Designing Nucleotide Sequences for Computation: A Survey of Constraints

Jennifer Sager sagerj@cs.unm.edu
Darko Stefanovic darko@cs.unm.edu
Department of Computer Science
University of New Mexico

Abstract

We survey the biochemical constraints useful for the design of DNA code words for DNA computation. We define the DNA/RNA Code Constraint problem and cover biochemistry topics relevant to DNA libraries. We examine which biochemical constraints are best suited for DNA word design.

1 Introduction

Most DNA[1] computation models assume that computation is error-free. For example, Adleman [2] and Lipton [3] used randomly generated DNA strings in their experiments because they assumed that errors due to false positives are rare. However, it has been experimentally shown that randomly generated codes are inadequate for accurate DNA computation as the size of the problem grows [4], since a poorly chosen set of DNA strands can cause errors. Therefore for many types of DNA computers, it may be practical or even necessary to create a 'library' or 'pool' of DNA word codes suitable for computation.[2]

A properly constructed library would help to minimize errors so that DNA computation is more practical, reliable, scalable, and less costly in terms of materials and laboratory time. The construction of a library is non-trivial for two reasons. First, there are 4^N unique DNA strings of length N; thus the number of candidate molecules grows exponentially in the length of the DNA string. Second, the constraints used to find a library are complex since they are subject to the laws of biochemistry as well as the specific algorithm and computation style. Deaton states that it is likely that the construction of a library "is as difficult [i.e., NP-hard or harder] as the combinatorial optimization problems they are intended to solve" [7].

We examine the biomolecular constraints typically used to choose a set of DNA strings suitable for computation. A combination of these constraints are a possible solution to the DNA Code Constraint Problem. Given an algorithm for a type of DNA computer, the DNA Code Constraint Problem is to find a set of constraints that the DNA strands must satisfy to minimize the number of errors due to the choice of DNA strands. The constraints are determined by the physical reality of performing the algorithm in the laboratory and the specific algorithm and computation style.

2 Positive And Negative Design

Even though there are many types of DNA computers, most share similar biochemical requirements because they use the same fundamental biochemical processes for computation. The fundamental computation step for most DNA computers occurs through the bonding (hybridization) and unbonding (denaturation) of oligonucleotides (short strands of DNA). A single strand of DNA is composed of a sequence of nucleotides. Each nucleotide contains a sugar (deoxyribose or ribose), a phosphate group, and one of four bases, adenine (A), thymine (T), guanine (G),

[1] Even though we describe most of the constraints in terms of DNA, RNA computers also exist (for an example see [1]) and all of the constraints are also relevant to RNA.

[2] For an overview of library design see [5]. For a survey of algorithms that have been used to solve the DNA/RNA Code Design Problem see [6].

or cytosine (C). RNA is composed similarly except that thymine is replaced by the closely related uracil (U). The nucleotides only form stable bonds in certain combinations: A hydrogen-bonds to T or U, and G hydrogen-bonds to C. Thus A is the Watson-Crick complement of T/U, and G is the Watson-Crick complement of C. In addition, the "wobble pair", G and U, can form weak bonds. Hybridization or annealing occurs when a sequence of nucleotides bonds to the nucleotides of another sequence, starting from the 5' end (the ribose end) of one sequence and the 3' end (the phosphate end) of the other sequence. For more comprehensive information about DNA chemistry, see [8, 9].

Creating an error-free library typically requires that planned hybridizations and denaturations (between a word and its Watson-Crick complement) do occur and unplanned hybridizations and denaturations (between all other combinations of code words and their complements) do not occur. The former situation is referred to as the *positive design problem* and the latter is referred to as the *negative design problem* [6, 10].

The positive design problem requires that there exists a sequence of reactions that produces the desired outputs starting from the given inputs. Thus, positive design attempts to "optimize affinity for the target structure" [10]. These reactions must occur within a reasonable amount of time for feasible concentrations. Usually the strands must satisfy a specified secondary structure criterion (e.g., the strand must have a desired secondary structure or have no secondary structure at all). Since a strand is typically identified by hybridization with its perfect Watson-Crick complement, the positive design problem requires that each Watson-Crick duplex is stable. In addition, for computation styles that use denaturation, the positive design problem often requires all of the strands in the library to have similar melting temperatures, or melting temperatures above some threshold. In short, positive design tries to maximize hybridization between perfect complements.

The negative design problem requires that (1) no strand has undesired secondary structure such as hairpin loops, (2) no string in the library hybridizes with any string in the library, and (3) no string in the library hybridizes with the complement of any string in the library. Thus negative design attempts to "optimize specificity for the target structure" [10]. Unplanned hybridizations can cause two types of potential errors: false positives and false negatives. False negatives occur when all (except an undetectable amount) of DNA that encodes a solution is hybridized in unproductive mismatches. Since mismatched strands are generally less stable than perfectly matched strands, false negatives can be controlled by adjusting strand concentrations. Deaton experimentally verified the occurrence of false positives, which happen when a mismatched hybridization causes a strand to be incorrectly identified as a solution [4]. False positives can be prevented by ensuring that all unplanned hybridizations are unstable. In short, negative design problem tries to minimize non-specific hybridization.

