A DNA B sed Artifici l Immune System for Self-Nonself Discrimin tion

R. Deaton
Electrical Engineering
The University of Memphis
Memphis, TN, 38152
rjdeaton@memphis.edu

M. Garzon
Computer Science
The University of Memphis
Memphis, TN, 38152
garzonm@cc.memphis.edu

J. A. Rose
Electrical Engineering
The University of Memphis
Memphis, TN, 38152

R. C. Murphy
Microbiology
and Molecular Cell Sciences
The University of Memphis
Memphis, TN, 38152

S. E. Stevens, Jr.
Microbiology
and Molecular Cell Sciences
The University of Memphis
Memphis, TN, 38152

D. R. Franceschetti
Physics
The University of Memphis
Memphis, TN, 38152

Abstract

Artificial immune systems attempt to distinguish self from nonself through string matching operations. A detector set of strings is selected by eliminating random strings that match the self strings. DNA based computers have been proposed to solve complex problems (e.g. Traveling Salesman) that defy solution on conventional computers. They are based on (hydrogen bonding based) matchings (called hybridizations) between Watson-Crick complementary pairs, A-T (Adenine-Thymine) or C-G (Cytosine-Guanine). Therefore, a single strand (an oligonucleotide) will bind with other oligonucleotides that match most closely its sequence under the operation of Watson-Crick complementation. In this paper, an algorithm for implementing an artificial immune system for self-nonself discrimination based on DNA is described. This procedure takes advantage of the inherent pattern matching capability of DNA hybridization reactions and the notion of similarity naturally found in DNA hybridization.

Introduction

Biology is now an inspiration for computer science. This is evident in algorithms that mimic neural networks[1], sexual reproduction[2], life itself[3], and the vertebrate immune system[4]. Until recently, however, the material of biological systems, organic molecules and cells, were not used for computation. Adleman[5] changed all that when he solved an NP-complete problem, the Hamiltonian Path, using DNA. In this paper, a biologically-inspired algorithm, an artificial immune system, is proposed for implementation in a computer constructed from biological material, DNA.

The vertebrate immune system is capable of combating a large number of different types of invading pathogenic microorganisms[6]. To accomplish this, the molecular agents of the immune system, T-cells, B-cells, and antibodies, recognize foreign antigens by structural and chemical properties of the binding sites between them. In addition, the immune system must be able to distinguish cells and molecules that belong to its host from foreign material, or self from nonself[7]. The immune system response to foreign material is highly specific, which means that a large number of detectors for antigens is required[6]. Also, the immune system has a memory since it will respond to a specific antigen for the remainder of the host individual's life[6]. More than 10^{16} antigens can be recognized by a mammalian immune system[7].

An artificial immune system is an algorithmic attempt to duplicate the ability of a natural immune system to recognize self from nonself. In [4], an artificial immune system was proposed to detect self-

nonself in a computer. The goal was to use the system for protection from computer viruses and other, unwanted changes in a computer system. In subsequent work[8], more efficient ways of generating the detector set were described.

Adleman[5] introduced a way to solve combinatorial optimization problems with DNA. As implemented by Adleman[5], the fundamental reaction in DNA based computation is a template matching reaction based on hydrogen bonding between Watson-Crick complement base pairs, $\overline{A} \equiv T$ and $\overline{G} \equiv C$[6]. This template matching reaction between two single strands of DNA (oligonucleotides) is called hybridization. In Adleman's original work[5], a Hamiltonian path through a graph was formed through successive hybridizations of oligonucleotides (oligos) which represented vertices and edges in the graph. Subsequent proposals for DNA computation have continued to rely upon the mechanism of hybridization[9, 10, 11, 12].

In this work, the DNA template matching reaction, or hybridization, is proposed to implement the self-nonself detection algorithm of [4].

