Thermodynamic Simulation of Deoxyoligonucleotide Hybridization, Polymerization, and Ligation

by

Alexander J. Hartemink

A.B., Economics
B.S., Mathematics
B.S., Physics
Duke University (1994)

M.Phil., Economics
Oxford University (1996)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

September 1997

Author ....
Department of Electrical Engineering and Computer Science
13 August 1997

Certified by ....
David K. Gifford
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Professor of Electrical Engineering and Computer Science
Chairman, Committee on Graduate Students

Thermodynamic Simulation of Deoxyoligonucleotide Hybridization, Polymerization, and Ligation

by

Alexander J. Hartemink

Submitted to the Department of Electrical Engineering and Computer Science
on 13 August 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Two programs, BIND and SCAN, are implemented in order to better predict the hybridization specificity of arbitrary deoxyoligonucleotides and design DNA-based computational systems exhibiting superior hybridization discrimination. These programs dramatically reduce the time required to analyze and design systems relying on hybridization specificity for their successful operation.

BIND is a hybridization simulator which computes the theoretical melting temperature between strands of DNA. After the user provides BIND with a template strand and any number of oligonucleotide primer strands, the theoretical melting temperatures of nascent primer/template duplexes are calculated using thermodynamic enthalpy and entropy values for the nearest neighbor stacking interactions between the template and primer at the given reactant and ionic concentrations. The simulator differs from previous melting temperature programs in that it is intended to be used with oligonucleotides, is designed to handle mismatched base pairs, makes use of the latest thermodynamic parameters, and provides features with DNA computation expressly in mind. BIND is being used to simulate DNA-based computational systems within the context of the programmed mutagenesis paradigm, but because of the properties distinguishing it from other simulators, it should also prove valuable to the larger DNA computing and molecular biological communities.

While BIND evaluates strands of DNA whose composition is known, SCAN was developed to tackle the inverse problem: selecting DNA sequences to satisfy predetermined hybridization specificity and secondary structure criteria. SCAN was used to design a new unary counter machine, winnowing billions of candidate sequences down to just two designs. The new unary counter designs are currently being manufactured and tested.

After a brief introduction to DNA computing, and programmed mutagenesis in particular, the thesis describes how BIND was implemented, provides corroborating evidence as to its accuracy, and offers instances of its usefulness to a range of DNA computing applications. It also illustrates how SCAN was used to design a new DNA-based unary counter.

Thesis Supervisor: David K. Gifford
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

The author would like to thank Professor David K. Gifford for both his insight and his oversight during the course of this research project, and also Julia Khodor for her generous assistance in the provision of laboratory gels and experiment protocols. In addition, the author gratefully acknowledges the support of the National Science Foundation which provides funding through its Graduate Research Fellowship program. Most importantly of all, the author wishes to thank his family for their unceasing and unsurpassing guidance and encouragement over the years.

Early research forming the basis of parts of this thesis was presented at the 3rd Annual DIMACS Workshop on DNA Based Computers and some of the text contained herein will also appear in those proceedings. The author would like to thank the DIMACS program committee for the opportunity to present some of the early results of this research at that workshop.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recent activity in the field of DNA computing has generated interest in the dynamics of nucleic acid interaction. By creating another consumer of precise information regarding enzyme activity, thermal and ionic denaturation, binding specificity, and the polymerase chain reaction (PCR), DNA computing has enhanced the already compelling case for deeper fundamental understanding of these processes.

To date, a number of approaches to DNA-based computation have been proposed. These approaches are generally quite disparate, but they all share a reliance on the nucleic acid hybridization process to implement particular primitive operations or to generate input DNA. To our knowledge, all current DNA-based computation proposals make use of hybridization steps, whether in constructing large libraries of DNA on which to perform computations [2, 4, 14], in using PCR to amplify or detect small quantities of DNA containing a specific sequence of nucleotides [2, 3], in performing sequence-specific separation operations using magnetic bead or affinity column techniques [2, 3, 4, 14], in setting specific bits in sticker-based models [14], in marking specific strands in surface-based models [7], or in performing sequence-specific mutagenic operations in programmed mutagenesis models. Because these operations all rely on site-specific nucleic acid hybridization, it is critical that significant progress be made in more precisely characterizing the way the temperature, ionic environment, and degree of mismatch influence the hybridization process.

By evaluating the current state of DNA-based computation, and by looking in the direction in which the field will need to expand in the next few years, it becomes clear that many of the proposed computational systems will need to move "from the blackboard into the laboratory" in order to address the subtle details associated with their implementation. Before particular problems can be investigated, however, the algorithms (and the input to those algorithms) will need to be translated into concrete DNA sequences, which is a daunting task, especially in light of the necessary binding specificity demands made by the systems' specifications. Because the design of DNA-based computational systems relies heavily on hybridization, it will be helpful to be able to simulate the conditions under which desired hybridization will occur. With this in mind, a simulator capable of quickly evaluating the hybridization specificity of oligonucleotides serving as PCR primers, stickers, markers, probes, or mutagenic rewrite rules should prove helpful to the DNA computing community, as well as to the molecular biological community at large.

1.1.1 The Importance of DNA Computing

Because of its inherent information storage capability, DNA has begun to be investigated as a possible substrate for computation. The sequence of nucleotides present in a strand of DNA is reminiscent of a Turing machine tape or random access memory in a conventional silicon-based computer, and as a result, many researchers believe that DNA could provide a basis for new models of computation.

The need for computational systems exhibiting higher speed, increased storage density, reduced power, and great versatility in the next century will be enormous. Nucleic acids have the capacity for such improved performance, in some cases many orders of magnitude beyond the limits of current technologies. As an example of this potential, consider the following parameters, calculated in the context of the programmed mutagenesis paradigm:

Higher Speed Because so many molecules of DNA can be present in a small volume of solution, it is possible for ultra-parallel computations to be performed in reasonably short amounts of time. For the purposes of comparison, consider that typical workstations today carry out approximately 10^7 low-level operations per second, while current supercomputing technologies are capable of carrying out approximately 10^{13} low-level operations per second. In contrast, using a 100 microliter volume of 1000-mer DNA in solution at a concentration of 100 nanograms per microliter, approximately 2.7×10^{11} high-level strand rewrite operations can be carried out per second, which corresponds to approximately 2.7×10^{14} low-level base pair polymerization operations per second. Not only does this already exceed current supercomputing technology limits, but the number of low-level operations per second scales linearly with the volume of solution, and it would therefore not be difficult to realize much faster speeds than that possible with this modest volume.

Increased Storage Density Because the volume occupied by a single nucleotide of DNA is so small, the information storage density of DNA is immense. Current information storage technologies require approximately 10^3 cubic micrometers to store a single bit of information. In solution at a concentration of 100 nanograms per microliter, the storage density of DNA is a remarkable 2×10^5 bits per cubic micrometer, more than eight orders of magnitude better. The storage density inherent in undissolved DNA is even more impressive at 1.25×10^9 bits per cubic micrometer.

Reduced Power Existing supercomputing technologies can carry out approximately 10^9 operations per joule, but a single joule provides enough energy to carry out the replication of approximately 10^{16} DNA molecules, each 1000 base pairs in length. Even when other energy requirements are factored into the calculation, the potential power consumption of a DNA-based computational system remains many orders of magnitude below that of existing technologies.

Great Versatility Since computation with DNA is likely to be universal, it follows that in theory any imaginable computation could be performed by a DNA-based computational system.

It remains to be seen which model of computation is best suited to harness this enormous potential, but many different paradigms have been proposed. Most employ combinatorial search techniques building on pioneering work by Adleman [2], but a programmatic approach relying on site-specific mutagenesis has recently been developed by Gifford [11]; this model of DNA computation, entitled programmed mutagenesis, provides the context for the research presented here.

1.1.2 The Importance of Programmed Mutagenesis

Because of its potential for parallelism, DNA presents an ideal medium for performing computations that are easily parallelizable or require amounts of time or space which would otherwise be impractical. Although fully realizing the inherent computational capability of DNA is still a long way off, programmed mutagenesis represents an important step towards attaining this goal.

In typical replication processes, both *in vivo* and *in vitro*, a single stranded sequence of DNA serves as the template for the polymerization of a complementary strand. Site-specific mutagenesis is a technique used to introduce specific changes into the sequence of the newly created strand during the replication process, producing two strands which are non-complementary.

The notion of programmed mutagenesis involves using a series of mutagenic operations to programmatically modify strands of DNA, with each operation enabling or disabling the actions of others; there can even be conditional transitions in which a particular transition is not allowed unless a previous one has already taken place. In this way, a series of changes can be introduced into the template strand which itself represents the state of the computation at any point in time. Unlike models of DNA computation being pursued elsewhere which rely on combinatorial search techniques, programmed mutagenesis offers the promise of programmatic computation. While combinatorial search techniques can only perform *selective computation*, programmed mutagenesis can perform *constructive computation.*

Another factor distinguishing programmed mutagenesis from other proposed models of DNA computation is that it has the potential to be much simpler to perform. Present combinatorial search techniques require many separation steps, which involve large amounts of laboratory manipulation. This not only makes such techniques labor intensive, but also introduces the potential for error, thereby reducing their accuracy. In contrast, programmed mutagenesis can be carried out without human intervention once the initial template and appropriate oligonucleotides have been selected. Since each sequence in the computation is the programmatic product of a previously existing computation sequence, and is produced by the reaction itself rather than being synthesized, the computation has the potential to be fully automated, obviating the need for human intervention.

Perhaps one of the greatest advantages of studying programmed mutagenesis is that investigation of this model of DNA computation does not preclude the study of other models, but rather reinforces it. Programmed mutagenesis is flexible enough to be synergistically combined with other techniques, such as combinatorial search. For example, one could imagine using programmed mutagenesis to perform a computation in parallel on a large class of initial states which are the result of some combinatorial search procedure.

Programmed mutagenesis is also powerful enough to perhaps lend some insights into the ways in which cells themselves may be exploiting some form of computation to direct developmental processes, control gene expression, regulate protein synthesis, or maintain internal clock mechanisms.