Positive design often uses GC-content and energy minimization as heuristics (see below). Negative design uses combinatorial methods (such as Hamming distance, reverse complement Hamming distance, shifted Hamming distance, and sequence symmetry minimization), and thermodynamic methods (such as minimum free energy). Constraints which incorporate both positive and negative design are probability, average incorrect nucleotides, energy gap, probability gap, and energy minimization in combination with sequence symmetry minimization. The best-performing models for designing single-strand secondary structure use simultaneous positive and negative design and significantly outperform either method alone; however, kinetic constraints must be considered separately since low free energy does not necessarily imply fast folding [10]. We believe that this same principle holds for designing hybridizations between pairs of strands.

3 Structure

Structure calculations attempt to predict which reactions will occur (i.e., which bonds will form and which will break). The tendency of the atoms in a molecule to bond together is referred to as the molecule's stability. Stability is affected by the sequence of bases, as well as environmental factors such as temperature, pH, the time given to allow reaction to complete, salt concentration, and the concentrations of the chemical components; temperature

is the most significant of these environmental factors. The DNA folding problem refers to the prediction of the structure and folding energy of a given sequence. The inverse of this problem is the selection of a sequence with a given structure.

DNA and RNA can fold back upon itself into loops or other irregular complex twisted shapes. The remaining sections can be a combination of different types of loop structures, which are single-stranded sections bounded by bonded base pairs (stem sections). A strand that has no stems is considered to have no secondary structure. Loops can be classified into several categories, Figure 1. A hairpin loop is a loop with a single stem. Internal loops are loops with single bases on both sides of the stem. Bulging loops are loops with single bases on only one side of the stem. Loops with three or more stems are called branching loops.

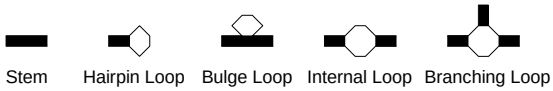

Figure 1: DNA loops. Solid areas represent double stranded sections. Lines represent single stranded sections.

The structure of DNA is categorized in a four-level hierarchy. The primary structure refers to the sequence of bases. The secondary structure describes which individual molecules bond to each other. Tertiary structure refers to the three-dimensional folding—the actual positions of the molecules within a single chain in three-dimensional space. Quaternary structure describes the three-dimensional interaction between two or more chains. The structure of DNA and RNA can be fairly accurately predicted from just the secondary structure because the tertiary interactions are much weaker than the secondary interactions. This assumption is particularly appropriate for random sequences since they have a low probability of having tertiary structures [11]. In contrast, sequences selected by evolution are likely to have tertiary interactions; however, even though the approximation will be less accurate, the structure and folding energy of non-random sequences can still be approximated from just the secondary structure [11]. Unfortunately, there is an exponential number (approximately 1.8^N) of possible secondary structures for a sequence of length N [11, 12].

The stability of a DNA structure is a result of the change in free energy owing to bonding. The simplest explanation of free energy is that "free energy is energy that has the ability to do work" [8]. When a spontaneous reaction occurs (at constant temperature and pressure), there is a decrease in free energy. This decrease in free energy is equal to the maximum amount of work that the system can do on its surroundings. Conversely, for a non-spontaneous reaction, the free energy is the amount of work that must be done to cause the reaction to occur. The change in free energy is denoted ΔG. If $\Delta G < 0$, the reaction is spontaneous in the forward direction. If $\Delta G = 0$, the reaction is at equilibrium. If $\Delta G > 0$, the reaction is spontaneous in the reverse direction. When a bond between atoms forms, stronger bonds produce bigger changes in free energy; consequently, atoms that bond strongly together are more likely to exist in bonded form.

Thus DNA is more stable when it has lower free energy and in most cases it will fold into the structure that has the minimum free energy. However, his structure is not necessarily the most likely structure to form. In fact, the equilibrium structure may not be a single structure at all; "what actually occurs, on the time scale of most enzymatic reactions relevant for biological function, is rather an ensemble of related structures interchanging more or less rapidly with one another" [13]. For example, the structure of the DNA of the bacterial virus T4 has several forms in solution including a tight coil and an extended form [14].

The most widely used method to estimate the free energy of DNA is the *nearest neighbor model*, which predicts the free energy of a duplex as the sum of the free energy of each nearest neighbor pair plus a few correction factors. The model is valid for single strands, Watson-Crick complementary duplexes, and mismatched duplexes, and it can be adjusted for various temperature, pH, and salt conditions. Nearest neighbor parameters have been measured for several different types of nearest neighbors including matched pairs, internal mismatched pairs, dangling ends, internal loops, hairpin loops, and bulge loops. However, the fastest algorithms assume that the structure has no pseudoknots. (A pseudoknot is

an occurrence of two pairs of bonded bases at positions (i,k) and (j,l) such that $i < j < k < l$.) Probabilistic measurements of free energy can also be derived from the nearest neighbor model to predict the most likely structure. Algorithms also exist which predict the energy landscape of the structures that a strand can form [15].