Self-Nonself Discrimination Algorithm of Forrest *et al.*[4]

For discrimination of self from nonself in a computer, the entities of interest are not molecules or microorganisms, but are strings composed from a finite alphabet. These strings can be bit strings, data strings, or strings of machine instructions. For computer security, self is defined as strings to be protected, and nonself as all other strings. The algorithm is as follows:

1. Detector Set Generation: Strings are generated at random. They are compared to the set of self strings. If a matching condition between the strings is met, then, reject the string. Otherwise, accept the string for the detector repertoire. This step is called censoring and is shown in Figure 1.

2. Monitor Protected Data: The protected strings are periodically compared to the detector repertoire. When detector strings are activated, a change is known to have occurred. The monitoring step is shown in Figure 2.

Several details of the algorithm in [4] are important for what follows. First, the self strings are divide into equal size segments. Second, the self strings are assumed not to change over time. Third, the particular matching rule is that two strings must match in r contiguous places. Most importantly for this paper, the algorithms have reduced the problem of change detection in a computer system to the detecting changes in strings. In the DNA implementation to follow, template matching hybridization reactions between DNA oligos will be used to detect change.

The advantages of the technique are that each copy of the detector set is unique to the system on which it is generated. This is desirable since if protection algorithms are identical on multiple sites, then, breaking the scheme at one site means all sites are compromised. In addition, the discrimination is probabilistic. Therefore, for a greater risk of intrusion at one site, high system wide protection is achieved. Third, the algorithm detects any intrusive activity, rather than looking for specific intruders. The chief disadvantage is the computational cost of exhaustively generating the detector set. As mentioned earlier, this has been addressed in [8].

DNA Computation

In Adleman's approach, the instances of a problem are encoded in oligonucleotides of DNA[6]. The encoding alphabet is the set of nucleic acid bases A, T, G, C, which bind according to the Watson-Crick complement condition, $\overline{A} \equiv T$ and $\overline{G} \equiv C$, and *vice versa*. Oligonucleotides bind in an antiparallel way with respect to the chemically distinct ends, $5'$ and $3'$, of the DNA molecule. The enzyme ligase seals the double-stranded DNA in a process called ligation. Polymerase chain reaction (PCR) is used to extract the answer from the length of the path, and from knowledge of the beginning and ending vertices.

A $5'$ to $3'$ oligonucleotide is denoted as $|O_i >$, and a $3'$ to $5'$ oligonucleotide as $< O_i|$, where O_i is an arbitrary sequence of nucleotide bases. The Watson-Crick complement of an oligonucleotide, $|O_i >$, is $< \overline{O}_i|$. A hybridization reaction between two arbitrary oligonucleotides, $< O_i|$ and $|O_j >$, is denoted by $< O_i|O_j >$, where perfect complement base pair matching occurs if $< O_i| =< \overline{O}_j|$. In Adleman's approach, each vertex is associated with a randomly chosen N-mer, $|O_i >$. Each N-mer, $|O_i >= |p_i q_i >$, is composed of elements from two $\frac{n}{2}$-mers, $|p_i >$ and $|q_i >$, from sets P and Q, respectively. The edges,

$i \rightarrow j$, in the graph are formed from the Watson-Crick complements of $|q_i >$ and $|p_j >$, $< \overline{q}_i|$ and $< \overline{p}_j|$, respectively. The edge from vertex i to vertex j, $< O_{i \rightarrow j}|$, is therefore $< \overline{q}_i \overline{p}_j|$. Through hybridization, a path through the graph is formed by the edge oligonucleotides, $< \overline{q}_i \overline{p}_j|$, splicing together the sequence of vertices, $|p_i q_i >$, $|p_j q_j >, \ldots$. The production of the Hamiltonian path depends on the vertex oligonucleotides hybridizing with the correct edge oligonucleotides (See Figure **??**). Non-Watson-Crick complement base pair and other unwanted hybridizations (see Figure 3) are possible in hybridized strands[13], and could lead to errors, false positives and negatives, in the original approach[14].

