UNIVERSITY OF CALIFORNIA
RIVERSIDE

Experimental Implementation of a Hopfield Neural Network Using DNA Molecules

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Physics

by

Dundar Karabay

March 2011

Dissertation Committee:
Dr. Allen P. Mills Jr., Chairperson
Dr. Roya Zandi
Dr. Ward Beyermann

The Dissertation of Dundar Karabay is approved:

_____

_____

_____

Committee Chairperson

University of California, Riverside

Acknowledgments

This work was funded in part by the U.S. National Science Foundation under grant no. CCF-0524203. The third chapter of this thesis, Feasibility of a Hopfield Neural Network using DNA molecules, is a reprint of the material as will appear in the Journal of Theoretical and Computational Nanoscience. The co-author, Allen P. Mills Jr., listed in that publication directed and supervised the research which forms the basis for this thesis. The other co-author, Dr. Bradley S. Hughes, contributed to the publication by the simulation section.

Dedication

I dedicate this thesis to my wife and son, Ayse G. Karabay and Ahmet A. Karabay, for their understanding throughout my graduate study. I also want to thank my advisor, Dr. Allen P. Mills Jr., for his patience and support.

ABSTRACT OF THE DISSERTATION

Experimental Implementation of a Hopfield Neural Network Using DNA Molecules

by

Dundar Karabay

Doctor of Philosophy, Graduate Program in Physics

University of California, Riverside, March 2011

Dr. Allen P. Mills Jr., Chairperson

According to its mathematical description, a Hopfield Neural Network serves as a content addressable memory with binary threshold units. The elements in that memory consist of the correlations between elements of memory vectors. In this thesis, the feasibility of a Hopfield Neural Network using DNA molecules as the working substance is introduced. In addition, I present an experimental study proving that forming a DNA based memory storing the information of two different 6-bit black and white images, representing memory vectors, and recalling one of original images with the use of a partial image are possible. It is observed that the recalling with a DNA based Hopfield Neural Network using incomplete inputs is more powerful comparing to theoretical one using corrupted inputs. Moreover, as a supplementary work, I show that application of T4

Gene 32 Protein to Isothermal Linear Amplification (ILA) reduces the production of fragment DNA strands, one of the biggest problems of this type of amplification.

Table of Contents

List of Tables

List of Figures

Chapter 1 – General Introduction

1.1. DNA Computing

In 1959, the possibility of building "sub-microscopic" computers was mentioned by Richard Feynman[1]. Although the technology had been improved significantly in years, this goal wasn't able to be achieved until 1994. In that year, Leonard Adleman demonstrated the feasibility of carrying out a computation at the molecular level by using DNA molecules and specific molecular reactions[2]. In his pioneering paper, he encoded a small graph in DNA molecules and solved an example of the directed Hamiltonian path problem (HPP) by using the tools of molecular biology.

In order to understand the innovation of Adleman's study, it is important to know what the HPP means and how complex it can be. This problem is a special case of the traveling salesman problem (TSP), in which a salesman needs to visit a number of cities on a map, all connected by roads. The task is to find the shortest possible route that visits each city exactly once. As the number of cities is increased, the computational complexity of such problem increases much, much faster. This is because there are (n-1)!/2 routes going through n cities. Dividing by two arises from the fact that it does not matter in which direction in time they (cities) come after each other. For instance, there will be 14!/2 (~ 4.36×10^9) different possible itineraries for a problem with 15 cities. If a standard PC with 3GHz processor is used to solve that problem, the required computation time will be ~15 seconds (assuming that the computer carries out 3×10^9 operations per second). The problem including 20 cities will need ~8 months. However, such a problem

with 30 cities cannot be solved by the same type of electronic based computer since this task requires ~4.6×10^{13} years of computation.

Although the HPP is a special type of TSP, as stated above, the purposes of these problems are slightly different. Unlike the TSP, the aim of HPP is not to find the shortest distance but a directed tour that starts at a given city, ends at a given city, and visits every other city exactly once for a given a set of cities and directed paths between them (not all pairs of cities have to be connected with each other).

In Adleman's paper, the cities and roads are represented by vertices (v_i) and edges (e_i), respectively (Figure1.1)[2]. A graph with designated vertices v_{in} and v_{out} is said to have a Hamiltonian path[3] only if there is a path consisted of "one way" edges that begins at v_{in} , ends at v_{out}, and visits every other vertex once and only once. For instance, no Hamiltonian path exist between v_{in}=2 and v_{out}=4 since there are no edges entering vertex 0. However, when v_{in}=0 and v_{out}=6, there is a unique Hamiltonian path: 0 →1, 1→2, 2→3, 3→4, 4→5, and 5→6.

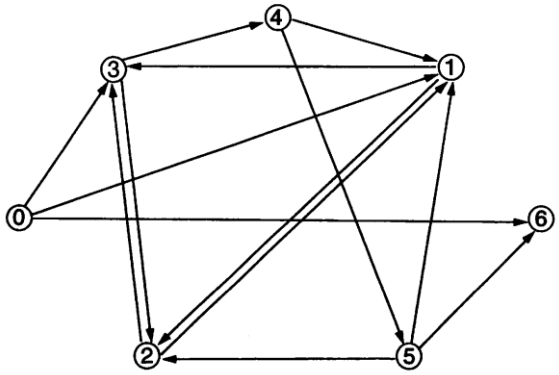


Figure 1.1: Directed Hamiltonian path graph.