For a summary of nearest-neighbor thermodynamics see [11]. For more information about nucleotide structures see [12, 16]. For more information about structure prediction algorithms see [17].

3.1 Secondary Structure of Single Strands

Most DNA computation styles need strands with no secondary structure (i.e., no tendency to hybridize with itself). There are, on the other hand, cases where specific secondary structures are desired, such as for deoxyribozyme logic gates [18]. Even there, structures different from the desired must be eliminated. Figure 2 shows the desired structure.

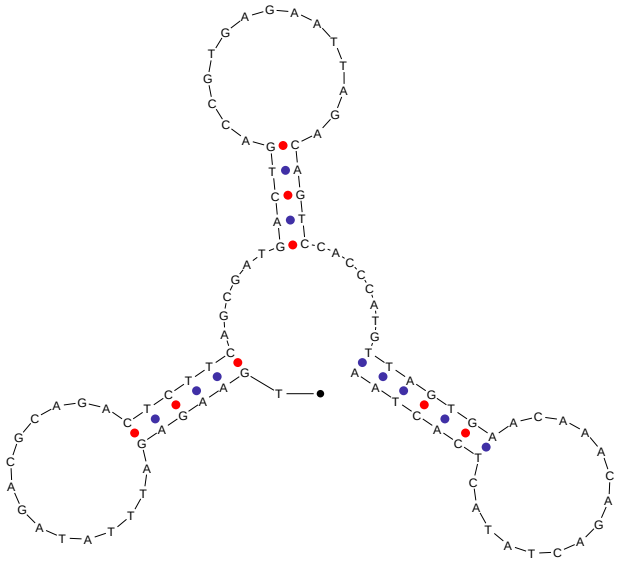

Figure 2: Example of secondary structure in Stojanovic and Stefanovic's DNA automaton [18] as computed by Mfold [19–21] using 140 mM Na^+, 2 mM Mg^{++}, and 25°C. The strand has three hairpin loops, which is the desired secondary structure. ΔG is -12.3 kcal/mol.

There are several heuristics that are used to prevent secondary structure. Sometimes, repeated substrings and complementary substrings within a single strand which are non-overlapping and longer than some minimum length are forbidden to prevent stem formation; this heuristic is often called *sequence symmetry minimization* [10] or *substring uniqueness*. Another heuristic is to forbid particular substrings; these *forbidden substrings* are usually strings known to have undesired secondary structure. Alternatively, strands are designed using only a *three-letter alphabet* (A, C, T for DNA, A, C, U for RNA) to eliminate the potential for GC pairs which could cause unwanted secondary structure [22].

In order to design a strand with a desired secondary structure, the nucleotides at positions which bond together must be complementary. This simple approach can be improved by also requiring the strands to satisfy some free-energy-based criteria, such as those described below from Dirks et al. [10].

The *minimum free energy* constraint, which can be calculated in $O(N^3)$ time for structures with no pseudoknots [23], is used to choose sequences such that the target structure is the structure with the minimum free energy. This method, however, does not ensure that there are no other structures that the sequence is likely to form. Algorithms exist which test that a given set of singly-stranded DNA strands have no unspecific hybridizations for word sets based on the minimum free energy constraint [24, 25].

The *energy minimization* constraint is used to choose sequences which have a low free energy in the target structure, but not necessarily the minimum free energy. To design strands with this constraint, first generate a random string s that satisfies the complementary requirements of the desired secondary structure. For each step (Dirks used 10^6 steps) choose a random one-point mutation. Let s' be the sequence with this random one-point mutation (and a mutation in the corresponding base required by the structure constraint, if any). Accept the mutation by replacing s with s' if:

$$e^{-\frac{\Delta G(s') - \Delta G(s)}{RT}} \geq \rho$$

where $\rho \in [0,1]$ is a random number drawn from a uniform distribution. Thus this equation always accepts any mutations which result in no change or a decrease in free energy, and accepts with some probability any mutations which increase the free energy.

The free energy of a structure can be calculated in $O(N)$ time.

Sequences can also be chosen which maximize the *probability* of sampling the target structure. The probability $p(s)$ that every nucleotide in the sequence σ exactly matches the target structure s at thermodynamic equilibrium is calculated by:

$$p(s) = \frac{1}{Q}\, e^{-\frac{\Delta G(s)}{RT}}$$

where $\Delta G(s)$ is the free energy of sequence σ in secondary structure s. The partition function, Q, is:

$$Q = \sum_{s \in \Omega} e^{-\frac{\Delta G(s)}{RT}}$$

where Ω is the set of all secondary structures that sequence σ can form in equilibrium. If s^* is the target secondary structure and $p(s^*) \approx 1$ then σ has a high affinity and high specificity for s^*. An optimal dynamic programming algorithm calculates $p(s^*)$ for structures with no pseudoknots in $O(N^3)$ time [13]. $p(s^*)$ for secondary structures with pseudoknots can be calculated in $O(N^5)$ time.