1.1.3 The Importance of Simulation

Currently, computation using biological substrates like DNA requires enormous amounts of time and the time required to carry out a computation is dwarfed by the time required to design the computation in the first place. In order to perform a particular computational task, an algorithm needs to be developed and then translated into an abstract DNA computing language. This abstract program must itself then be instantiated as a specific set of nucleotide sequences along with a protocol for the various accompanying chemical, thermal, and enzymatic operations. The actual design and evaluation of these materials and mechanisms is extremely time intensive, in addition to requiring considerable laboratory expertise. Experiments must be performed in order to determine how effectively various aspects of the computation are functioning and often the results require an entirely new design to be tested. Much of the time required to conduct the somewhat *ad hoc* approach to sequence selection and analysis could be eliminated if various parts of the process could be carried out in simulation.

For this reason, a simulator capable of evaluating the hybridization specificity of various primers and templates by calculating duplex melting temperatures would be of great use in reducing the time needed to design and implement a particular DNA-based computational system. Additionally, a program capable of performing an automated search for sequences satisfying various constraints would allow extremely optimized computational systems to be developed in a very short amount of time, saving years of laboratory work and tens or hundreds of thousands of dollars worth of reagents.

Programs which compute DNA melting temperatures do exist, but most either use outdated thermodynamic parameters or do not handle oligonucleotides, mismatched nucleotide binding, or multiple binding possibilities. Moreover, it seems that none of these programs is designed with DNA computation in mind. Therefore, a niche exists for an accurate simulator that uses the latest thermodynamic parameters, is capable of handling mismatched nucleotides, and is designed to provide needed functionality to researchers investigating DNA computation.

Furthermore, it appears that no programs exist which are capable of conducting a customizable search for nucleic acid sequences which are optimized with respect to a range of hybridization and secondary structure constraints.

1.2 Accomplishments

During the course of this research, two programs were designed and implemented in an attempt to dramatically reduce the amount of time required in sequence design and analysis. First, a thermodynamic simulator named BIND was implemented to predict the melting temperatures of DNA duplexes. The user provides the ionic and reactant concentrations, and the particular sequences of the strands under consideration, one serving as a template and the remainder serving as primers which can hybridize with the template. BIND then calculates, for each primer, the melting temperature of the nascent primer/template duplex at each possible binding position along the template. It reports the theoretical melting temperature of each such nascent duplex for every primer provided by the user. The resulting data can be used to generate plots revealing the most stable binding sites along the template for each of the primers.

Second, a program named SCAN was created to assist in searching through the billions of nucleotide combinations which are possible for a particular computational system design. Once the design is determined and various hybridization and secondary structure constraints are imposed, SCAN searches for nucleotide combinations satisfying all of the given constraints.

The benefits of simulation tools for designing machines using programmed mutagenesis are evident. However, tools like BIND and SCAN should also prove invaluable to other research groups involved in DNA computing, as well as those involved in any aspect of molecular biology which might benefit from detailed information about the hybridization of nucleic acids.

1.3 Overview of Thesis

The next chapter presents a brief introduction to the basic properties of DNA, as well as thermodynamic and enzymatic requirements for oligonucleotide hybridization, polymerization, and ligation. In order to provide contextual motivation for the development of simulation tools, chapter 3 briefly describes programmed mutagenesis and offers an example of how this technique can be used to construct a simple computational machine: the original unary counter. Chapter 4 proceeds to discuss the BIND simulator, deriving the thermodynamic equations used in its melting temperature calculations and evaluating its accuracy by comparing its predictions with melting temperature data determined in the laboratory and reported in the literature. Chapter 5 illustrates how BIND can be used to evaluate specific oligonucleotide sequence choices, both in the context of the original unary counter and also in the context of sticker model of DNA computation. Chapter 6 describes a program named SCAN which was implemented to tackle the inverse problem of selecting sequences of DNA to satisfy predetermined machine design constraints rather than evaluating sequences after they have already been chosen. The final chapter presents a number of possible extensions to BIND and SCAN which are currently being developed.

Chapter 2

A Molecular Biology Primer

2.1 Basic Properties of DNA

Deoxyribonucleic acid (DNA) is a heteromeric molecule consisting of a sequence of repeated nucleotide units, covalently linked together into a long chain. Each nucleotide consists of a purine or pyrimidine base attached to a deoxyribose sugar. The sugar ring is in turn coupled with a phosphate group which is used to link the nucleotide to the sugar ring of the neighboring nucleotide in the chain, as depicted in figures 2.1 and 2.2.

In the case of naturally occurring DNA, the bases are either adenine, cytosine, guanine, or thymine, commonly represented by their first initials, A, C, G, or T, respectively. As demonstrated by Watson and Crick, *in vivo* deoxyribonucleic acid typically assumes a helical double stranded conformation, and in this conformation, nucleotides containing adenine tend to pair with nucleotides containing thymine while nucleotides containing cytosine tend to pair with nucleotides containing guanine. For this reason, adenine is called the Watson-Crick complement of thymine, and *vice versa*, and cytosine is called the Watson-Crick complement of guanine, and *vice versa.* These complementary base pairs can hydrogen bond to one another, and it is this hydrogen bonding which enables the two opposing strands to hybridize and enter a helical double stranded conformation.

While strands of DNA are often represented as a seemingly reversible sequence of bases,

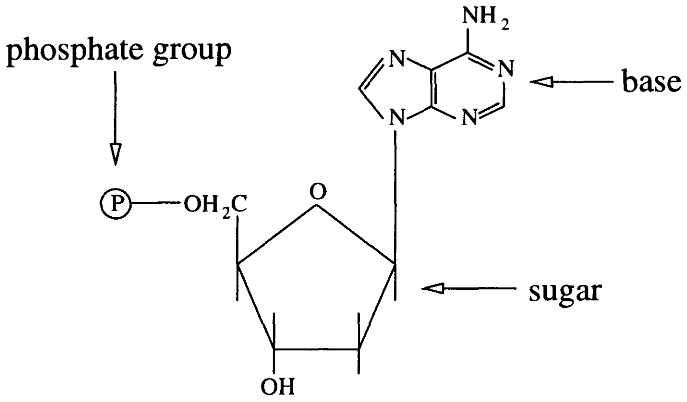

Figure 2.1: *A single deoxyribonucleotide: A deoxyribose sugar ring is bound to a phosphate group and to a base, which is the purine adenine in this case.*

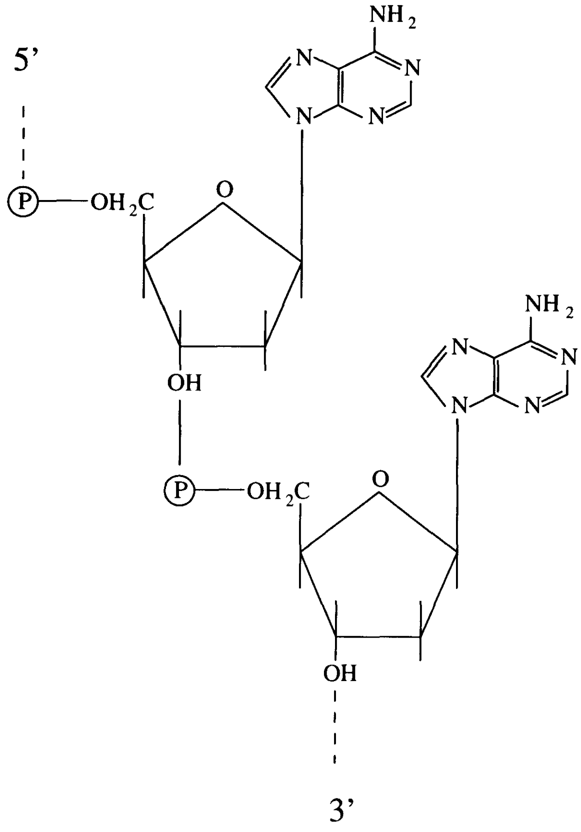

Figure 2.2: *Two deoxyribonucleotides covalently linked together: The phosphate group from one nucleotide bonds to the sugar ring of the next.*

it is important to realize the the molecule itself is directed. Because of the arrangements of the two phosphate groups on each sugar ring in a strand of DNA, one end is labelled the $5'$ end and the other end is labelled the $3'$ end. The two strands in any duplex are anti-parallel and consequently always run in opposite directions. By convention, DNA sequences are always written in the $5' \rightarrow 3'$ direction.

Under various circumstances, the hydrogen bonds holding the double stranded duplex together can be broken and the constituent strands can dissociate from one another. This process is catalyzed during replication *in vivo* under the guidance of enzymes like DNA helicase but can also be induced by lowered ionic concentration, increased temperature, or extreme levels of pH. For a given ionic environment and at a given level of pH, the temperature necessary to cause the dissociation of double stranded DNA can be calculated based on thermodynamic considerations.

2.2 Thermodynamics of DNA Hybridization

Nucleic acid hybridization is a chemical reaction like any other and therefore equilibrium conditions can be determined with sufficient knowledge of the thermodynamic parameters governing the transition. In the case of reversible reactions, reactants are continually interacting to form products and *vice versa.* The rates at which these transitions are called forward and reverse rates, respectively. Equilibrium is the condition of stasis and thus occurs when the forward and reverse rates are equal to one another.

Equilibrium conditions can also be calculated using thermodynamic principles. In particular, the net energetic change associated with any transition is defined as the Gibbs free energy, typically indicated $\Delta G°$. Free energy is defined as the weighted sum of two other thermodynamic parameters, enthalpy and entropy, typically indicated $\Delta H°$ and $\Delta S°$ respectively:

$$\Delta G° = \Delta H° - T\Delta S° \tag{2.1}$$

where T is the temperature in Kelvin. Fortunately, the transition enthalpy and entropy of oligonucleotide hybridization can be estimated from knowledge of the specific nucleotide sequences. Consequently, the melting temperature of a nucleic acid duplex can in general be predicted simply from knowledge of the two base pair sequences comprising the duplex.

A primitive technique for the prediction of duplex melting temperature was proposed by Marmur and Doty [12] over three decades ago. This technique relied solely on the relative frequency of the various bases, rather than their sequence, and was well suited as a description of polynucleotide melting but was not intended as a description of oligonucleotide melting.