Although there are well-known algorithms for deciding whether a randomly formed graph for given vertices has a Hamiltonian path or not, it seems likely that no efficient (that is, polynomial time) algorithm exists for solving it because the HPP has been proven to be nondeterministic polynomial complete (NP-complete). Similar to TSP case discussed earlier, these algorithms require an impractical amount of computer time to render a decision even for a modest size problem because all known algorithms for such a problem have exponential worst-case complexity[3, 4](i.e. the time increases exponentially as the number of vertices).

However, Adleman proposed an algorithm which simplifies the problem and makes it solvable. Simply, the solutions were filtered by going through a series of elimination steps, so that only those meeting the criteria to be a Hamiltonian path were left after the algorithm was completed. Adleman's algorithm was[2]:

1) Generate random paths through the graph.
2) Keep only those paths that begin with the starting vertex and finish with the ending vertex.
3) If the graph has n vertices, then keep only those paths that enter exactly n vertices.
4) Keep only those paths that enter all of the vertices of the graph at least once.
5) If any paths remain, say "Yes"; otherwise, say "No".

To implement Step 1 of the algorithm, Adleman assigned a 20bp single-stranded DNA (O_i) oligomer for each vertex in the graph *i*. Also, an oligonucleotide $O_{i \to j}$ was created to represent each edge $i \to j$. The sequence of $O_{i \to j}$ was the same as the 3' half

(10bp) of O_i (unless i=0, in which case it was all of O_i) followed by the 5' half (10bp) of O_j (unless j=6, in which case it was all of O_j). It is essential to note that this kind of construction maintains edge orientation, thus, $O_{2\rightarrow 3}$ is not the same as $O_{3\rightarrow 2}$. The Watson-Crick complementary oligomer of O_i was indicated by $\bar{O}_i$. Simply, the algorithm was begun by mixing 50 pmol of $\bar{O}_i$ and 50 pmol of $O_{i\rightarrow j}$ representing each vertex i (except i=0 and 6) and each edge $i \rightarrow j$, respectively in a solution. The $\bar{O}_i$ oligos served as a linker, allowing DNA molecules encoding the random paths shown in the graph to be produced as a result of the ligation reactions.

To implement Step 2, the DNA molecules constructed at the end of Step 1 were amplified by polymerase chain reaction (PCR) using O_0 and $\bar{O}_6$ as the primers. This reaction amplifies only the paths that begin with vertex 0 and end with vertex 6.

To implement Step 3, the DNA molecules produced at the end of Step 2 were run on an agarose gel so that different DNA oligos were separated since their mobility depends on the number of base pairs found in their sequences. Then, the 140bp double-stranded DNA band representing the paths going through seven vertices was extracted.

To implement Step 4, first, single-stranded DNA molecules were generated from double-stranded product of Step 3, and then incubated with $\bar{O}_1$ coupled with biotin-avidin magnetic beads. Only those single-stranded DNA molecules including O_1, which indicates that encoded pathway entered vertex 1 at least once, hybridized with $\bar{O}_1$and were retained. Exactly the same procedure was repeated with $\bar{O}_2$, $\bar{O}_3$, $\bar{O}_4$, and $\bar{O}_5$. Finally,

to implement Step 5, the product of Step 4 was amplified by PCR and run on an agarose gel to see if there was any Hamiltonian path remaining.

In reality, it can be easily detected that there is a unique Hamiltonian path between vertices 0 and 6 on the graph (Figure 1.1) by visual inspection. However, it took 7 days of lab work to solve the same problem using DNA molecules. Although this first study was slow, it was still a big step that demonstrated the following important features of DNA computing as stated in Adleman's paper[2]:

(i) Promising speed of DNA computing: If the ligation of two DNA molecules is considered as an operation and assuming that half of the approximately 4.2×10^{14} edge oligonucleotides (there are 14 edges and 50 pmol of DNA molecules was used for each edge) mixed in Step 1 were successfully ligated, then, approximately 10^{14} operations were carried out. This number could be easily scaled-up to $\sim 10^{20}$ if a micromole amount was used instead of picomole. Clearly, such number of operations per second during the ligation step was much bigger than that of a supercomputer ($\sim 10^{12}$ operations) available in 1994.

(ii) Remarkable Energy efficiency: The Gibbs free energy required for one ligation operation can be provided by the hydrolysis of a single ATP (adenosine triphosphate) molecule to adenosine monophosphate plus phosphate ($\sim -5.6 \times 10^{-20}$ J)[5, 6]. Thus, 1 J is enough for $\sim 2 \times 10^{19}$ such operations. This is remarkable energy efficiency since at most 34×10^{19} irreversible operations per joule can be executed according to the second law of thermodynamics[7, 8]. As a result, comparing to the energy consumed by a supercomputer

of that time (10^9 operations per J), the DNA computing was proven to be far more efficient.

(iii) High information storing density: It was shown that storing the information in DNA molecules allows for an information density of approximately 1 bit per cubic nanometer. This was a great improvement considering the low information density (~1 bit per 10^{12} nm^3) of videotapes, an example of existing storage media available at that time.