DNA Implementation of Self-Nonself Discrimination

The basic idea is to implement the censoring and monitoring algorithms of [4] in DNA with techniques from molecular biology. Since string matching is an important part of [4], the template-matching hybridizations between DNA oligonucleotides would seem a natural mechanism for implementation. Other biotechnologies would have to be used as well. Enzymes[6] have the capability of modifying the contents of a tube of DNA. In gel electrophoresis[6], DNA strands are pulled through agarose gels by an electric field. The strands move at speeds that are inversely proportional to their lengths. Therefore, gel electrophoresis can be used to separate DNA molecules by mass. In addition, with the use of restriction enzymes which cut DNA double strands at specific sequences, gel electrophoresis can be used to sequence DNA molecules. Sequencing is the determination of the base sequence that composes the DNA string. High density DNA arrays or chips[15] are also available for sequencing by hybridization (SBH). The DNA chip has many short, oligonucleotides attached to it. A strand to be sequenced is labeled with a flourescent marker and washed over the surface of the chip where it hybridizes with the attached oligos. By proper design of the chip, the sequence of the target strand can be determined from the pattern of hybridizations on the chip. The patterns are optically detected by the flourescense of the markers on the target sequence.

For the censoring, a random set on oligonucleotides of length n, or n-mers, is generated. The self strings, then, are encoded in DNA n-mers. A self set is then constructed from the Watson-Crick complements of the encoded n-mers. Many copies ($\approx 10^{12}$) of the random set and self set of oligos are mixed together at an elevated temperature. The temperature is lowered which allows hybridization to take place. At this point, the self set n-mers will have hybridized with with their Watson-Crick complements in the random set. Since the self set was composed of the Watson-Crick complements of the self strings, the random n-mers that have hybridized correspond to the self strings. At this point, an enzyme, Exonuclease III, is added to the tube. This enzyme chops up the double-stranded hybridization products into mononucleotides, effectively removing the self strands from the mix. Since these are chemical processes, not all copies of the self strands will have been removed. The process of adding the self set, hybridizing, and exonuclease would have to be repeated to remove all the self strands. At the end of the process, the remaining oligonucleotides (those not chopped up by the double strand exonuclease) represent the detector set. These oligos are then sequenced by gel electrophoresis or by using a DNA chip and sequencing by hybridization. The entire process is shown in Figure 4.

For the monitoring algorithm, after the detector set is sequenced, they are attached to a DNA chip. The Watson-Crick complements of the self or protected strings are labeled with flourescent markers and washed over the chip. If hybridization occurs with the detector set, the change in the self set can be optically detected. The process is shown in Figure 5.

The detector set could be updated by adding additional random oligos, and redoing the censoring process.

Discussion

The DNA implementation of self-nonself discrimination maintains all the advantages of the original algorithm. A unique DNA system could be generated for each system to be monitored for change. The change detection is general. That is, any change in the self set is detected, not just specific patterns of change.

The change detection is probabilistic because this is the nature of the hybridization reaction upon which theaTD0Tc995he the not0Tc5099.30000.de5(ddi

ing rules, and to weight different inputs. The hybridization between two oligos is dependent on the pattern of matching between them, and the reaction conditions. Contiguous matchings are more important than isolated matchings for inducing a binding event[13, 16]. The length of a contiguous match that induces binding is temperature dependent. At higher temperatures, longer stretches of bases have to match between two oligos for them to bind. Therefore, the contiguous matching rule of [4] has a direct physical analog in the DNA implementation, and the number of contiguous matches, r, could be controlled by the reaction temperature. The temperature dependence of the hybridization could also be used to intentionally induce mismatches in either the censoring or monitoring processes. In this way, all detector string that are close to a given self string would be removed from the detector repertoire, and a certain amount of latitude for errors or small changes in self strings is taken into account.