Additionally, sequences can be chosen to minimize the *average number of incorrect nucleotides*, $n(s)$, over all equilibrium secondary structures Ω. For $1 \leq i \leq N$ and $1 \leq j \leq N$, the structure matrix $S(s)$ for the sequence σ of length N in structure s is:

$$S(s)_{i,j} = \begin{cases} 1, \text{ if base } i \text{ is paired with base } j \text{ in } s \\ 0, \text{ otherwise} \end{cases}$$

$$S(s)_{i,N+1} = \begin{cases} 1, \text{ if base } i \text{ is unpaired in } s \\ 0, \text{ otherwise} \end{cases}$$

$S(s)$ can be thought of as a matrix with elements that are 0 or 1. The sum of each row of $S(s)$ is 1. For $1 \leq i \leq N$ and $1 \leq j \leq N$, the probability matrix $P(s)$ is:

$$P(s)_{i,j} = \sum_{s \in \Omega} p(s) S(s)_{i,j}$$

where $P(s)_{i,j}$ is the probability of forming a base pair between the nucleotides at position i and j. $P(s)_{i,N+1}$ is the probability that base i is unpaired. $P(s)$ can be thought of as a matrix with elements that are real numbers in $[0,1]$, and the sum of each row of $P(s)$ is 1. $n(s)$ is the average number of incorrect nucleotides over the equilibrium ensemble of secondary structures Ω. If s^* is the target structure then:

$$n(s^*) \quad = \quad N - \sum_{i=1}^{N} \sum_{j=1}^{N+1} P(s)_{i,j} S(s^*)_{i,j}$$

$n(s^*)$ can be calculated in $O(N^3)$ time in structures with no pseudoknots and $O(N^5)$ in structures with pseudoknots.

The best-performing models are probability, average incorrect nucleotides, and energy minimization in combination with sequence symmetry minimization for the substrings that are not constrained by the desired secondary structure. The middle-performing models are the negative design methods (minimum free energy, and sequence symmetry minimization alone). The worst performing model is energy minimization (a positive design method). Surprisingly, minimum free energy performs similarly to sequence symmetry minimization. These results show that free energy measurements do not guarantee good design; an effective search must use both positive and negative design methods.

3.2 Secondary Structure of Duplexes

The Watson-Crick complement of a strand is obtained by reversing it and then complementing each base. 'Perfectly matched' strands are Watson-Crick complements of each other. In general, two DNA strands that are Watson-Crick complementary tend strongly to bind together. However different sequences have relatively stronger or weaker tendencies to bind with their perfect complements. In addition, some mismatched pairs of strands can also form stable structures, and different mismatched pairs can also have stronger or weaker tendencies to bond. In general, mismatched strands are less stable than Watson-Crick-complementary sequences.

DNA has two types of bonds that determine its secondary structure. The nucleotides in a single strand are held together by phosphodiester bonds. Hydrogen bonds form between nucleotides of separate chains. The change in free energy when a perfectly matched duplex forms is often estimated by either (1) the type of hydrogen bonds, AT vs. GC, expressed as the percentage of nucleotides that are G and C bases in a strand or duplex, which is known

as *GC-content*; or (2) both the hydrogen and the phosphodiester bonds, which is the nearest-neighbor model.

Since GC base pairs are held together by three hydrogen bonds and AT base pairs are held together by only two hydrogen bonds, double-stranded DNA with a high GC content is *often* more stable than DNA with a high AT content. Many DNA library searches require each strand to have a 50% GC-content to balance the requirement of stable matched hybridizations for identification purposes with the requirements of denaturation. The advantage of using the GC-content heuristic is that it is simple to calculate; only the length and the number of GC bases are needed, where the length refers to the number of nucleotide base pairs. However a disadvantage is that the nearest-neighbor heuristic is more accurate than the GC-content heuristic because the nearest neighbor base stacking energies account for more of the change in free energy than the energy of the hydrogen bonding between nucleotide bases. Thus the GC-content measure is a coarse heuristic for indirectly estimating the stability of a duplex, whereas the nearest-neighbor model attempts to approximate the actual change in free energy.

Digital codes and DNA are similar because codes are used to store information in digital strings, and DNA can be thought of as their biological equivalent. Thus, many early attempts to describe the differences between two DNA strings used results from coding theory. Requiring all pairs of strings in the library to have at least a given minimum *Hamming distance* (i.e., the number of characters in corresponding places which differ between two strings), is intended to satisfy the requirement that no pair of strings in the library should hybridize. Other extensions to Hamming distance have been developed in the literature. For example, the *reverse complement Hamming distance* is the number of corresponding positions which differ in the complement of s_1 and the reverse of s_2 (where s_1 and s_2 are not Watson-Crick complementary). This constraint is used to reduce the false positives that occur from hybridization between a word and the reverse of another word in the library. For more information on algorithms for strings see [26].

The advantage of Hamming distance (and its variations) is its theoretical simplicity and the vast body of extant work in coding theory. Many bounds have been calculated on the optimal size of codes with various Hamming-distance-based constraints [27]. Many early search algorithms used only Hamming distance as a constraint to develop combinatorial algorithms based on the results from coding theory. However, Hamming distance alone appears to be a problematic constraint.