In 1995, Lipton outlined a series of DNA experiments to solve the famous satisfaction (SAT) problem, another famous NP-complete problem[9]. This work was important to show that biological computers were not only capable of solving the HPP as summarized above, but also, any other NP-complete problem.

In Lipton's paper, the following specific problem was studied. Consider the formula

$$F = (x \vee y) \wedge (x` \vee y`) \qquad (1.1)$$

where x and y are allowed to be 0 (false) or 1 (true), and $\vee$ and $\wedge$ are the logical "OR" and "AND" operations, respectively. The result of an "OR" operation ($x \vee y$) is 0 (false) only if both x and y are 0. Similarly, the result of an "AND" operation ($x \wedge y$) is 1 (true) only if both x and y are 1. Finally, x` and y` denotes the "negation" of x and y, respectively (x`=0 if x=1, and x`=1 if x=0). The aim of this SAT problem is to find Boolean values for x and y by which the formula F (Eq. 1.1) will be true. For the formula

given above, x=0 and y=1 satisfies that condition, as does x =1 and y=0, whereas x=y=0 and x=y=1 do not.

In general, a SAT problem is a formula (such as $C_1 \wedge C_2 \wedge \ldots\ldots \wedge C_m$) consisting of m clauses (C_i where *i=1, 2,...., m*) all of which are connected by "AND" ($\wedge$) operation. A clause is of the form $\upsilon_1 \vee \upsilon_2 \vee \ldots\ldots \vee \upsilon_k$ where each υ_j (*j=1, 2,...., k*) is a variable or its negation. More specifically, the formula F (Eq. 1.1) consists of two clauses: (1) $x \vee y$ and (2) $x` \vee y`$. To get the result of this formula true, the value of both clauses has to be 1 since they are connected with an "AND" operation. Thus, the goal can be redefined as finding values for the variables that make each clause have the value 1. Lipton states that any Boolean expression may be put into normal form as a sum of products or a product of sums. Therefore the SAT problem as presented above is completely general.

Lipton constructed the graph G_n (Figure 1.2)[9] to find the right solution for the formula F given in Eq. 1.1. The graph G_n has 7 vertices a_1, x, x`, $a_2$, y, y`, and a_3, which are connected by edges. In general, a graph constructed in this fashion starting at a_1 and ending at a_{n+1} encodes an n-bit binary number. At each stage, there are two possible paths: going through (i) an unprimed vertex that encodes 1 or (ii) a primed vertex that encodes 0. In particular, the graph G_n consists of 4 different 2-bit binary numbers as shown in Table 1.1. For example, the path $a_1 \rightarrow x \rightarrow a_2 \rightarrow y` \rightarrow a_3$ encodes the binary number 10 (Figure 1.2).

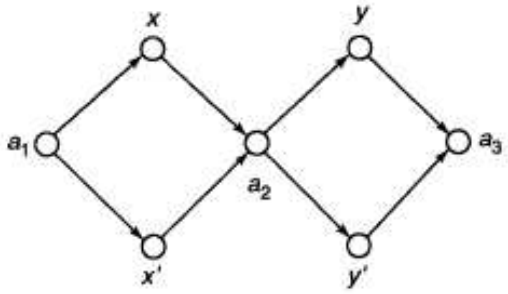


Figure 1.2: The graph G_n, which describes how to encode two-bit numbers.

The path	Encoded 2-bit binary number
$a_1 \rightarrow x` \rightarrow a_2 \rightarrow y` \rightarrow a_3$	00
$a_1 \rightarrow x` \rightarrow a_2 \rightarrow y \rightarrow a_3$	01
$a_1 \rightarrow x \rightarrow a_2 \rightarrow y` \rightarrow a_3$	10
$a_1 \rightarrow x \rightarrow a_2 \rightarrow y \rightarrow a_3$	11

Table 1.1: 4 different 2-bit binary numbers are encoded in graph G_n.

Lipton suggested that the graph G_n should be encoded in DNA molecules as follows[9]. Each vertex in the graph was represented by a randomly chosen 20bp short DNA strand consisting of two halves, p_i and q_i corresponding to the first and second halves. Thus, 5'$p_1q_1$3', 5'p_xq_x3', 5'$p_x{\cdot}q_x{\cdot}$3', 5'$p_2q_2$3', 5'p_yq_y3', 5'$p_y{\cdot}q_y{\cdot}$3', and 5'$p_3q_3$3' were the sequences related to vertices a_1, x, x`, $a_2$, y, y`, and a_3, respectively (5' and 3'

refer to the chemically distinct ends of DNA strands). The edge from *i*th vertex to *j*th vertex was also a 20bp DNA strand, 3' → 5' sequence of which is 3'$q\text{-}bar_i\, p\text{-}bar_j 5'$ (*q-bar* and *p-bar* are Watson-Crick complements of *q* and *p*). For example, the edge from vertex a_2 to y` was coded as 3'$q\text{-}bar_2\, p\text{-}bar_y$5'.

The sketch of proposed experimental procedure for a general case was[9]:

1) Put many copies of DNA sequence of the form 5'$p_i q_i$3' for each vertex in a test tube.

2) Add many copies of DNA sequence of the form 3'$q\text{-}bar_i\, p\text{-}bar_j 5'$ for each edge from vertex *i* to vertex *j* to the same test tube.