In the early 1970s, a more sophisticated model for the prediction of oligonucleotide melting temperature was proposed. It was based on nearest neighbor stacking interactions and accounted for specific nucleotide sequence, to a first order approximation. In this model, each pair of adjacent base pairs in the duplex (nearest neighbors) is said to form a stack (as shown in figure 2.3). The total enthalpy and entropy associated with the formation of the duplex is then simply expressed as the sum of the enthalpies and entropies associated with the formation of all the stacks, plus a few other terms. With this model, total enthalpy and entropy are sequence dependent in that they depend on how neighboring base pairs interact with one another.

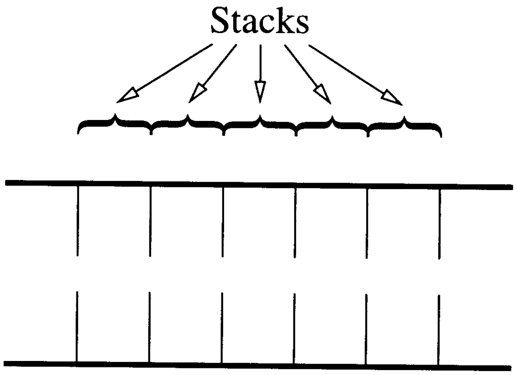

Figure 2.3: *Nearest neighbor stacking interactions: Each pair of adjoining base pairs forms a single stack.*

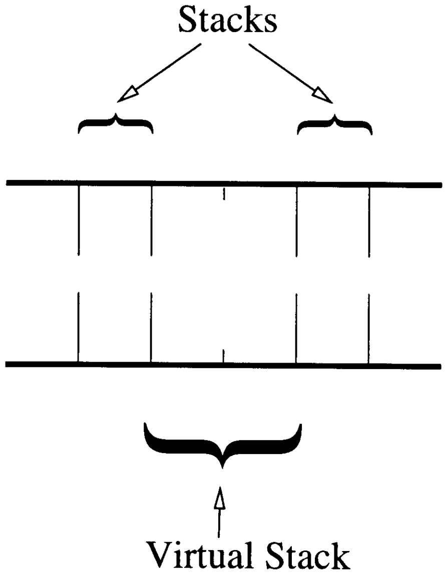

Figure 2.4: *Stacking interactions in the presence of mismatched base pairs: Mismatched base pairs form a "virtual stack" with adjoining base pairs.*

The model is not practical without knowing the thermodynamic parameters associated with the formation of the various stacks. So later work has attempted to develop sets of precise values for these parameters. Breslauer, *et al.* [6] published a complete nearest neighbor basis set in 1986; updated values for the entropic data were presented by Quartin and Wetmur [13] three years later. The most recent published set of nearest neighbor enthalpy and entropy values was compiled in 1996 by SantaLucia, *et al.* [16].

Since arbitrary duplexes may not consist entirely of Watson-Crick complementary base pairs, more data is needed when mismatched base pairs are present. Notable research aiming to characterize the effects of mismatched base pairs has been undertaken by Aboul-ela, *et al.* [1], Werntges, *et al.* [18], and Zenkova and Karpova [21]. In particular, Werntges, *et al.* [18] present a model in which the mismatched base pair and its adjoining neighbors are treated as a single "virtual stack" (see figure 2.4). This enables total enthalpy and entropy to be expressed as a summation of individual stack enthalpies and entropies as before.

The ionic environment in which DNA hybridization occurs also affects the equilibrium properties of the transition. Since the backbone of any strand of DNA contains negatively charged phosphate groups, the presence of positively charged ions in solution tends to stabilize the molecule, thereby raising its melting temperature. Results quantifying the influence of ionic concentration on duplex melting temperature have been presented by Wetmur [19] and Braunlin and Bloomfield [5].

Using the nearest neighbor model as developed in these papers to determine the enthalpy and entropy of oligonucleotide hybridization, together with knowing the ionic and reactant concentrations present in the reaction, the melting temperatures of arbitrary DNA duplexes can be straightforwardly predicted. A detailed derivation will be presented in chapter 4.

2.3 Enzyme Function

Enzymes are proteins present *in vivo* which serve to catalyze various chemical reactions, enabling them to take place when they would not otherwise be energetically favored. There are thousands of enzymes at work in complex organisms and the task of characterizing enzyme structure and function is extremely difficult. Nevertheless, a number of enzymes have been studied rather intensively and many of these are widely used within the molecular biological community because of their ability to manipulate nucleic acids like DNA.

Within the field of DNA computing, commonly used enzymes include kinases, phosphatases, and exo- and endonucleases (including restriction enzymes), but perhaps the two most important classes of enzymes are polymerases and ligases.

DNA polymerases are used to synthesize new strands of DNA, one nucleotide at a time. They operate by moving along an existing "template" strand of DNA and adding bases to a "primer" strand of DNA, complementing the bases contained in the template (see figure 2.5). Polymerases are only capable of adding bases to the $3'$ end of an existing strand of DNA so they are said to extend a strand in the $5' \rightarrow 3'$ direction.

DNA ligases catalyze the formation of covalent bonds between abutting strands of DNA. If two strands of DNA are hybridized to a template and are immediately adjacent to one another, the two can be linked together into a single longer strand through the action of

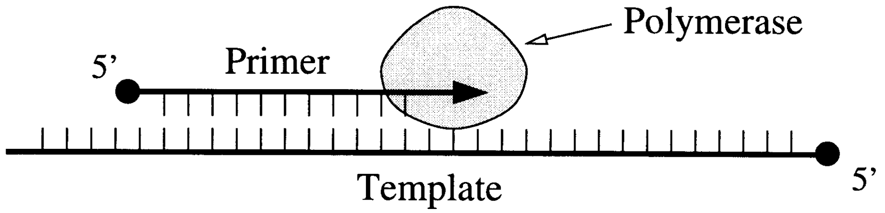

Figure 2.5: *Polymerase catalyzes the extension of the open* $3'$ *end of a strand of DNA along a template.*

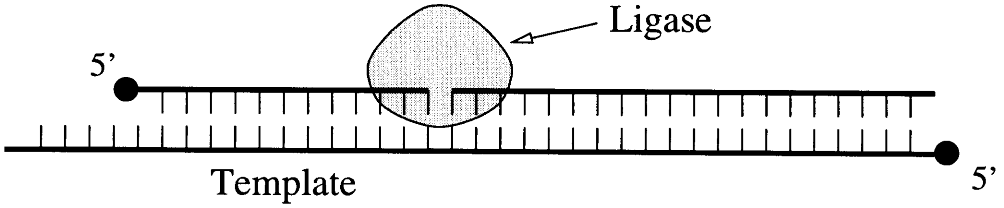

Figure 2.6: *Ligase catalyzes the concatenation of adjacent* $3'$ *and* $5'$ *ends of strands of DNA hybridized to a template.*

a DNA ligase which catalyzes the formation of a phosphodiester bond between the two, effectively concatenating them. Other ligases are capable of ligating two blunt ended pieces of double stranded DNA together but these are less widely used within the field of DNA computing.

The specific enzymes used in the experiments described in this thesis are Vent DNA Polymerase and Taq DNA Ligase. These are thermostable enzymes, isolated from bacteria which live in thermal ocean vents. As a result, they are able to retain their enzymatic activity after exposure to temperatures near boiling and are therefore good candidates for use in reactions requiring high temperature thermocycling like programmed mutagenesis.

Chapter 3

Programmed Mutagenesis

3.1 The Promise of Programmed Mutagenesis

As described in the introductory chapter, programmed mutagenesis is a technique for programmatically rewriting DNA sequences by incorporating highly sequence-specific oligonucleotides into newly manufactured strands of DNA. A programmed mutagenic machine can be represented as a set of states $S = \{\sigma_i\}$ and a set of rewrite rules $R = \{\rho_i\}$, along with a transition relation $T : S \times R \mapsto S$ and a set of initial states $S_0 \subseteq S$. If only a single template strand is present initially, S_0 is a singleton; if the action of a single rewrite rule is deterministic, T is a function. For the computation itself to be deterministic, it must also be the case that for each state $\sigma_i \in S$, there exists only one element of the domain of T which is of the form (σ_i, ρ_j).

The salient point regarding programmed mutagenesis is that it relies on the hybridization and enzyme specificity of its rewrite rules to ensure that the strand of DNA is being rewritten in a systematic way. For example, if rewrite rule ρ_i is meant to be applied to a strand of DNA representing state σ_i, producing a strand representing state σ_{i+1}, and rewrite rule ρ_{i+1} is subsequently meant to be applied to the strand of DNA representing state σ_{i+1} to produce a strand representing σ_{i+2}, it should be the case that ρ_{i+1} cannot be applied to σ_i and ρ_i cannot be applied to σ_{i+1}. If this condition is satisfiable, then both of the rewrite rules can be present in the reaction and yet the system can only evolve from the state representing σ_i to the state representing σ_{i+2} by first passing through σ_{i+1}. Each rewrite rule is applied in series, thereby capturing the notion of programmatic computation.

Since programmed mutagenesis differs quite substantially from previously proposed models of DNA-based computation, it might be instructive to elaborate on the several reasons why this paradigm holds such great promise.

- The pool of oligonucleotides can be designed to cause sequence-specific programmed changes to occur, including the propagation of programmed changes up and down a DNA molecule and the evolution of a programmed sequence of changes over the course of future replication events. Thus, serial computations with programmatically evolving state can be carried out, resulting in *constructive computation*, as contrasted with *selective computation* which requires all possible solutions to a problem to be present *ab initio.*

- The sequence specificity of the oligonucleotides permits a set of oligonucleotides to be present at each step of the reaction, with only a fraction of them being active during each cycle. This reduces human effort since it allows the computation to be carried forward by thermocycling the reactants in the presence of thermostable polymerase and ligase. Ideally, there would be no need for human (or robotic) intervention between computation steps.

- Although input and output of data from programmed mutagenesis reactions can be accomplished by the direct synthesis of input molecules and sequencing of output molecules, other more indirect methods exist which lend this technique greater flexibility and modularity. For example, input molecules could be created by ligating computational DNA subunits together, with the subunits chosen to encode specific problems.

- Programmed mutagenesis can be used in conjunction with other proposed computational systems by creating output strands to be used as input to other systems. In this manner, input strands for other systems could be generated in a systematic fashion. Because of this "intelligent design" property, the need to have all possible solutions present at the start of a selective computation may be obviated.