One problem with Hamming-distance-based heuristics is that this measure assumes that position i of the first string is aligned with position i of the second string. However, since duplexes can be formed with dangling ends and loops, this is not the only possible alignment. Various *Hamming distance slides*, *substring uniqueness* [28], partial words [29], and H-measure [30] constraints have been developed to fix the alignment problem.

Another problem with heuristics based on Hamming distance is that the percentage of matching base pairs necessary to form a duplex is not necessarily known. Melting temperature can be used to approximate what the minimum Hamming distance should be.[3] However, for a given temperature and word set, there can be significant variation in the required minimum distance, because the necessary distance depends on the reaction conditions, especially the temperature.

Now that accurate free-energy information is available for all but the most complicated secondary structures (e.g., branching loops), the nearest-neighbor model is a much more accurate method to use than the constraints based on Hamming distance. One possible way of using free-energy-based calculations as a constraint to prevent mismatched duplexes is to maximize the gap between the free energy of the weakest specific hybridization and the free energy of strongest nonspecific hybridization, which we refer to as the *energy gap*; this approach was used by Penchovsky [31]. A metric also exists which calculates the maximum number of bonded base pairs formed from two strands using the nearest neighbor model [32]. The probability, $p(s^*)$, mea-

[3] Deaton estimates the melting temperature of mismatched duplexes by decreasing 1°C per 1% mismatch between oligonucleotides [4]. Since this calculation is outdated (see Section 4), if this heuristic is used for a library search, it is recommended that the melting temperature should be estimated from free energy calculations.

surement could also be applied to duplexes. A reasonable heuristic would be to maximize the gap between the lowest probability of the desired specific hybridizations and the highest probability of undesired non-specific hybridizations, which we refer to as the *probability gap*. Algorithms exist which calculate the probability, $p(s^*)$, for all possible combinations of single and double stranded foldings between a pair of strands [33]. Various equilibrium thermodynamic approaches have been used [34–38]. Computational incoherence, ξ, predicts the probability of error hybridization per-structure based on statistical thermodynamics [34, 39].

The physically-based models can be divided into categories based on the level of chemical detail [40]. Techniques which model single molecules include molecular mechanics models such as Monte Carlo minimum free energy simulations and molecular dynamics which models the change of the system with time. Techniques which average system behavior, or mass action approaches, are less accurate but more computationally feasible. Molecular mechanics (which models the movement of the system to the lowest energy), chemical kinetics, melting temperature, and statistical thermodynamics are all mass action approaches.

4 Melting Temperature

Melting temperature is typically used as a constraint in DNA paradigms that use multiple hybridization and denaturation steps to identify the answer, for an example see [1]. When DNA is heated considerably above physiological temperatures, to 100°C, the hydrogen bonds that bind two bases together tend to break apart, and the strands tend to separate from each other. The probability that a bond will break increases with temperature. This probability can be described by the melting temperature, which is the temperature in equilibrium at which 50% of the oligonucleotides have hybridized to their perfect complements and 50% of the oligonucleotides are separated. Since temperature control is often used to help denature the strands in between steps, it is advantageous for these paradigms to require all of the strands in the library to have similar melting temperatures or melting temperatures above some threshold.

The melting temperature of a perfectly matched duplex can be roughly estimated from the 2–4 rule [5] which predicts the melting temperature as twice the number of AT base pairs plus 4 times the number of GC base pairs. Another rough estimate of the change in melting temperature due to mismatched duplexes can also be obtained by decreasing the melting temperature of a corresponding matched duplex by 1°C per 1% mismatch; unfortunately, the inaccuracy is typically greater than 10°C [11]. Neither method is recommended. A better method is to use the nearest-neighbor model regardless of whether the duplex is perfectly matched or mismatched. This method produces more accurate results because melting temperature is closely related to free energy and can be predicted from the nearest-neighbor model. Melting temperature has been used to characterize the hybridization potential of a duplex [41, 42], but this measure cannot be used to predict whether two strands are bound at a given temperature and the melting temperatures of different duplexes do not necessarily correspond to relative rankings of stability.

5 Reaction Rates

Once the structure of candidate strands is known, the next logical question to ask is how fast do these reactions occur and what concentration is needed. Kinetics deals with the rate of change of reactions. For some implementations of DNA computers, the rate of the reaction could be an additional search constraint. System-level simulation software has been described for this purpose [43].

6 Evaluating a Set of DNA Strands for DNA Computation

Of the heuristics previously mentioned, the most appropriate method for obtaining an estimate of the absolute or relative rate of hybridization error is thermodynamics and statistical thermodynamics. For example, $p(s^*)$, $n(s^*)$, pair probabilities, and free energy have been used to evaluate whether a singly stranded sequence will have the desired secondary structure, s^* [10]. Statistical thermodynamics (the

partition function of all hybridized configurations) has been used to evaluate the set of strands used in Adleman's original Hamiltonian Path problem by predicting the error rate [44]. Computational incoherence [34,39], ξ, could also be used for evaluation. In addition, the energy gap or probability gap could be used for evaluation. The final evaluation criterion must be how the strands perform in the laboratory, since this is what the library is ultimately designed for.