3) Add many copies of 3'$p\text{-}bar_1$5' and 3'$q\text{-}bar_{n+1}$5', which are complementary to the first half of the initial vertex and the last half of the final vertex, to the same test tube.

Incubating at a proper temperature to hybridize Watson-Crick complementary DNA strands with each other would form different double-stranded DNA molecules encoding all of the paths though the graph (n-bit binary numbers). As Lipton said, the probability of inadvertent paths being formed was very low since number of base-pairs for each DNA strand (20bp) is large enough.

Once all possible binary numbers were encoded in a tube (t_0), then, it was possible to find those ones which were solutions to the problem through the appropriate combination of extraction and recombination of relevant elements between tubes. For

example, following series of operations to find the correct solution to the formula F (Eq 1.1) represented by the graph G_n was suggested. First, *E (t, i, a)* was chosen to denote all of the sequences in test tube *t* for which the *i*th bit was equal to *a* (0 or 1). Then, the rest of the operation was[9]:

1) From test tube t_0, extract only those DNA strands which have "1" as their first bit. Put them in tube t_1 that corresponds to $E\ (t_0, 1, 1)$. Put the remaining strands in tube t_1', from which extract those DNA strands which have "1" as their second bit. Put them in tube t_2 that corresponds to $E\ (t_1', 2, 1)$. Combine t_1 and t_2 together in tube t_3.
2) From test tube t_3, extract only those DNA strands which have "0" as their first bit. Put them in tube t_4 that corresponds to $E\ (t_3, 1, 0)$. Put the remaining strands in tube t_4', from which extract those DNA strands which have "0" as their second bit. Put them in tube t_5 that corresponds to $E\ (t_4', 2, 0)$. Combine t_4 and t_5 in tube t_6.
3) The DNA strands left in tube t_6 represent 2-bit binary numbers that are the correct solutions to the problem.

Table 1.2 helps to understand how the operation outlined above works. Tube t_3 contains all those sequences (01, 10, 11) that satisfy the first clause. In the same way, t_6 contains all those from t_3 (01, 10) that satisfy the second clause. Note that these are the correct answers to the original problem. In general, solving a SAT problem including more clauses and variables is straightforward by following this procedure. Any SAT

Test tube	Values found in test tube
t_0	00, 01, 10, 11
$t_1 = E\ (t_1, 1, 1)$	10, 11
$t_1' = t_0 - t_1$	00, 01
$t_2 = E\ (t_1', 2, 1)$	01
$t_3 = t_1 + t_2$	01, 10, 11
$t_4 = E\ (t_3, 1, 0)$	01
$t_4' = t_3 - t_4$	10, 11
$t_5 = E\ (t_4', 2, 0)$	10
$t_6 = t_4 + t_5$	01, 10

Table 1.2: Values encoded by DNA strands in test tubes during the biological solution of Eq 1.1.

problem consisted of m clauses and n variables can be solved with a number of extract steps that is linear in m and one detect step[9].

After Lipton's proof of concept study, 4, 6 and 9 variables SAT problems were solved by several groups using different DNA computation techniques[10-13]. As a landmark, a group led by Adleman[14] extended the size to 20-variable 3-SAT problem by using a clever design to find the only correct solution over 2^{20} candidate (~ a million) molecules for the problem. Candidate molecules were filtered out by using electrophoresis with bound oligonucleotides and a few correct molecules left in a large

pool of candidates was amplified by PCR[15-16] (polymerase chain reaction) so that they could be detected.

For all of the studies summarized so far, the fundamental reactions in molecular biology, such as Watson-Crick DNA hybridization, PCR, or etc., are employed as a part of DNA based computation algorithms. However, none of these reactions are error free. As Deaton pointed out[17], the hybridization between strands that occurs at a temperature rather than the optimum melting temperature might create mishybridized structures, in which some of bases in the strand are "mismatched". Although the probability of such a mistake is low, this mistake can be further amplified, if the PCR is employed in the algorithm, and lead to difficulties in schemes for implementing large Boolean functions using DNA. Thus, an error-correcting scheme could perhaps be devised similar to the codes used in ordinary computers.

1.2. Mills, Platzman, Yurke (MYP) Model Hopfield Neural Network

A Mills, Yurke, Platzman (MYP)[18-20] network is a type of neural network in which DNA is used as the working substance. This network is effectively the implementation of a Hopfield neural network (HNN)[21], where the axons and neurons are replaced by DNA molecular reactions. Modeling of such a network is possible using matrix operations. However, in the MYP implementation, information consists of sets of oligomers, with each oligomer representing one particular element of information. Memory matrix elements are then represented by sets of ligated oligomers from the memory vectors. All of this is possible through an extension of Oliver's[22] matrix algebra,

where Oliver described a method for calculating the product of matrices using DNA. More detail describing the theory and construction of such a network is given below.

Neural Networks

Artificial neural networks (ANN) were originally constructed to be an approximation to real brain behavior. An ANN consists of physical analogues of both neurons and their connections. These analogues exhibit behaviors (such as the ability to alter synaptic weights) which are the critical underpinnings of computation in a biological neural network.