- All the components necessary to implement programmed mutagenesis are present *in vivo*. Therefore it may eventually be possible to harness the internal workings of the cell for computation, thereby capitalizing on the cell's homeostatic capabilities to ensure that the computation takes place in a stable chemical environment.

In addition to these strengths, programmed mutagenesis has the distinct advantage of being tested in the laboratory. Preliminary programmed mutagenic machines have been constructed by D. Gifford and J. Khodor and methods for optimizing machine performance continue to be investigated. The following section describes how the technique of programmed mutagenesis has been applied to build a simple unary counter.

3.2 The Original Unary Counter Machine

In this section, the design of the original unary counter machine is presented. After the SCAN program was implemented, new unary counter machines were developed which are similar to the original machine but incorporate a few substantive changes. SCAN and the new unary counter machines are discussed in detail later in chapter 6.

To demonstrate the function of programmed mutagenesis, a unary counter was implemented with the hope that it would be incremented every time the DNA was replicated. The value of the counter is encoded in a strand of DNA as the number of **X** and **Y** symbols it contains, where **X**, **Y**, and **Z** symbols are shorthand representations of specific 12-nucleotide sequences, designed so that both **X** and **Y** differ from **Z** at two base positions, but from each other at four. The initial template strand contains the sequence **ZZZZZX**[1], which encodes the number one. The mutagenic oligonucleotides which implement the counting mechanism are **X′Y′** and **XY**, denoted ρ_1 and ρ_2, respectively (see figure 3.1).

[1] Recall that strands of DNA are always written in the $5' \rightarrow 3'$ direction.

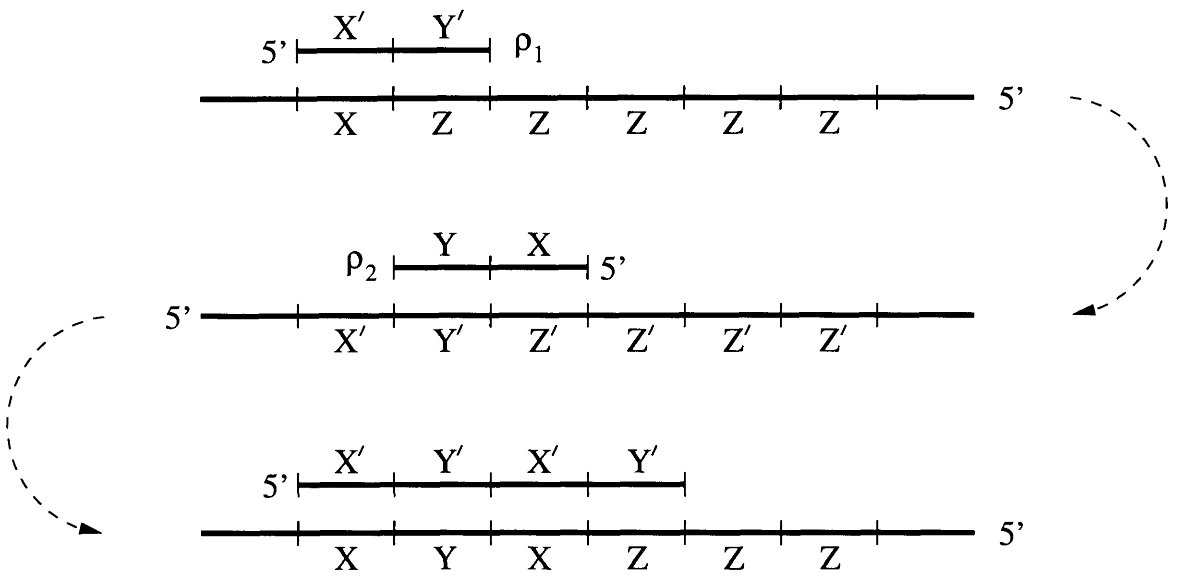

Figure 3.1: *The original unary counter machine: In each cycle, a mutagenic primer hybridizes with the current template and is incorporated into a new template strand.*

This counter construction presumes that 24-mer oligonucleotides which bind to the template with two mismatches can be extended and successfully ligated to other polynucleotides, while oligonucleotides with a greater number of mismatches cannot be effectively incorporated into the new strand. Note that the number of oligonucleotides necessary for counting is fewer than the number of counter values; in this case, two oligonucleotides are sufficient to allow the counter to advance to an arbitrary value.

Each cycle, the counter uses as its input template the product of the immediately preceding cycle, and the product of cycle N encodes the number $N+1$. The counter is implemented by thermocycling a reaction that begins with the initial template strand, ρ_1, ρ_2, thermostable polymerase and ligase, and outside primers LP and RP (needed to produce the full-length product on each cycle). Each thermal cycle consists of a high temperature step, which denatures the double stranded DNA and prepares it for the polymerization-ligation step; and a low temperature step, which is permissive for primer hybridization, polymerization, and ligation. Ideal values for these temperatures were chosen on the basis of data from laboratory experiments. In chapter 5, the BIND simulator is used to demonstrate how such choices might be made more easily, circumventing the need for detailed laboratory experiments.

As this description of the unary counter reveals, successful operation of the machine requires the mutagenic oligonucleotides to be incorporated in the correct place and at the correct time. If oligonucleotides hybridize at alternate binding sites or if "inactive" rules hybridize during the wrong cycle, undesired computations will be carried out.

This constraint is a serious one in the unary counter example, but the ability of mutagenic oligonucleotides to bind at multiple sites along a template may provide a straightforward way of implementing constructive non-deterministic computation, as contrasted with the selective forms of non-deterministic computation being pursued elsewhere. In selective non-deterministic computation, non-determinism is only exploited during the formation of the pool of possible solutions which are later deterministically filtered. Contrastingly, in the constructive version of non-deterministic computation, non-determinism would allow the machine to explore different branches of a computation non-deterministically. Of course, the degree of non-determinism remains limited by the amount of physical DNA present in a reaction, but since an extraordinary amount of DNA can be present in even a small volume of solution, the ability to explore these branches in a massively parallel fashion *may* provide DNA-based computation a comparative advantage over silicon-based computation.

Chapter 4

The BIND Simulator: Development

Since all models of DNA-based computation depend on assumptions about hybridization specificity for their proper operation, programmed mutagenesis in particular, a thermodynamic hybridization simulator would be of great utility. As described in section 2.2, it is possible to predict the melting temperature of deoxyribonucleic acid duplexes in a relatively straightforward manner. With this in mind, a program named BIND was implemented to assist in the prediction of the melting temperatures of DNA duplexes. The user provides the ionic and reactant concentrations and also the particular sequences of the strands under consideration, one serving as a template and the remainder serving as primers which can hybridize with the template. BIND then calculates, for each primer, the melting temperature of the nascent primer/template duplex at each potential binding site along the template. It reports the theoretical melting temperature of all such nascent duplexes for each primer the user provides. The resulting data can be used to generate plots revealing the most stable binding sites along the template for each of the primers.

Note that if the template is the same length as the primer, BIND just calculates the melting temperature between the two strands at a single position, the position where the two align perfectly. Therefore, calculation of the melting temperature of a simple double stranded segment of DNA is easily handled. Consequently, BIND's melting temperature calculation can be calibrated by comparing its output with data reported in the literature for a number of oligonucleotide duplexes.

The initial version of BIND was written in C and is an extremely simplified simulator. While the thermodynamic prediction engine is more sophisticated than that of other simulators currently available, the user interface is somewhat primitive. The program is command-line driven and reads the input DNA sequences from user specified text files. Program output is directed to a file and can also be displayed on screen. A newer version of BIND is currently in production. It is being written in C++ and will have a much improved user interface. A more detailed description of this future work is presented in chapter 7.

4.1 Calculation of Theoretical Melting Temperature

Theoretical melting temperature is typically calculated assuming that the coil-helix transition is two-state, which is a justifiable assumption for small oligonucleotides.[1] SantaLucia, *et al.* [16] suggest that the two-state model is capable of providing a reasonable approximation of melting temperature for duplexes with non-two-state transitions, but the applicability of the assumption obviously decreases as the size of the duplex under consideration increases.

For a two-state model of the

$$\text{single-strand} + \text{single-strand} \rightleftharpoons \text{double-strand}$$

transition between two distinct oligonucleotides in equimolar concentration, the equilibrium constant is given by

$$K = \frac{2f}{(1-f)^2[C_T]} \tag{4.1}$$

where f is the fraction of strands in the double-stranded state and $[C_T]$ is the total molar strand concentration.

[1] Usually defined in this context as fewer than 15-25 bases.

But the equilibrium constant can also be expressed in thermodynamic terms as

$$K = \exp(-\frac{\Delta G^{\circ}}{RT}) = \exp(-\frac{\Delta H^{\circ} - T\Delta S^{\circ}}{RT}) \tag{4.2}$$

where R is Boltzmann's constant and T is the temperature in Kelvin. Since the melting temperature, T_m, is defined as the temperature at which half of the strands are in the double stranded state, it follows that $f = 1/2$ when $T = T_m$. Setting $f = 1/2$ and setting the two expressions for K in equations 4.1 and 4.2 equal to one another, we get an equation relating the melting temperature, T_m, to the total molar strand concentration and the enthalpy and entropy of the forward state transition:

$$T_m = \frac{\Delta H^{\circ}}{\Delta S^{\circ} + R\ln([C_T]/4)} \tag{4.3}$$

In order to adjust for [Na$^+$] concentrations different from 1M, the salt adjustment term[2] introduced by Wetmur [19] is appended to the right hand side of equation 4.3, yielding equation 4.4:

$$T_m = \frac{\Delta H^{\circ}}{\Delta S^{\circ} + R\ln([C_T]/4)} + 16.6\log_{10}\left(\frac{[\text{Na}^+]}{1 + 0.7[\text{Na}^+]}\right) - 269.3 \tag{4.4}$$

The nearest neighbor stacking model allows the total enthalpy and entropy of the transition to be expressed as

$$\Delta H^{\circ} = \Delta H^{\circ}_{ends} + \Delta H^{\circ}_{init} + \sum_{k\in\{stacks\}} \Delta H^{\circ}_{k} \tag{4.5}$$

$$\Delta S^{\circ} = \Delta S^{\circ}_{ends} + \Delta S^{\circ}_{init} + \sum_{k\in\{stacks\}} \Delta S^{\circ}_{k} \tag{4.6}$$

[2] Note that when [Na$^+$] is 1M, the term drops out.