7 DNA Prediction Software

RNA free energy nearest neighbor parameters are available from the Turner Group [45]. MFold [19], Hyther [20, 46, 47], the Vienna Package [48], NUPACK [49, 50], EdnaCo [51], NucleicPark [38], RNAstructure [45], Dynalign [52], and RNAsoft [53] are DNA/RNA structure prediction software which is available on the web. Visual OMP (Oligonucleotide Modeling Platform; DNA Software Inc.) [54] is commercially available software. Kinfold [55] is software for kinetic simulation.

8 Conclusion

Structure prediction can be separated into two problems. The first is to understand how DNA folds in nature. The second is to understand how computers should fold DNA strands to obtain the structure. Since nature has the advantage of parallel processing and the proximity of the molecules in space, the way nature finds the solution to the folding problem should not necessarily be the same as the way a computer finds the solution. Early algorithms to find DNA word sets focused on the Hamming distance constraint or variations thereof to achieve a theoretical abstraction of the constraints, which allowed the use of combinatorial algorithms and proofs of completeness (i.e., that the size of the pool is optimal or near optimal). However, in the process the constraints are simplified so much that they no longer accurately predict DNA structure. Current algorithms tend to use a more complex combination of the constraints. However, since these constraints are difficult to abstract, more recent programs resort to genetic algorithms, random search, exhaustive search, and local stochastic search algorithms.

Thermodynamics, melting temperature and kinetics are best at predicting DNA structure and reaction rates. Calculating thermodynamics and kinetics can be costly, however. According to the requirements mentioned for the negative design problem, checking that a library of size M meets specifications requires $O(M^2)$ string comparisons, where each comparison of a pair of strings of length N is potentially polynomial in N. However, this does not mean that the weaker combinatorial and heuristic predictors are useless. Many of these alternative structure heuristics could be used to quickly filter a candidate set of library molecules, and then the free energy model could be used to more accurately check this set. If this approach is adopted, the correlation between these alternative heuristics and free energy measurements should be explored. Alternatively, free energy or probability approximation algorithms could be used. This approach has the advantage that techniques from randomized algorithm analysis could be used to prove the correctness of the approximation.

Research in DNA libraries has two main goals: (1) to further understand DNA chemistry, and (2) to understand search techniques useful for constructing sets of DNA codes. Although there is a growing consensus that DNA computers will never be as practical or as fast as conventional computers, biological computers have the advantage that their style of computation is closer to natural processes. Successful research in DNA libraries will help to reduce errors in DNA computation and may discover new information about how DNA interacts with itself. Although current DNA computers are simplistic in comparison to natural biochemical processes, DNA computation may help to develop alternative theories for how cells work or could have evolved [56]. In addition, research in DNA design also pertains to DNA nanotechnology, PCR-based applications, and DNA arrays. Breakthroughs in this field will add to the current knowledge of DNA chemistry as well as DNA computers.

9 Acknowledgments

We are grateful to Milan Stojanovic for his advice and encouragement, and to the anonymous reviewers

for their extensive comments. This material is based upon work supported by the National Science Foundation (grants CCR-0219587, CCR-0085792, EIA-0218262, EIA-0238027, and EIA-0324845), Sandia National Laboratories, Microsoft Research, and Hewlett-Packard (gift 88425.1). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

[1] Dirk Faulhammer, Anthony R. Cukras, Richard J. Lipton, and Laura F. Landweber. Molecular computation: RNA solutions to chess problems. *Proceedings of the National Academy of Sciences of the USA (PNAS)*, 97(4):1385–1389, February 2000. The PERMUTE Program is available at `http://www.pnas.org/cgi/content/full/97/4/1385/DC1`.

[2] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, November 1994.

[3] Richard J. Lipton. DNA solution of hard computational problems. *Science*, 268:542–545, April 1995.

[4] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr. Good encodings for DNA-based solutions to combinatorial problems. In Landweber and Baum [57], pages 247–258.

[5] Arwen Brenneman and Anne E. Condon. Strand design for biomolecular computation. Technical report, University of British Columbia, March 2001.

[6] G. Mauri and C. Ferretti. Word design for molecular computing: A survey. In Junghuei Chen and John H. Reif, editors, *DNA Computing: 9th International Workshop on DNA-Based Computers, DNA 2003 (University of Wisconsin: Madison, WI)*, volume 2943 of *Lecture Notes in Computer Science*, pages 37–47. Springer, 2004.

[7] R. Deaton and M. Garzon. Thermodynamic constraints on DNA-based computing. In Gheorghe Păun, editor, *Computing with Bio-Molecules*, pages 138–152. Springer-Verlag, Singapore, 1998.

[8] James D. Watson, Nancy H. Hopkins, Jeffrey W. Roberts, Joan Argetsinger Steitz, and Alan M. Weiner. *Molecular Biology of the Gene*. Benjamin/Cummings, Menlo Park, CA, fourth edition, 1988.

[9] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland, New York, fourth edition, 2002.

[10] Robert M. Dirks, Milo Lin, Erik Winfree, and Niles A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, 2004.