Ordinarily, these networks are implemented using electronic components[23-25], such as a network of amplifiers and a set of resistive connections. The resistive connections are placed in such a way that the outputs of the amplifiers are altered by the resistive connections before being used as inputs for the amplifiers on another cycle. The saturating nature of the amplifiers ensures that output is quenched to be in a range of values, and the actual output value is determined by the sum of the exhibitory and inhibitory inputs from the set of amplifiers. The nonlinearity of this arrangement allows for emulation of some properties of biological neural networks, such as memory, classification, and decision-making[26].

The ANN may be configured as a single or multi-layer system, where the output from one layer of amplifiers is fed forward (or backward) to another layer. It has been demonstrated that any neural network with at least one hidden layer of neurons is sufficient to solve any mathematically realizable operation[27].

To represent the network mathematically, a layer of neurons is modeled by a vector. Each vector entry is the output from one of the amplifiers in this layer. Connections between layers are represented by matrices, with the components of the matrix representing the strength of a connection between a particular neuron in one layer and a particular neuron in the next layer.

As an example of an ANN, consider the perceptron. This network is a single layer of inputs connected to a decision node. This node then produces an output based upon the decision function being used by the node. This process can be represented by using a set of input neurons $\{I_i\}$, and a set of connection strengths with the decision node, $\{W_{ij}\}$. These connection strengths can each be either inhibitory or exhibitory. As was mentioned above, a more general architecture (one with a layer of nodes which are hidden from the input and output layers) can be used to solve any soluble function.

Hopfield Neural Networks

A particularly interesting case of an ANN can be realized in the Hopfield Neural Network (HNN)[21]. In the HNN, the network is constructed to serve as a content addressable memory[23-24]. The vectors introduced to the network are information, where each vector entry represents a portion of the information to be remembered, such as a pixel in a picture, or a note in a song. A particular piece of information (piece 'a'), then, is represented by a D-dimensional vector $\vec{V}^a = \sum_{i=1}^{D} V_i^a \vec{e_i}$ with basis vectors e_i, (i=1,2,…*D*). All amplitudes satisfy the condition that $V_i = \pm 1$. The establishment of the

memory matrix is then performed by constructing the sum the outer products of each memory vector with itself. Given 'M' memories, any matrix component can be represented as

$$T_{ij} = \sum_{a=1}^{M} V_i^a V_j^a \tag{1.2}$$

We assume that the information contained in the memory vectors has been optimally compressed, so that the vector components are close to a random string of +1's and -1's. Hopfield assumed that all diagonal components of the memory matrix needed to be set to zero ($T_{ij} = 0$ for i=j), and that the symmetry condition implied in the matrix construction rule $T_{ij} = T_{ji}$ was allowed to stand without alteration. (Recall of an information vector is accomplished by the introduction of a "clue" vector to the matrix, which is a vector with some of the entries missing (set to '0'). For example, consider the clue vector

$$\vec{U}^b = \sum_{i=1}^{M} U_i^b \vec{e_i} = \sum_{i=1}^{n} V_i^b \vec{e_i} \tag{1.3}$$

with $U_i^b = \pm 1$ for $i=1, \ldots, n$ and $U_i^b = 0$ for $i=n+1, \ldots, D$. This has converted the representation of information to a duobinary system. Acting on the clue vector with the memory matrix gives[19]

$$X_i^{(1)} = \sum_{j=1}^{D} T_{ij} U_j = \sum_{j=1}^{n} \sum_{a=1}^{M} V_i^a V_j^a V_j^b = n V_i^b + \sum_{j=1}^{n} \sum_{a=1, a\neq b}^{M} \pm V_i^a = n V_i^b + \sum_{k=1}^{n(M+1)} \pm 1 \approx n V_i^b \pm \sqrt{n(M+1)} \tag{1.4}$$

where we can use the assumption of a random distribution of +1 and -1 in memory vector entries to get the approximate mean and standard deviation for our results. Repeating this process gives[19]

$$X_i^{(2)} = \sum_{j=1}^{D} T_{ij} X_j^{(1)} = \sum_{j=1}^{D}\sum_{a=1}^{M} V_i^a V_j^a [nV_j^b \pm \sqrt{n(M+1)}] = n\sum_{a\neq b=1}^{M} V_i^a \sum_{j=1}^{D}(\pm 1) + \sum_{j=1}^{D} nV_i^b + \sqrt{n(M+1)}\sum_{a\neq b=1}^{M} V_i^a \sum_{j=1}^{D}(\pm 1) \approx$$

$$nDV_i^b \pm \sqrt{(M+1)Dn(M+1) + n^2(M+1)D} \qquad (1.5)$$

This result is obtained by using the property that uncorrelated terms in the result can be combined to get the standard deviation by using the sum of squares method. If n<M, the amplitude of $X_i^{(2)}$ will be appreciably larger that its standard deviation, provided $M^2<nD$. Finally, a third iteration gives[19]

$$X_i^{(3)} = nD^2V_i^b \pm \sqrt{(M+1)D^3n^2 + (M+1)DnD(M+1)[M+1+n]} \quad (1.6)$$

The standard deviation in this iteration will be negligible compared to the mean, as long as $n<M$ and $M^3<nD^2$. If this trend continues unabated, after the p-th iteration, there will be an unambiguous recall based on a clue vector of n-elements as long as n<M, and $M<D(n/D)^{1/p}$. Therefore, if there are enough iterations, it should be possible to recall almost as many memories as there are components in the information vectors, even for a small clue vector. This is due to the pseudoorthogonality of the information vectors assumed previously.