Equations 4.4, 4.5, and 4.6 are similar to the ones reported in Wetmur [19], but they make use of the entropy, ΔS°, rather than the free energy, ΔG°, which is temperature dependent. Since temperature is the variable of interest in this calculation, the expression for melting temperature which does not include free energy terms is preferred.

If divalent cations like Mg^{++} are present in solution, their effect on duplex stability can be expressed in terms of an $[\mathrm{Na}^+]$ equivalent using the conversion [20]:

$$[\mathrm{Na}^+]_{equiv} = 4\sqrt{[\mathrm{Mg}^{++}]} \tag{4.7}$$

For self-complementary oligonucleotides, the $[C_T]/4$ term is replaced by $[C_T]/2$: the equilibrium constant K changes by a factor of 4, but because of symmetric entropic considerations, another factor of 2 must be introduced in the other direction, partially offsetting the first factor [8, 20]. If the strands are not in equimolar concentration, but one strand is present in gross excess over the other, the $[C_T]/4$ term becomes $[C_T]$. Intermediate cases can also be handled by modifying this term appropriately.

The thermodynamic parameters used in calculating the enthalpy and entropy values in equations 4.5 and 4.6 are taken from the literature. BIND's modular design enables it to use multiple thermodynamic basis sets, so it was tested on two different basis sets as discussed in the following section. The enthalpy and entropy values for nearest neighbor stacks were taken first from Quartin and Wetmur [13] and then from SantaLucia, Allawi, and Seneviratne [16]. The former paper uses enthalpy values previously determined by Breslauer, *et al.* [6] but provides new entropy values.

Neither of these basis sets accounts for mismatched base pairs, so mismatches were handled by treating them as virtual stacks, as described by Werntges, *et al.* [18]. The simulator uses the thermodynamic parameters associated with virtual stacks reported in their paper, but a correction is introduced to compensate for the specific nucleotide context in which those mismatches occurred. Since it is possible for multiple mismatched base pairs to occur consecutively and produce a large internal loop, BIND treats such a loop as a single large virtual stack, adding the average of all the mismatch enthalpies and entropies for the bases in the loop, rather than adding the mismatch enthalpy and entropy for each.

4.2 Calibration of BIND

In order to calibrate the BIND simulator, the melting temperatures for ninety-three oligonucleotide sequences were calculated and compared with experimental melting temperatures reported in the literature. BIND computed the melting temperatures for all ninety-three sequences using each of the two thermodynamic parameter basis sets, with the primary intention of ensuring BIND's accuracy and with the secondary intention of determining which basis set's parameters yielded better predictions of oligonucleotide duplex melting temperature.

The calibration process was relatively *ad hoc* in that it was only used to verify that the parameter basis sets were reasonable predictors of melting temperature for a vast range of oligonucleotides. The results are not a statistically accurate assessment of the relative performance of the two basis sets. For example, there is a dependence issue arising from the fact that forty-four of the oligonucleotide sequences tested were also used by SantaLucia, *et al.* [16] in the determination of their thermodynamic parameters so one would expect their basis set to perform quite well on those sequences. Similarly, the Quartin and Wetmur [13] thermodynamic parameter basis set was calculated using some of the other test sequences, again invalidating independence assumptions. Consequently, these results should not be misinterpreted as conclusive proof that one basis set is superior to the other.

Thirty-two mismatched sequences from work by Aboul-ela, *et al.* [1] and Werntges, *et al.* [18] were also tested to verify that the simulator predicts melting temperatures for sequences with mismatched base pairs reasonably well. Additionally, a few sequences from Quartin and Wetmur [13] were used to test the simulator's ability to handle sequences with dangling ends. Finally, a single sequence from work by Braunlin and Bloomfield [5] was tested at five different salt concentrations to verify that the melting temperature is being

Table 4.1: *Results of* BIND *calibration tests*

		Quartin & Wetmur		SantaLucia		Weighted 3:8	
Test	Number	*av* ΔT_m	*av* $\lvert\Delta T_m\rvert$	*av* ΔT_m	*av* $\lvert\Delta T_m\rvert$	*av* ΔT_m	*av* $\lvert\Delta T_m\rvert$
Quartin & Wetmur	12	-1.0	2.1	4.2	4.2	2.7	2.8
SantaLucia	44	-2.5	4.5	0.3	1.8	-0.5	1.9
Aboul-ela Mismatches	16	-1.4	3.3	-2.1	3.4	-1.9	3.3
Werntges Mismatches	16	1.6	1.8	1.8	1.9	1.8	1.8
Salt Concentration	5	-5.2	5.2	0.5	0.9	-1.1	1.1
Total	93	-1.6	3.5	0.6	2.3	0.0	2.2

av ΔT_m indicates the average of the difference between the calculated T_m and the experimental T_m.

av $|\Delta T_m|$ indicates the average of the absolute value of the difference between the calculated T_m and the experimental T_m.

predicted correctly as the ionic environment changes.

A summary of the results of these tests is presented in table 4.1. As expected, the Quartin and Wetmur basis set outperformed the SantaLucia basis set on its own sequences and *vice versa.* Both sets of parameters performed equally well on the sequences containing mismatches. On the whole, the average absolute deviation from experimentally determined melting temperature was 3.5° C for the Quartin and Wetmur basis set and 2.3° C for the SantaLucia basis set. The former tended to underpredict T_m by 1.6° C, while the latter tended to overpredict T_m by 0.6° C. When the predicted temperatures from the two basis sets were averaged with weights of 3 and 8 to compensate for the under- and overprediction, the average absolute deviation fell to 2.2° C.

It should be emphasized that no outlying data points were discarded, the test sequences were not culled for their simplicity, a single algorithm is being used for all the sequences, whether they contain mismatches or dangling ends or exist in an environment with low salt concentration, and no changes were made to the algorithm or underlying parameters on the basis of these test sequences. In fact, absolutely no optimization of the algorithm was performed after this analysis. Therefore, it is expected that this performance could be improved somewhat if a new set of thermodynamic parameters were determined (especially parameters describing the thermodynamics of base pair mismatches) or a few adjustments were made to the algorithm in light of these test results. Nevertheless, the fact that the predictions were quite accurate even without optimization lends credence to the claim that BIND is capable of generating accurate melting temperature predictions under a wide range of conditions.

In conclusion, neither basis set is definitively superior to the other, especially in light of the dependence issues mentioned earlier. Nevertheless, because the SantaLucia basis set slightly outperformed the Quartin and Wetmur basis set, melting temperatures for the remainder of this paper will be calculated using the SantaLucia basis set.

4.3 Laboratory Confirmation of BIND Accuracy

In addition to calibrating the simulator using melting temperature data reported in the literature, two laboratory experiments were carried out by J. Khodor and D. Gifford to solidify the conclusion that BIND is useful in providing accurate melting temperature predictions.

In the first experiment, three distinct primers with differing degrees of mismatch were individually mixed with a single template strand at two different temperatures and given time to be extended by polymerase. Lanes 1-3 in the gel shown in figure 4.1 correspond to the three primer reactions run at 45° C, while lanes 4-6 in the gel correspond to the three primer reactions run at 55° C. The gel clearly indicates that while all three primers are successfully extended at 45° C, only the first and third primer are successfully extended at 55° C. This would suggest that the melting temperature of the primer-template duplex at the optimal binding position[3] along the template for the first and third primers would be above 55° C, but would be between 45° C and 55° C for the second primer. In fact, BIND predicts the melting temperatures to be 60.2° C, 47.7° C, and 65.6° C, respectively.

In the second experiment, two distinct solutions containing a primer and a template were given time to be extended by polymerase at four different temperatures. Both 24-mer primers possessed two mismatches in the optimal binding position along their respective templates. Lanes 1-4 in the gel shown in figure 4.2 correspond to the first reaction being run

[3] Optimal binding position simply means the position with the highest melting temperature.

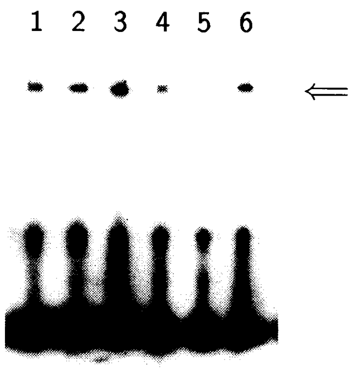

Figure 4.1: *Experiment 45 gel, lanes 1-6: The six lanes are indicated by the numerals at the top, while the band of interest is indicated by the arrow at right. Descriptions of the reactions in each lane appear in the text.*

at 45° C, 50° C, 55° C, and 60° C, respectively. Lanes 5-8 correspond to the second reaction being run at the same four temperatures. The gel indicates that while the first reaction allows polymerization at all four temperatures, the second reaction seems to falter between 55° C and 60° C. Indeed, BIND predicts the melting temperature of the primer/template duplex to be 60.4° C for the first reaction and 57.1° C for the second, in close agreement with the conclusions drawn in the laboratory.

BIND's ability to find the optimal binding location along the template strand is further evidenced by the fact that in each of these experiments involving mismatched oligonucleotides, the optimal binding position reported by BIND corresponded precisely with the optimal binding position as determined by the length of the polymerase-extended product appearing in the gels. Thus it seems likely that BIND is successful in locating optimal primer binding position along template strands, even in the presence of base pair mismatch.