[11] John SantaLucia, Jr. and Donald Hicks. The thermodynamics of DNA structural motifs. *Annual Review of Biophysics Biomolecular Structure*, 33:415–40, June 2004.

[12] Peter Schuster. Counting and maximum matching of RNA structures. Preprint, `http://www.tbi.univie.ac.at/~pks` accessed on 2/1/2005, January 2004.

[13] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, May-Jun 1990.

[14] Ignacio Tinoco, Jr., Kenneth Sauer, James C. Wang, and Joseph D. Puglisi. *Physical Chemistry: Principles and Applications in Biological Sciences*. Prentice Hall, fourth edition, 2002.

[15] Mitsuhiro Kubota and Masami Hagiya. Minimum basin algorithm: An effective analysis technique for dna energy landscapes. In *Preliminary Proceedings of 10th International Workshop on DNA-Based Computers, DNA 2004 (University of Milano-Bicocca: Milan, Italy)*, pages 202–213, 2004.

[16] Peter Schuster, Peter F. Stadler, and Alexander Renner. RNA structures and folding: from conventional to new issues in structure predictions. *Current Opinion in Structural Biology*, 7(2):229–235, April 1997.

[17] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, June 1988.

[18] Milan N. Stojanovic and Darko Stefanovic. A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21(9):1069–1074, September 2003.

[19] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003. Mfold is available at `http://www.bioinfo.rpi.edu/applications/mfold`.

[20] John SantaLucia, Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences of the USA (PNAS)*, 95:1460–1465, 1998.

[21] N. Peyret. *Prediction of Nucleic Acid Hybridization: Parameters and Algorithms*. PhD thesis, Wayne State University, Dept. of Chemistry, 2000.

[22] Kalim U. Mir. A restricted genetic alphabet for DNA computing. In Landweber and Baum [57].

[23] Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.

[24] Mirela Andronescu, Danielle Dees, Laura Slaybaugh, Yinglei Zhao, Anne Condon, Barry Cohen, and Steven Skiena. Algorithms for testing that sets of DNA word designs avoid unwanted secondary structure. In Masami Hagiya and Azuma Ohuchi, editors, *DNA Computing: 8th International Workshop on DNA-Based Computers, DNA 2002 (Hokkaido University: Sapporo, Japan)*, volume 2568 of *Lecture Notes in Computer Science*, pages 182–195. Springer, 2003.

[25] Satoshi Kobayashi. Testing structure freeness of regular sets of biomolecular sequences. In *Preliminary Proceedings of 10th International Workshop on DNA-Based Computers, DNA 2004 (University of Milano-Bicocca: Milan, Italy)*, pages 395–404, 2004.

[26] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, New York, 1999.

[27] Amit Marathe, Anne E. Condon, and Robert M. Corn. On combinatorial DNA word design. *Journal of Computational Biology*, 8(3):201–220, 2001.

[28] Udo Feldkamp, Hilmar Rauhe, and Wolfgang Banzhaf. Software tools for DNA sequence design. *Genetic Programming and Evolvable Machines*, 4(2):153–171, June 2003.

[29] Peter Leupold. Partial words for DNA coding. In *Preliminary Proceedings of 10th International Workshop on DNA-Based Computers, DNA 2004 (University of Milano-Bicocca: Milan, Italy)*, 2004.

[30] M. Garzon, P. Neathery, R. Deaton, R.C. Murphy, D.R. Franceschetti, and S. E. Stevens, Jr. A new metric for DNA computing. In *Proceedings 2nd Genetic Programming Conference*, pages 472–478, 1997.

[31] Robert Penchovsky and Jorg Ackermann. DNA library design for molecular computation. *Journal of Computational Biology*, 10(2):215–229, 2003.

[32] Arkadii G. D'Yachkov, Anthony J. Macula, Wendy K. Pogozelski, Thomas E. Renz, Vyacheslav V. Rykov, and David C. Torney. A weighted insertion-deletion stacked pair thermodynamic metric for DNA codes. In *Preliminary Proceedings of 10th International Workshop on DNA-Based Computers, DNA 2004 (University of Milano-Bicocca: Milan, Italy)*, 2004.

[33] Roumen A. Dimitrov and Michael Zuker. Prediction of hybridization and melting for double-stranded nucleic acids. *Biophysical Journal*, 87:215–226, July 2004.

[34] J. A. Rose, R. J. Deaton, D. R. Franceschetti, M. Garzon, and S. E. Stevens, Jr. A statistical mechanical treatment of error in the annealing biostep of DNA computation. In *Special program in GECCO-99*, pages 1829–1834, June 1999.

[35] John A. Rose and Russell J. Deaton. The fidelity of annealing-ligation: A theoretical analysis. In Anne Condon and Grzegorz Rozenberg, editors, *DNA Computing: 6th International Workshop on DNA-Based Computers, DNA 2000 (Leiden Center for Natural Computing: Leiden, The Netherlands)*, volume 2054 of *Lecture Notes in Computer Science*. Springer, 2001.

[36] John A. Rose, Russell J. Deaton, Masami Hayiya, and Akira Suyama. The fidelity of the tag-antitag system. In Nataša Jonoska and Nadrian C. Seeman, editors, *DNA Computing: 7th International Workshop on DNA-Based Computers, DNA 2001 (University of South Florida: Tampa, FL)*, volume 2340 of *Lecture Notes in Computer Science*. Springer, 2002.