Biological Implementation of the HNN

The implementation of the HNN using DNA as a working substance involved finding a way to represent matrices and vectors (and their allowed operations). Since DNA is a stable helical polymer with selective hybridization, it was possible to exploit these properties to effect the aforementioned operations. DNA consists of a sugar-phosphate backbone with bases attached to it (A,C,G,T). There are 2 of these backbones in a DNA strand, zipped together by attractive interactions between complementary bases (A and T, G and C).

In the proposed system of computation, each vector entry is a set of identical DNA oligomers, which serves as a "basis" for one of the memory vectors. The connection strengths of the outer products in the memory matrix are formed by construction of a set of ligated vector entries, where each vector entry is ligated to itself in solution. The memory matrix, then, is nothing more than that set of all ligated vector entries with themselves, stored in solution and ready to act on an input clue vector. DNA implementation of HNN will be discussed in Chapters 3 and 4 with more details.

References:

[1] Feynman, R. P. *Minaturization*. Ed. Gilbert, D. H. New York: Reinhold Publishing Corporation, 1961. 282-296. Print.

[2] Adleman, L. M. "Molecular Computation of Solutions to Combonatorial Problems" *Science* 266 (1994): 1021-1024. Print.

[3] Garey, M. R. and Johnson, D. S. *Computers and Intracibility: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979. Print.

[4] Karp, R. M. *Complexity of Computer Computations*. Eds. Miller, R. E. and Thatcher, J. W. New York: Plenum Press, 1972. 85-103. Print.

[5] Watson, J. D., et al. *Molecular Biology of the Gene.* Menlo Park, CA: Benjamin/Cummings, 1987. Print.

[6] Engler, M. J. and Richardson, C. C. *The enzyme*. Ed. Boyer, P. D. New York: Academic Press, 1982. 3-29. Print.

[7] Schneider, T. D. "Theory of Molecular Machines. II. Energy Dissipation from Molecular Machines" *J. Theor. Biol.* 148 (1991): 125-137. Print.

[8] Merkle, R. C. "Reversible Electronic Logic Using Switches" *Nanotechnology* 4 (1993): 21-40. Print.

[9] Lipton, R. J. "DNA Solution of Hard Computational Problems" *Science* 268 (1995): 542-545. Print.

[10] Liu, Q., et al. "DNA Computing on Surfaces." *Nature* 403 (2000): 175-179. Print.

[11] Yoshida, H. and Suyama, A. "Solution to 3-SAT by Breadth First Search." *DNA Based Computers V*, Eds. Winfree, E. and Gifford, D. K. Providence, RI: American Mathematical Society, 1999. 9-20. Print.

[12] Sakamoto, K., et al. "Molecular Computation by DNA hairpin formation." *Science* 288 (2000): 1223-1226. Print.

[13] Faulhammer, D., et al. "Molecular Computation: RNA solutions to chess problems." Proc. Natl. Acad. Sci. U.S.A. 97 (2000): 1385-1389. Print.

[14] Braich, R. S., et al. "Solution of a 20-variable 3-SAT problem on a DNA computer." Science 296 (2002): 499–502. Print.

[15] Saiki, R. K., et al. "Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia." *Science* 230 (1985): 1350–1354. Print.

[16] Mullis, K., et al. "Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction." *Cold Spring Harb. Symp. Quant. Biol.* 51 Pt 1 (1986): 263–273. Print.

[17] Deaton, R., et al. "Reliability and Efficiency of a DNA-based Computation." *Phys. Rev. Lett.* 80 (1998): 417-420. Print.

[18] Mills Jr., A. P., Yurke, B., and Platzman, P. M. "Article for analog vector algebra computation." *BioSystems* 2 (1999): 175-180. Print.

[19] Mills Jr., A. P., Yurke, B., and Platzman, P. M. "DNA analog vector algebra and physical constraints on large-scale DNA-based neural network computation." *DNA Based Computers V.* Eds. Winfree, E. and Gifford, D. K. Providence, RI: American Mathematical Society, 2000. 65-73. Print.

[20] Mills Jr., A. P. "Experimental aspects of DNA neural network computation." *Soft Computing* 5 (2001): 10-18. Print.

[21] Hopfield, J. J. "Neural networks and physical systems with emergent collective computational abilities." *Proc. Natl. Acad. Sci. USA* 79 (1982): 2554–2558. Print.

[22] Oliver, J. "Matrix Multiplication with DNA." *Journal of Molecular Evolution* 45 (1997): 161 – 167. Print.

[23] Jackel, L. D., et al. "Artificial neural networks for computing." *J. Vac. Sci. Technol.* B4 (1986): 61-63. Print.

[24] Wasserman, P. D. *Neural Computing, Theory and Practice.* New York: Van Nostrand Reinhold, 1989. Print.

[25] Bose, N. K. and Liang, P. *Neural Networks Fundamentals with Graphs, Algorithms and Applications.* New York: McGraw-Hill, 1996. Print.

[26] Penny, W. and Frost, D. "Neural Networks in clinical medicine." *Med. Dec. Making* 16, (1996): 386-398. Print.