In addition, the mole fractions of the hybridization products are related to the mole fractions of the oligo reactants. The stoichiometric equation for hybridization of two arbitrary oligonucleotides, x_i and x_j, is

$$x_i + x_j \rightleftharpoons x_{ij} \tag{1}$$

where x_{ij} represents the hybridized oligonucleotides. The mole fraction of the hybridization product, $[x_{ij}]$ is related to the mole fractions of the oligo reactants, $[x_i]$, $[x_j]$, by

$$[x_{ij}] = K_{eq}[x_i][x_j], \tag{2}$$

where K_{eq} is the equilibrium constant. In either the censoring or monitoring phase, by adjusting the mole fractions of the input strings, the probability of hybridization products containing that string are affected. Therefore, adjusting oligo mole fractions is a way to weight the importance of either detector or input data strings.

The chief disadvantage of the algorithm of [4] was the cost of detector string generation. By implementing the algorithm in DNA, the massive parallelism of the hybridizations in censoring process do the required string comparisons in an effective manner. The complexity of implementing multiple DNA chips and processes, however, has been substituted for the cost of generating the detector repertoire. The censoring algorithm implemented in DNA is an exhaustive search. Other methods for the design of the original random set of detectors would improve the efficiency of the discrimination scheme.

Conclusion

In this paper, a DNA implementation of an artificial immune system for self-nonself discrimination[4] has been proposed. The DNA implementation uses the template-matching ability of hybridization reactions between DNA oligonucleotides to detect changes in a protected set of strings (self). The DNA implementation uses the massive parallelism of the hybridization reactions and a double-strand exonuclease to select the DNA strands for detection of nonself. A high-density DNA array is used to monitor the set of protected strands and detect changes.

References

[1] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation.* Redwood City, CA: Addison–Wesley Publishing Company, Inc., 1991.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Cambridge, MA: MIT Press, 1992.

[3] C. G. Langton, ed., *Artificial Life*, Santa Fe Institute, Addison–Wesley Publishing Company, Inc., 1989.

[4] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, (Los Alamitos, CA), IEEE, IEEE Computer Society Press, 1994.

[5] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.

[6] J. D. Watson, N. H. Hopkins, J. W. Roberts, J. A. Steitz, and A. M. Weiner, *Molecular Biology of the Gene.* Menlo Park, CA: The Benjamin/Cummings Publishing Co., Inc, fourth ed., 1987.

[7] J. K. Percus, O. E. Percus, and A. S. Perelson, "Predicting the size of the T-cell receptor and antibody combining region from consideration of

efficient self-nonself discrimination," *Proc. Natl. Acad. Sci.*, vol. 90, pp. 1691–1695, 1993.

[8] P. D'haeseleer, S. Forrest, and P. Helman, "An immunological approach to change detection: Algorithms, analysis, and implications," in *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, IEEE, IEEE Computer Society Press, 1996.

[9] R. J. Lipton, "DNA solution of hard computational problems," *Science*, vol. 268, pp. 542–545, 1995.

[10] F. Guarnieri, M. Fliss, and C. Bancroft, "Making DNA add," *Science*, vol. 273, pp. 220–223, 1996.

[11] DIMACS, *Proceedings of the First Annual Meeting on DNA Based Computers*, (Providence, RI), American Mathematical Society, 1996. DIMACS Proc. Series No. 27.

[12] DIMACS, *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers*, (Providence, RI), American Mathematical Society, 1997. DIMACS Proc. Series.

[13] J. Sambrook, E. F. Fritsch, and T. Maniatis, *Molecular Cloning: A Laboratory Manual.* Cold Spring Harbor Laboratory Press, second ed., 1989.

[14] 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 *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [12], pp. 159–171. DIMACS Proc. Series.

[15] M. Chee, R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris, and S. P. A. Fodor, "Acessing genetic information with high-density DNA arrays," *Science*, vol. 274, pp. 610–614, 1996.

[16] K. J. Breslauer, R. Frank, H. Blöcker, and L. A. Marky, "Predicting DNA duplex stability from the base sequence," *Proc. Natl. Acad. Sci.*, vol. 83, pp. 3746–3750, 1986.