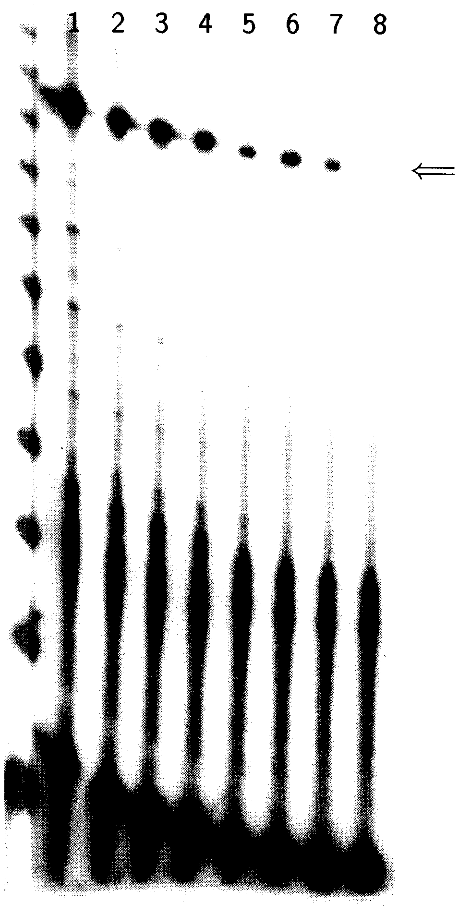

Figure 4.2: *Experiment 74 gel, lanes 1-8: The eight lanes are indicated by the numerals at the top, while the band of interest is indicated by the arrow at right. Descriptions of the reactions in each lane appear in the text. This gel is slightly skewed, with the bands sloping gradually downwards moving from left to right.*

Chapter 5

The BIND Simulator: Application

Now that the design and testing of the BIND simulator have been presented, some examples of how BIND can be used in designing DNA-based computational systems are discussed, with reference to hybridization specificity requirements. In the first two sections of this chapter, a few steps in the design of a simple unary counter are considered. Throughout these sections, the template under consideration consists of 224 nucleotides and contains within it the sequence **ZZZZZX**, as mentioned before. This sequence will henceforth be referred to as the *active site* of the template. Also as before, the first cycle rewrite rule, denoted ρ_1, is $\mathbf{X'Y'}$ and the second cycle rewrite rule, denoted ρ_2, is $\mathbf{XY}$. In the third and final section, we demonstrate how BIND can be used in other DNA computing contexts by considering an application of the simulator within the proposed sticker model [14].

5.1 Primer Specificity

As an example of BIND's usefulness, consider the choice of an outside primer for the template used in the unary counter machine. An appropriate site away from the template's active site is selected and then the outside primer is chosen to bind perfectly to this site. For this outside primer to be safe to use, however, it must be the case that it does not interfere with any of the other reactions which may be taking place in the test tube simultaneously.

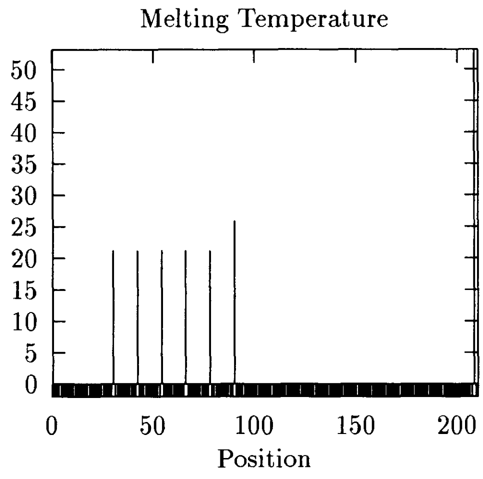

Figure 5.1: *Primer binding: Melting temperature as a function of position for an outside primer with respect to the cycle 1 template. The primer should bind specifically at the end of the template but not interfere anywhere else.*

In particular, the primer should not be able to bind to the active site of the template.

Once chosen, the template and a candidate primer are passed to BIND as input, whereupon BIND returns the information displayed in figure 5.1. In the rightmost position (corresponding to the 3′ end of the template), there is a perfect match between the template and the selected 17-mer primer, yielding a melting temperature around 53° C. The primer is unlikely to hybridize inappropriately for most of the length of the template strand, but some potential interference can be detected near the active site for temperatures in the 20-25° C range. Although this is unlikely to cause a problem for high-temperature reaction cycles, BIND has indicated a possible source of interference between the outside primer and the active site, leading a machine designer to ensure that temperatures remain well above 25° C. It would be even better for the machine designer to select a different outside primer, one exhibiting less active site interference.

5.2 Rewrite Rule Incorporation and Serial Computation

Programmed mutagenesis relies on the sequence specificity of its rewrite rules in order to guarantee that certain steps are being performed before others. Recall that if rewrite

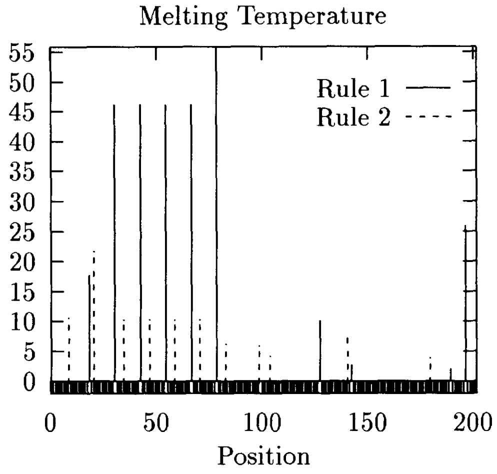

Figure 5.2: *Cycle 1 binding: Melting temperature as a function of position for rewrite rules 1 and 2 with respect to the cycle 1 template. In this cycle, rule 1 should bind specifically while rule 2 remains uninvolved.*

rule ρ_i is meant to be applied to a strand of DNA representing state σ_i, producing a strand representing state σ_{i+1}, and rewrite rule ρ_{i+1} is subsequently meant to be applied to the strand of DNA representing state σ_{i+1} to produce a strand representing state σ_{i+2}, it should be the case that ρ_{i+1} cannot be applied to σ_i and ρ_i cannot be applied to σ_{i+1}. In the context of the unary counter, BIND can be used to verify that ρ_1 can bind to the template in the first cycle but not in the second, while ρ_2 can bind in the second cycle but not in the first. Additionally, BIND can easily determine a temperature which would guarantee this hybridization specificity condition is met.

To test this, the same template as above can be passed to BIND, along with both rewrite rules, ρ_1 and ρ_2. The results for this first cycle reaction are shown in figure 5.2. Then the template which is the product of the first cycle can be passed to BIND, along with both rewrite rules once again. Figure 5.3 displays the results for this second cycle reaction. As the figures indicate, the correct rewrite rule binds to the template in the correct cycle, with very little interference from the opposing rewrite rule. As in the case of the outside primer, if the temperature of the reaction is maintained well above 25° C, no cross-rule interference should be observed.

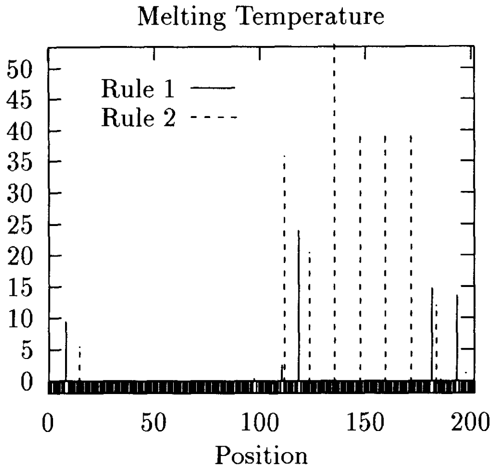

Figure 5.3: *Cycle 2 binding: Melting temperature as a function of position for rewrite rules 1 and 2 with respect to the cycle 2 template. In this cycle, rule 2 should bind specifically while rule 1 remains uninvolved.*

However, these two figures reveal that it is possible for the rewrite rules themselves to inappropriately bind to the template elsewhere in the active site, annealing to any **ZZ** subsequence in the template. In order to prevent this, reactions should be run somewhere in the 47-52° C range. In fact, these same observations were made in the laboratory before BIND was developed.

5.3 Sticker Specificity

To provide a sample application of BIND in a domain unrelated to programmed mutagenesis, an experiment was conducted to measure the likelihood of sticker interference in the sticker model proposed by Roweis, *et al.* [14]. A template strand consisting of 10,000 nucleotides was generated at random. Then five stickers were chosen, of lengths 10, 15, 20, 25, and 30, respectively. Each sticker was guaranteed to match the template in at least one location because the stickers were chosen to be complementary to a subsection of the original template.

The 10,000-mer template was passed to BIND along with each sticker. The sticker of length 10 bound to its complementary subsection at a temperature of 53° C, at another location at 46° C, and at a third at 34° C. No other locations had melting temperatures above 34° C. For stickers of length 15 and 20, no "false" binding position was found with a melting temperature within 30° C of that of the optimal binding location, while for stickers of length 25 and 30, no "false" binding position was found with a melting temperature within 40° C of that of the optimal binding location. Of course, one strand constructed haphazardly does not a proof of specificity make, but it is reassuring that BIND is able to confirm that the randomly chosen stickers of length at least 15 hybridize with the template strand in this example in a highly specific fashion.

This simple-minded demonstration is intended to illustrate how BIND can be used in other DNA computation contexts, such as the verification of sticker choices. Although sticker interference is unlikely at sufficiently long lengths, BIND can be used to conclusively rule out such interference or point out where such interference might be occurring. It also provides valuable information about the temperature at which sticker operations should be carried out in order to maximize binding specificity.

For this reason, the simulator seems to hold a great deal of promise, not only in the context of programmed mutagenesis but also in the contexts of other proposed models of computation. In fact, it should prove valuable to any research effort involving site-specific hybridization, whether in biological computation or molecular biology at large.

Chapter 6

Using SCAN to Search for Optimally Designed Machines

In its initial incarnation, BIND proved useful in analyzing designs for a unary counter based on the technique of programmed mutagenesis. Sequences representing the template and the mutagenic primers were fed to BIND and it reported the theoretical melting temperatures of the nascent primer/template duplexes at each possible binding site along the entire length of the template. By observing the relative stability of each of these nascent duplexes, the likelihood of the mutagenic oligonucleotide primers binding in specific positions and being incorporated into subsequent template strands was able to be determined. While the primers were engineered to be incorporated in specific sites, BIND was able to verify this conclusively and more importantly, give a quantitative assessment of the likelihood of primers being falsely incorporated into a new template.

Thus if the DNA sequences under consideration are known, it is simple to use BIND to determine their hybridization specificity. The inverse problem, however, is significantly more difficult: if a number of hybridization specificity criteria are given, can DNA sequences be found to satisfy them?