[27] Funahashi, K. "On the approximate realization of continuous mappings by neural networks." *Neural Networks* 2, (1989): 183-192. Print.

Chapter 2 - Application of T4 Gene 32 Protein to the improvement of isothermal linear amplification (ILA) of DNA

Abstract

The Isothermal Linear Amplification (ILA) reaction using double-stranded DNA (dsDNA) as the template can be used to produce single-stranded DNA (ssDNA) output products[1-2]. We consider this type of method as an alternative of Asymmetric Polymerase Chain Reaction (APCR), in which amplification yields ssDNA as well[3]. A drawback to the use of ILA is that in the majority of cases undesired products composed of strands of varying lengths are formed in addition to the desired output. According to our results, the quantity of fragmented strands increases monotonically as the reaction time and length of the desired output product increase. Herein 15, 30, 45 and 60 minutes of ILA reactions are performed for each of 3 different lengths of dsDNA templates (50, 70 or 100bp). We demonstrate that adding T4 Gene 32 single-strand binding protein (ssBP) to the reaction mix prevents the formation of fragment strands. With this addition, ILA is freed from its hitherto deleterious characteristic. In addition, we show that a ssDNA would rather hybridize with its complement than be attached to ssBP molecules. This leaves the mystery of what is the mechanism by which ssBP prevents fragmentation.

2.1. Introduction:

The ability to amplify specific DNA sequences is one of the most important tools in modern molecular biology. There are currently two widely used methods: Polymerase Chain Reaction (PCR) [4-6] and isothermal rolling-circle amplification[7-9]. In this study, Isothermal Linear Amplification (ILA), a variant of the isothermal rolling-circle amplification method, is employed[1-2].

To produce an amplification reaction, we need to design a cyclic chain reaction that will restore the initial state after each synthesis of an output product. The cyclic process of ILA (Figure 2.1) composed of "Nicking" and "Release & Extend" parts can be summarized as follows[1-2]:

1) The single-strand nicking enzyme (Nt.BbvCI) cleaves the phosphodiester bond at the nicking site on the lower strand of dsDNA.

2) DNA polymerase (Klenow fragment 3'→5' exo-) is employed to release the output strand as it begins elongating the lower strand starting from the nicking point to the left until the structure becomes fully dsDNA. In cases single-strand binding proteins (T4 Gene 32) are present in the solution, the output strands will be captured by those binding proteins.

3) Steps 1 and 2 are repeated until the reaction reaches a saturation point. Most likely, inactivation of the nicking enzyme as the reaction continues causes the saturation[10].

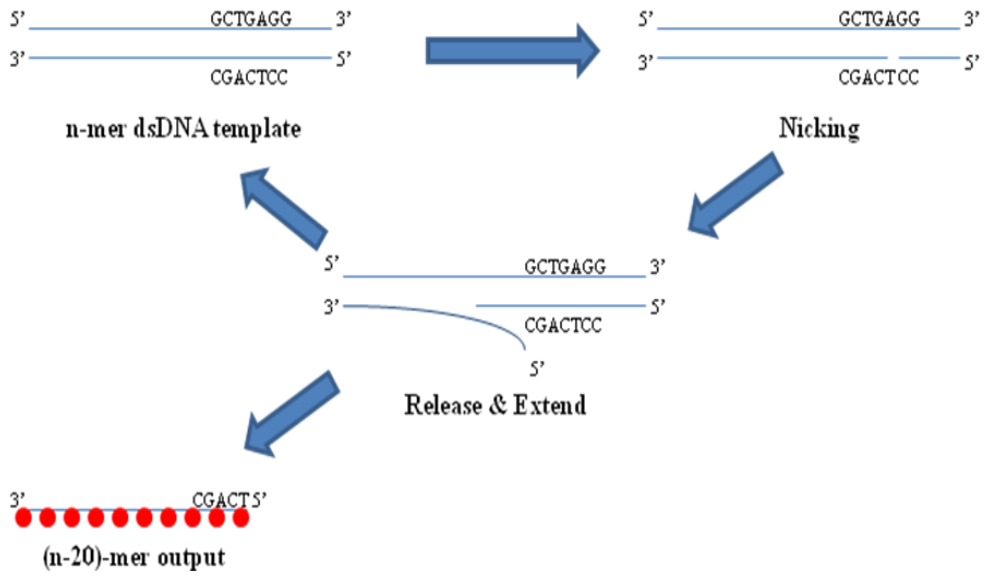


Figure 2.1: Visual description of ILA reaction. The dots represent the T4 Gene 32 ssBP molecules.

We don't use a high reaction temperature (55 - 60 °C) that must be also compatible with enzymes to dissociate the output strand[10]. Instead, this job is fulfilled by the strand displacement activity of Klenow fragment (3'→5' exo-) polymerase[1-2] as stated above. The advantage of this method is that it has the potential to produce any arbitrary sized output product, while one can only produce oligonucleotides up to a short length (15-20bp) using an ILA reaction employing the high temperature to release output strands[10].

