29 September 2000)

[64] B-T. Zhang and S-Y. Shin, "Molecular algorithms for efficient and reliable DNA computing," Proc. 3rd Annual Genetic Programming Conference, Edited by J. R. Koza, K. Deb, M. Doringo, D.B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, Morgan Kaufmann, 1998, pages 735-742.

[65] M. Zuker and P. Stielger, "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information," Nucleic Acids Res 9,(1981), pages 133-148.

[66] M. Zuker, "Algorithms, thermodynamics, and databases for RNA secondary structure." 6 September 2000. http://www.ibc.wustl.edu/z̃uker/rna/. (September 29, 2000.)

Appendix

Here we report on runs of the inverse-MFE and inverse-partition-function algorithms for inverse RNA secondary structure prediction (see section 3.3). These heuristic search algorithms are available through the Vienna Package [31]. Both algorithms take as input a description of a pseudoknot-free secondary structure. Ideally, the algorithms output a strand which is predicted (by a variant of Zuker's dynamic programming algorithm) to fold into the given input structure. If no such strand is found, the algorithms output the "best" strand found during the search.

The input secondary structure is represented as a string over the alphabet $\{(,),.\}$, such as ((..(((...)))..)). In this representation, matching parentheses represent base pairs in the secondary structure and .'s represent unpaired bases, with the left end of the string corresponding to the 5' end of the strand. Thus, the example string above represents the secondary structure

$$\{(1,17),(2,16),(5,13),(6,12),(7,11)\}.$$

Figures 2 and 3 show the outputs of the inverse-mfe and inverse-partition-function algorithms, respectively, on the input $S =$ ((..(((...((...))...))))). The inverse-mfe algorithm failed to find a correct output strand; the strand found, namely *UUCUAGUUACGCGGUGCAAUGCUAA*, is predicted to fold to the structure ...((((...((...))...)))), with a (minimum) free energy of -2kcal/mol. The free energy of the output strand, when folded into the input structure S, is 3.1kcal/mol. In contrast, the inverse-partition-function algorithm finds a correct output, namely *GGUUACCUCG-GCAUUGCAUUGGUCC* which has a minimum free energy of -3kcal/mol when folded according to S, in a reported time of 1.24 seconds.

Our second pair of example runs from the two programs is on the input

$$S' = ((((..(((....))).(((....))).(((....)))..)))).$$

Figures 4 and 5 show the outputs of the inverse-mfe and inverse-partition-function algorithms, respectively, on the input S'. Again, the inverse-mfe algorithm failed to find a correct output strand but the inverse-partition-function succeeds.

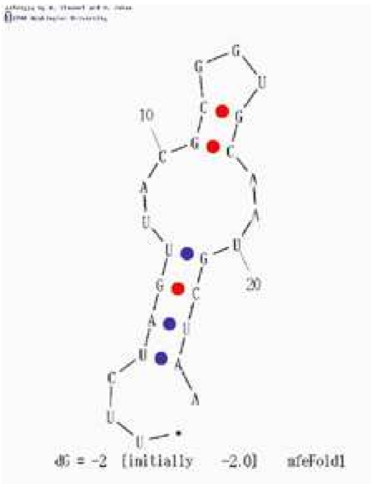

Figure 2: Secondary structure of the output of the inverse-MFE algorithm on input ((..(((...((...))...))))). Dots represent bonds between base pairs. The output strand does not form the base pairs at the ends specified in the input structure. The call to the program in this case was "RNAinverse -Fmp -f 0.5" and the search started from the string *UAUUAAUUACUCGGUAAAAU-GUUAA*.

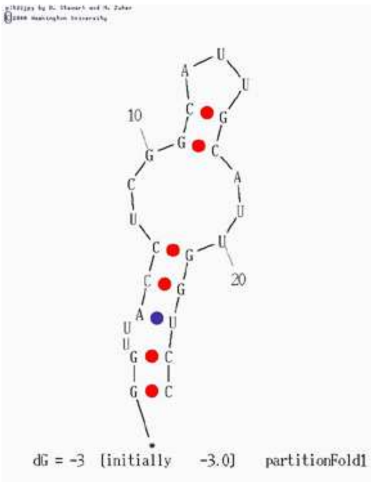

Figure 3: Secondary structure of the output of the inverse-partition-function algorithm on input ((..(((...((...))...))))). The output strand is predicted to fold in accordance with the input structure.

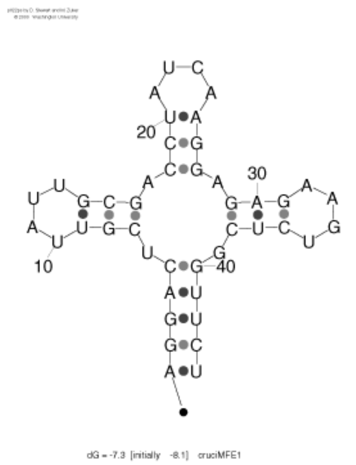

Figure 4: Secondary structure of the output of the inverse-MFE algorithm on input ((((..(((....))).(((....))).(((....)))..)))). In the southern helix of the secondary structure there are five helix pairs rather than four.

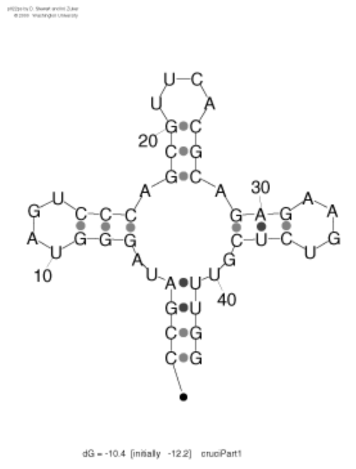

Figure 5: Secondary structure of the output of the inverse-partition-function algorithm on input ((((..(((....))).(((....))).(((....)))..)))). The output strand is predicted to fold in accordance with the input structure. The strand was found in a reported time of 3.3 seconds.