Because some of the results from the original unary counter machine design were not as clean as hoped, it became clear that designing a new programmed mutagenic unary counter machine would be a productive exercise. BIND was an integral part of this design process because it allowed for designs to be evaluated without extensive lab work, transforming two weeks in the lab and hundreds of dollars worth of reagents into fifteen minutes on a computer, most of which consisted of entering the sequences to be analyzed. Nevertheless, selecting sequences and then testing them one by one with BIND quickly became tedious. For this reason, the SCAN program was developed.

The next section describes the design of the new unary counter machines and describes the context in which SCAN was developed. SCAN itself is discussed in the section thereafter.

6.1 The New Unary Counter Machines

6.1.1 Design Modifications

While experiments investigating the original unary counter machine yielded positive results, the results were not clean enough to demonstrate conclusively that the hypothesized behavior was being observed in practice. The original design contained a few suboptimal elements, so the decision to produce a new machine provided the opportunity to make some modifications to the original design.

First, the machine's direction of operation was reversed. In the new machine, the **X** in the initial template strand is at the $5'$ end of the strand whereas in the original machine, the **X** in the initial template strand is at the $3'$ end. As a result, all the ligation and polymerization operations are in the opposite direction for any given cycle of the counter.

Second, the lengths of the oligonucleotide primers were reduced from 24 bases to 21 bases by simply truncating each primer after 21 bases. If a primer hybridizes with the template, the last three bases will be replaced by the action of polymerase during the replication step, but shortening the rules significantly reduces the likelihood of cross-rule $3'$ dimers forming between primers in solution.

The new unary counter machines are constructed from three 12-mer nucleotide sequences, labelled by the symbols **X**, **Y**, and **Z**, as before. In each case, the template

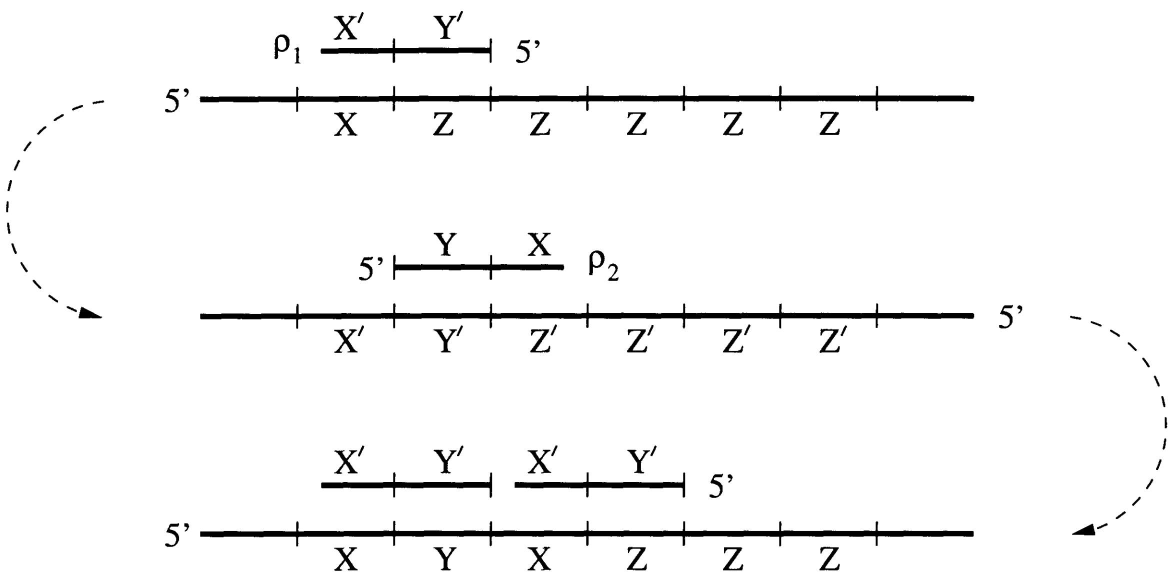

Figure 6.1: *The new unary counter machines: In each cycle, a mutagenic primer hybridizes with the current template and is incorporated into a new template strand. The machines based on this design use shorter 21-mer primers and operate in the opposite direction relative to the original machine.*

consists of a sequence of **X**'s and **Y**'s followed by a sequence of **Z**'s, also as before, modulo orientation. The number of **X**'s and **Y**'s present in the template at any given time encodes the number to which the machine has successfully counted. As shown in figure 6.1, the initial template consists of a sequence of six 12-mers, designated by the symbol sequence **XZZZZZ**, and encodes the number one. During each counter cycle, the first **Z** in the sequence is replaced by either an **X** or a **Y**, thereby increasing the value stored in the counter.

In order to transform **Z**'s into **X**'s and **Y**'s, mutagenic oligonucleotide rewrite rules ρ_1 and ρ_2 are employed. Each rule binds to the template at a temperature permissive for non-specific binding, thereby allowing some nucleotides to be altered and an **X** or **Y** to be written in place of a **Z**.

Rewrite rule 1 consists of the 12-mer $\mathbf{Y}'$ followed by the first nine bases of $\mathbf{X}'$ while rewrite rule 2 consists of the 12-mer **Y** followed by the first nine bases of **X**. In the original unary counter machine, the rewrite rules contained all twelve bases of $\mathbf{X}'$ and **X**, respectively, but here they were shortened to reduce the likelihood of cross-rule 3′ dimer formation, as previously mentioned.

In the first cycle, rule 1 is designed to hybridize with the template so that the nine bases of $\mathbf{X}'$ bind to the **X** in the template and the $\mathbf{Y}'$ binds to the **Z** following the **X** (as shown in the figure). If this hybridized mutagenic oligonucleotide is successfully incorporated into full-length product after the action of thermostable ligase and polymerase, then the newly produced template strand will consist of four copies of $\mathbf{Z}'$ followed by a $\mathbf{Y}'$ and then an $\mathbf{X}'$ [1].

In the second cycle, rule 2 is designed to hybridize with the new template so that **Y** binds to the $\mathbf{Y}'$ in the template and the nine bases of **X** bind to the $\mathbf{Z}'$ preceding the $\mathbf{Y}'$. If this hybridized mutagenic oligonucleotide is successfully incorporated into full-length product after the action of thermostable ligase and polymerase, then the newly produced template strand will consist of the six 12-mers represented by the symbol sequence **XYXZZZ** and the machine will have thus counted to three.

6.1.2 Mismatch Geometry Selection

Once this schematic design was decided upon, the selection of the various matched and mismatched bases comprising the 12-mers **X**, **Y**, and **Z** began. Selecting nucleotide sequences for these new unary counters consisted of a number of stages. The first was to determine the mismatch geometry of the mutagenic primers. Since the precise locations of the various mismatches have a large effect on the operation of the thermostable polymerase and ligase, the mismatch geometries of the primers are configured to allow successful polymerization and ligation of correctly bound rewrite rules, as well as strict enzyme specificity for incorrectly bound rewrite rules.

The mutagenic primers are able to modify the native template sequence by being incorporated into a complementary strand even though they are not perfectly matched with the template. Because the primers need to possess mismatches relative to the desired binding

[1] Recall that the orientation of the newly formed complementary strand is opposite that of the template.

site in the template, they are inherently less stable than perfectly matched primers. In order to be successfully incorporated, they need to be ligated and extended with thermostable ligase and polymerase, respectively. However, these enzymes require that the primer bind to the template with sufficient stability for the appropriate enzyme complex to be formed. This means that the mismatches within the mutagenic primers which are intended to be incorporated should not be too close to either the $5'$ ligation end of the primer or the $3'$ polymerization end of the primer. These mismatches do provide some specificity leverage in that it is possible to design the counter such that mutagenic primers which are *not* intended to be incorporated have mismatches at either their $5'$ or $3'$ ends to prevent the formation of full-length product.

To test various mismatch geometries, a number of laboratory experiments were performed in which mutagenic primers with different mismatch geometries were placed in test tubes with a template sequence and thermostable ligase and polymerase. Each geometry was evaluated in terms of its ability to be successfully polymerized and ligated. Based on this series of experiments, a few geometries were selected for consideration in the next selection stage.

6.1.3 Base Pair Selection

The mismatch geometries are merely skeletons describing the positions where Watson-Crick pairings should occur and where mismatches should occur. They do not indicate which nucleotides should be used in each position. Therefore, specific nucleotides remain to be selected. Previously, this was done by making "intelligent guesses". Such designs seem to have been successful but they have not produced results which are clean enough to verify that the expected behavior is in fact being observed. Hence the need for SCAN.

6.2 The SCAN Program

In order to produce a machine with superior discrimination properties capable of demonstrating that the unary counter was operating as hypothesized, a new program was implemented. This program, called SCAN, generated candidate DNA sequences by selecting specific nucleotides to occupy each position in the skeleton strands, and then simulated the unary counter machine's discrimination properties in that scenario. Five different mismatch geometries were considered and over 2 billion nucleotide combinations were scanned for two of those geometries, while over 1 billion nucleotide combinations were scanned for the other three. In total, over 7 billion different machine design candidates were scanned and screened for suitability.

Unary counter machine candidates were filtered on the basis of a number of distinct criteria, as presented in table 6.1. First, candidates were screened on the basis of their discrimination and non-interference properties. Rules which are "active" in any particular cycle must be incorporated into product strands at the correct site along the template. If it is possible for the rewrite rule to be incorporated in the wrong location, the computation will not be reliable; the machine must be discriminating in its rewrite rule incorporation. In addition to verifying that "active" rules are being incorporated in the correct positions, we must also verify that the "inactive" rules are not interfering with the incorporation process. It is an important feature of programmed mutagenesis that various rewrite rules be present in the system simultaneously. Therefore, it becomes a critical factor in machine design that rules which are not applicable in a particular cycle should not be incorporated into product strands. In other words, it should be the case that "inactive" rules have a low binding affinity over the entire length of the template. Therefore, in any successful machine design, rules must be carefully selected so as to prevent them from being incorporated at the wrong time.