[37] J. A. Rose, R. J. Deaton, M. Hagiya, and A. Suyama. An equilibrium analysis of the efficiency of an autonomous molecular computer. *Physical Review E*, 65(021910), 2002.

[38] John A. Rose, Masami Hagiya, and Akira Suyama. The fidelity of the tag-antitag system II: Reconcilation with the stringency picture. In *Proceedings of the Congress on Evolutionary Computation*, pages 2749–2749, 2003. NucleicPark is available at `http://hagi.is.s.u-tokyo.ac.jp/johnrose/andhttp://engronline.ee.memphis.edu/molec/demos.htm`.

[39] J. A. Rose, R. J. Deaton, D. R. Franceschetti, M. Garzon, and S. E. Stevens, Jr. Hybridization error for DNA mixtures of N species. *submitted to Physical Review Letters*, 1999.

[40] John A. Rose and Akira Suyama. Physical modeling of biomolecular computers: Models, limitations, and experimental validation. *Natural Computing*, 3(4):411–426, 2004.

[41] Alexander J. Hartemink and David K. Gifford. Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania, June 23 – 25, 1997*, pages 15–25, 1997.

[42] Alexander J. Hartemink, David K. Gifford, and Julia Khodor. Automated constraint-based nucleotide sequence selection for DNA computation. *4th Annual DIMACS Workshop on DNA-Based Computers, Philadelphia, Pennsylvania*, June 1998.

[43] Akio Nishikawa, Masayuki Yamamura, and Masami Hagiya. DNA computation simulator based on abstract bases. *Soft Computing*, 5(1):25–38, 2001.

[44] R. Deaton and J. A. Rose. Simulations of statistical mechanical estimates of hybridization error. In Anne Condon and Grzegorz Rozenberg, editors, *Preliminary Proceedings of 6th International Workshop on DNA-Based Computers, DNA 2000 (Leiden Center for Natural Computing: Leiden, The Netherlands)*, pages 251–252, June 2000.

[45] David H. Mathews, Matthew D. Disney, Jessica L. Childs, Susan J. Schroeder, Michael Zucker, and Douglas H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the USA (PNAS)*, 101(19):7287–7292, May 2004. The free energy nearest neighbor parameters are available at `http://rna.chem.rochester.edu/`, RNAstructure is available at `http://128.151.176.70/RNAstructure.html`.

[46] Nicolas Peyret, Pirro Saro, and John SantaLucia, Jr. Hyther server. Hyther Version 1.0 is available at `http://ozone2.chem.wayne.edu/`.

[47] N. Peyret, P. A. Seneviratne, H. T. Allawi, and J. SantaLucia, Jr. Nearest-neighbor thermodynamics and NMR of DNA sequences with internal A-A, C-C, G-G, and T-T mismatches. *Biochemistry*, 38:3468–3477, 1999.

[48] Ivo Ludwig Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431, 2003. Vienna Package is available at `http://www.tbi.univie.ac.at/~ivo/RNA/`.

[49] Robert M. Dirks and Niles A. Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry*, 24(13):1664–1677, October 2003. NUPACK is available at `http://www.acm.caltech.edu/~niles/software.html`.

[50] Robert M. Dirks and Niles A. Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of Computational Chemistry*, 25:1295–1304, 2004.

[51] M.H. Garzon, R.J. Deaton, J.A. Rose, L. Lu, and D. Franceschetti. Soft molecular computing. *Proc. DNA5-99 Workshop, AMS DIMACS Series in Theoretical Computer Science*, 54:91–100, 2000. EdnaCo is available at `http://zorro.cs.memphis.edu/~cswebadm/csweb/research/pages/bmc/` or `http://engronline.ee.memphis.edu/molec/demos.htm`.

[52] D. H. Mathews and D. H. Turner. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, 317(217):191–203, 2002.

[53] Mirela Andronescu, Rosalia Aguirre-Hernandez, Anne Condon, and Holger H. Hoos. RNAsoft: a suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Research*, 31(13):3416–3422, 2003. RNAsoft is available at `http://www.rnasoft.ca/`.

[54] Visual OMP (Oligonucleotide Modeling Platform), DNA Software, Inc. Visual OMP is available at `http://www.dnasoftware.com`.

[55] Christoph Flamm, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000. Kinfold is available at `http://www.tbi.univie.ac.at/~xtof/RNA/Kinfold/`.

[56] Warren D. Smith. DNA computers in vitro and vivo. In Richard J. Lipton and Eric B. Baum, editors, *DNA Based Computers, DIMACS Workshop 1995 (Princeton University: Princeton, NJ)*, volume 27 of *Series in Discrete Mathematics and Theoretical Computer Science*, pages 121–185. American Mathematical Society, 1996.

[57] Laura F. Landweber and Eric B. Baum, editors. *DNA Based Computers II, DIMACS Workshop 1996 (Princeton University: Princeton, NJ)*, volume 44 of *Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1999.