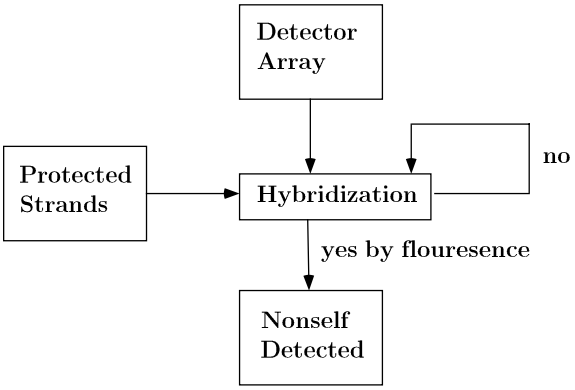

Figure 1: Algorithm for Censoring (after [4]).

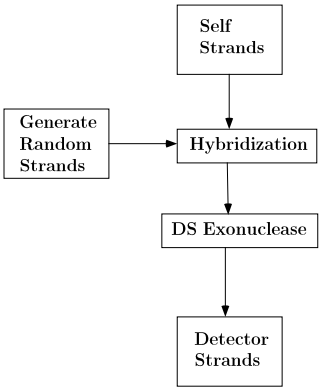

Figure 2: Algorithm for Monitoring (after [4]).

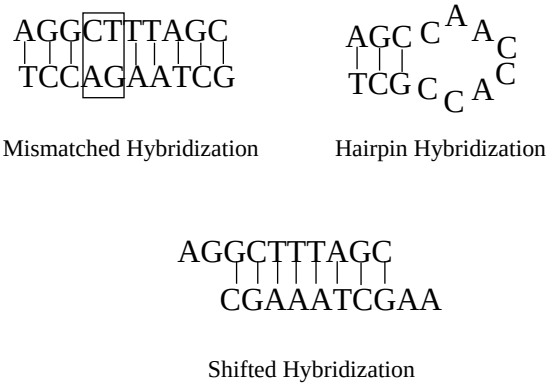

Figure 3: Hybridizations that produce errors and poor efficiency in a DNA computation.

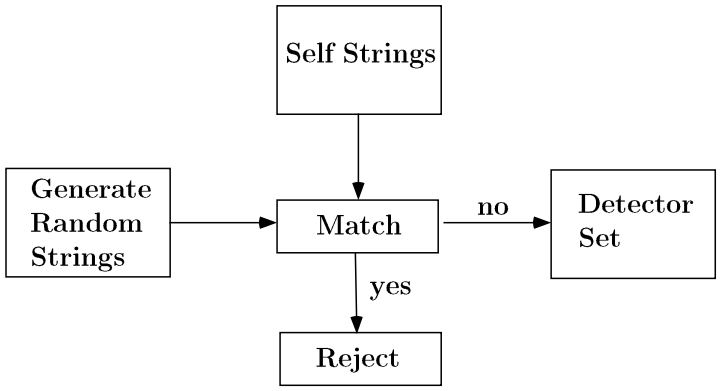

Figure 4: DNA Implementation of Censoring.

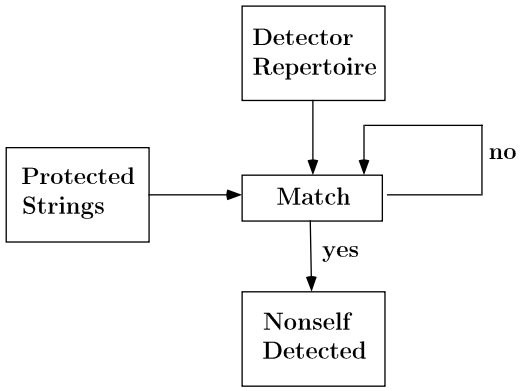

Figure 5: DNA Implementation of Monitoring.