Second, candidate designs were screened to ensure that the constituent strands had minimal secondary structure. Since all the strands are present in the system at once, it is quite

Table 6.1: *Specifications for* SCAN *filter criteria*

Filter Criterion	Specification
Strong Discrimination	
Min Correct Binding T_m	45° C
Min Difference between Correct and Incorrect Binding T_m	20° C
Max Inactive Rule Binding T_m	-5° C
Max Difference between Correct Binding T_m across Cycles	6° C
Minimal Secondary Structure	
Max 3′ Hairpin T_m	35° C
Max 5′ Hairpin T_m	50° C
Max 3′ Self-Dimer T_m	0° C
Max 3′ Cross-Dimer T_m	-5° C
Low Plasmid Interference (pUC19)	
Max Binding T_m along Entire Length of Plasmid	20° C

possible that undesirable side reactions could be taking place unintentionally. For example, a strand may possess some secondary structure which causes it to fold back onto itself and hairpin, thereby preventing it from interacting with other strands as desired. Alternatively, an oligonucleotide can hybridize with a copy of itself or with another oligonucleotide rewrite rule, forming unwanted dimers and thereby allowing undesired side reactions to proceed or preventing necessary reactants from interacting as planned. Therefore, it is critical that the constituent strands of any unary counter design be screened for possible secondary structure and undesirable side reactions.

Third, since the unary counter is embedded in a longer plasmid, it is important that the chosen sequences be compatible with the plasmid in which they will later be inserted. For this reason, the plasmid should be scanned for possible alternate primer binding locations and if any are found, either another plasmid needs to be selected or a different design needs to be considered.

After the three screening stages, there were only a handful of designs remaining: roughly 5 designs for the geometries where 2 billion nucleotide combinations were considered and no more than 1 design for the geometries where 1 billion nucleotide combinations were considered. Complete results are shown in table 6.2.

Table 6.2: *Number of candidates passing through successive* SCAN *filters*

Filter	Geometry 1	Geometry 2	Geometry 3	Geometry 4	Geometry 5
None (Original Pool)	2147483648	2147483648	1073741824	1073741824	1073741824
Strong Discrimination	21326	30728	727	498	1086
Minimal Secondary Structure	135	395	19	29	0
Low Plasmid Interference	5	6	1	1	0

After using BIND to display the binding specificity of the rewrite rules along the template and along the entire length of the plasmid for each of these thirteen designs, two designs were finally selected. The designs are entitled pUC.1.3, which is based upon mismatch geometry 1, and pUC.4.1, which is based upon mismatch geometry 4. Table 6.3 contains a full description of the performance characteristics for both machines.

Once the sequences for **X**, **Y**, and **Z** were chosen, other nucleotide sequences needed to be selected to serve as spacer regions (isolating the template from the remainder of the plasmid), and upstream and downstream PCR primers needed to be picked. Hundreds of sequence possibilities were considered before contexts for the two designs were selected. With this in mind, well over a trillion different design combinations were considered before two were finally selected; clearly, the benefits of using computer simulation to assist in a process such as this are immeasurable.

As a final note, both new machine designs are being inserted in the pUC19 plasmid and the process of cloning and preparation is currently underway. Preliminary experiments will be conducted once this process is complete.

Table 6.3: *Performance characteristics of new unary counter machines*

Characteristic	pUC.1.3	pUC.4.1
Cycle 1 Discrimination		
Correct Binding T_m	45.49	46.86
Incorrect Binding T_m	24.88	26.52
Inactive Rule Binding T_m	<-20.00	<-20.00
Cycle 2 Discrimination		
Correct Binding T_m	47.54	45.08
Incorrect Binding T_m	25.85	24.90
Inactive Rule Binding T_m	-19.14	-15.38
Rule 1 Secondary Structure		
3′ Hairpin T_m	31.37	31.80
5′ Hairpin T_m	38.65	29.27
3′ Self-Dimer T_m	-10.04	-14.63
Rule 2 Secondary Structure		
3′ Hairpin T_m	34.04	25.44
5′ Hairpin T_m	40.44	9.96
3′ Self-Dimer T_m	-4.61	-17.56
Cross-Rule Interference		
3′ Cross-Dimer T_m	<-20.00	<-20.00

Chapter 7

Extensions

7.1 Improving BIND

The BIND simulator is being extended in a number of directions. Currently BIND is being rewritten in C++, in order to provide an improved user interface. The code, which is cleaner and more modular due to the object oriented nature of C++, is also being extended to provide greater functionality to the user. In this new version of BIND, tentatively called BIND2, multiple reactions can be simulated, each in its own "test tube", allowing different reactant and ionic concentrations to be tested simultaneously. The user can also more easily create, load, and save DNA sequences, or even entire test tubes. BIND2 will enable the user to check sequences for secondary structure like hairpin formation and dimer production. Because polymerization plays a large role in some proposed DNA-based computational systems, BIND2 will be able to categorize primer/template duplexes by the stability of their open $3'$ end. That is, BIND2 will verify that a primer which is intended to be extended by polymerase does in fact bind to the template strand with sufficient stability at the primer's $3'$ end to allow for successful polymerization. Finally, BIND2 will benefit from the availability of a new basis set of enthalpy and entropy parameters, recently compiled by SantaLucia, *et al.*, but currently unpublished [15].

Another extension involves explicitly incorporating a time element into the simulator's calculations, in order to be able to handle more complex reaction types where more than one binding site is possible at the prevailing temperature or competition between primers arises. This dynamic version of BIND would also provide information on how long reactions would need to be run in order to achieve certain predetermined levels of binding and/or primer extension.

7.2 Improving SCAN

There are also a number of ways in which SCAN could be improved. Currently SCAN code is modified and recompiled with each machine design change in order to improve running time. While each executable image runs significantly more quickly as a result, the time needed to write the source code for each executable is unnecessarily long. Therefore, it would be extremely worthwhile to implement a simple SCAN translator/compiler which would take a given programmed mutagenic machine design, particular mismatch geometry, and desired performance criteria, and write efficient C code capable of scanning the design space for solutions satisfying all the constraints.

A further improvement to SCAN involves improving the algorithm used to screen candidate solutions. It may be that a better algorithm exploiting dynamic programming or memoization could be developed to improve the program's running time dramatically.

Finally, SCAN should be able to interface easily with BIND, though it remains to be seen whether it would be best to leave the two programs as distinct entities in an integrated package or fold the functionality of SCAN into that of BIND.

7.3 In the Laboratory

As mentioned in an earlier chapter, better thermodynamic enthalpy and entropy values would be extremely beneficial in making a more precise version of BIND. In particular, accurate parameters describing mismatched binding would be of great use. Therefore, experiments attempting to more precisely characterize the thermodynamic effects of mismatched base pairs would be quite helpful. The current data are based on a number of restrictive assumptions regarding mismatch geometry and more general results would enhance BIND's overall predictive power in the face of base pair mismatch. Additionally, more elaborate models of the influence of the ionic and chemical environments on deoxyoligonucleotide hybridization would allow BIND to provide accurate melting temperature predictions over a wider range of possible reactions.

By far the most exciting laboratory work, however, involves testing the two new unary counter machines designed with the aid of SCAN. Due to the strict constraints imposed during the design phase, the machines will hopefully exhibit high signal-to-noise ratios and optimized performance characteristics, enabling us to determine more conclusively that the expected programmed mutagenic behavior is being observed. Successful operation of these new unary counter designs will provide a great impetus for continued development of BIND and SCAN, but more importantly, it will demonstrate the potential of programmed mutagenesis as a paradigm for the implementation of constructive computation using DNA.

Bibliography

[1] Aboul-ela, F., Koh, D. & Tinoco, I. (1985) *Nucleic Acids Res.* **13**, 4811-4824.

[2] Adleman, L. (1994) *Science* **266**, 1021-1024.

[3] Adleman, L. (1995) On Constructing a Molecular Computer. Univ. of Southern California, draft, January 8.

[4] Boneh, D., Dunworth C. & Lipton, R. (1995) Breaking DES Using a Molecular Computer. Princeton CS Tech Report CS-TR-489-95.

[5] Braunlin, W. & Bloomfield, V. (1991) *Biochemistry* **30**, 754-758.

[6] Breslauer, K., Frank, R., Blöcker, H. & Marky, L. (1986) *Proc. Natl. Acad. Sci. USA* **83**, 3746-3750.

[7] Cai, W., Condon, A., Corn, R., Fei, Z., Frutos, T., Glaser, E., Guo, Z., Lagally, M., Liu, Q., Smith. L. & Thiel, A. (1996) The Power of Surface-Based DNA Computation. Univ. of Wisconsin, draft, August 13. Available by anonymous ftp from *corn-info.chem.wisc.edu* as */Papers/powerDNA.ps.*

[8] Cantor, C. & Schimmel, P. (1980) *Biophysical Chemistry, Part III: The Behavior of Biological Macromolecules.* W. H. Freeman, San Francisco.

[9] Doktycz, M., Paner, T., Amaratunga, M. & Benight, A. (1990) *Biopolymers* **30**, 829-845.

[10] Freifelder, D. (1987) *Molecular Biology*, Second Edition. Jones and Bartlett, Boston.

[11] Gifford, D. (1996) Programmed Sequential Mutagenesis. Talk at 2nd Annual DIMACS Workshop on DNA Based Computers, May.

[12] Marmur, J. & Doty, P. (1962) *J. Mol. Biol.* **5**, 109-118.

[13] Quartin, R. & Wetmur, J. (1989) *Biochemistry* **28**, 1040-1047.

[14] Roweis, S., Winfree, E., Burgoyne, R., Chelyapov, N. V., Goodman, M. F., Rothemund, P. W. K. & Adleman, L. (1996) A Sticker Based Model for DNA Computation. *Proceedings of 2nd Annual DIMACS Workshop on DNA Based Computers*, May.

[15] SantaLucia, J. (1997) Personal communication.

[16] SantaLucia, J., Allawi, H. & Seneviratne, P. A. (1996) *Biochemistry* **35**, 3555-3562.

[17] Watson, J., Hopkins, N., Roberts, J., Steitz, J., & Weiner, A. (1987) *Molecular Biology of the Gene*, Fourth Edition. Benjamin/Cummings, Menlo Park.

[18] Werntges, H., Steger, G., Riesner, D. & Fritz, H. (1986) *Nucleic Acids Res.* **14**, 3773-3790.

[19] Wetmur, J. (1991) *Crit. Rev. in Biochem. and Mol. Biol.* **26**, 227-259.

[20] Wetmur, J. (1997) Personal communication.

[21] Zenkova, M. & Karpova, G. (1993) *Uspekhi Khimii* **62**, 414-435.