Although the quantity of DNA product produced by PCR is a geometric function of the number of cycles, the reaction requires special laboratory instruments (such as a thermocycler) to ensure the reaction proceeds at different temperatures in a cyclic process. This can be considered one disadvantage of the method. On the other hand, ILA

is also a cyclic process as stated above but occurs at a single fixed temperature, thus, requiring only a simple incubator. However, the amount of product in this process is a linear function of cycle number, unless a special protocol to create geometric amplification is utilized[10].

In our experience, neither PCR nor ILA is able to guarantee that there will be a clean result (no extra strands other than target output product) at all times. Because of various factors, i.e. the size of the template or expected output DNA, the sequence, unwanted reactions between impurities found in the reaction mix and enzymes, etc., there is a high probability of fragmentation (polymerization) at the end of some reactions. The purpose of this study is to assess whether adding T4 Gene 32 single-strand binding protein to the reaction mix at the start of ILA will assist in the production of fragment-free ILA results[11-12]. We also test whether or not a ssDNA attached to ssBP molecules hybridize with its complementary strand by a "ssBP vs. Hybridization" experiment.

2.2. Materials and Methods:

The Oligonucleotides used in the ILA reactions and the "ssBP vs. Hybridization" experiment are shown in Table 2.1 and 2.2, respectively. Note that the nicking enzyme recognition nucleotides are underlined in all sequences and the targeted cleavage site is indicated by || sign in Table 2.1. Oligonucleotides were synthesized by Integrated DNA Technologies, Inc. The Nuclease-free water (DEPC-Free) used to dilute DNA samples and added into our reaction mixes was also purchased from Integrated DNA Technologies, Inc. Nt.BbvCI the nicking enzyme (Cat#R0632S, 10 units/µl), Klenow

fragment (3'→5' exo-) polymerase (Cat#M0212S, 5 units/µl), and T4 Gene 32 Protein (Cat#M0300S, 10 µg/µl) were obtained from New England Bio Labs.

ILA 50mer - upper strand:	5'- CTTGTCAACCCTCCTGCCACCAACTGCTGAGGAGCACCT GACTATGTTGG -3'
ILA 50mer - lower strand:	5'- CCAACATAGTCAGGTGCTCC||TCAGCAGTTGGTGGCAGGA GGGTTGACAAG -3'
ILA 70mer - upper strand:	5'- GATAGAGTGTAGGCATTAGGCTTGTCAACCCTCCTGCCA CCAACTGCTGAGGAGCACCTGACTATGTTGG -3'
ILA 70mer - lower strand:	5'- CCAACATAGTCAGGTGCTCC||TCAGCAGTTGGTGGCAGGA GGGTTGACAAGCCTAATGCCTACACTCTATC -3'
ILA 100mer - upper strand:	5'- ACGATGCGACGCAGGTCTAACACCGACATTGATAGAGTG TAGGCATTAGGCTTGTCAACCCTCCTGCCACCAACTGCTGAGG AGCACCTGACTATGTTGG -3'
ILA 100mer - lower strand:	5'- CCAACATAGTCAGGTGCTCC||TCAGCAGTTGGTGGCAGGA GGGTTGACAAGCCTAATGCCTACACTCTATCAATGTCGGTGT TAGACCTGCGTCGCATCGT -3'

Table 2.1: Upper and lower strands of the 3 dsDNA templates (50, 70 and 100bp) used in the ILA reactions.

30mer	5’- TCAGCAGTTGGTGGCAGGAGGGTTGACAAG -3’
30mer-bar	5’- CTTGTCAACCCTCCTGCCACCAACTGCTGA -3’
50mer	5'- TGGATACTGAGTCACATCACACGCTTAGGAACCGTT GAGTCCGTATGTCA -3'

Table 2.2: Oligonucleotides used in the “ssBP vs. Hybridization” experiment. 30mer and 30mer-bar are complementary oligonucleotides.

Producing dsDNA templates. dsDNA templates (50, 70, or 100bp) used in ILA reactions were prepared by simple hybridization reactions in 3 different vials. Each reaction mixture with a total volume of 100 µl was composed of 50 mM Tris-HCl (pH 7.5 @ 25 °C), 10 mM $MgCl_2$, 1mM ATP, 10 mM Dithiothreitol, 1 µM corresponding upper and lower DNA strands (Table 2.1), and Nuclease-free water. After vortexing, all mixes were incubated at 56 °C (the average optimum hybridization temperature for all 3 cases) for 1 hour[13]. Then, 40 µl of SOPE™ (Solid-phase Oligo Protein Elimination) Resin was added into each mix and vortexed. After waiting 3 minutes, the mixes were run through Performa® Gel Filtration cartridges (a brand-new cartridge for each mix) by centrifuging at 2900 rpm for 2 minutes. dsDNA templates were purified by this last step, in which all unhybridized upper or lower ssDNA were removed. QuickStep™2 Purification System kits containing SOPE and Performa Gel Filtration cartridges were purchased from Edge Bio Inc.

Isothermal Linear Amplification (ILA) Reaction. ILA reactions are divided into two different groups: (i) ILA without ssBP, (ii) ILA with ssBP. The only difference between the two cases was that T4 Gene 32 Protein wasn't added into the reaction mix for the first group of reactions while it was used for the second group. All reactions were carried out in a reaction mixture with a total volume of 50 µl containing 10 mM Tris-HCl (pH 7.9 @ 25 °C), 50 mM NaCl, 10 mM $MgCl_2$, 1 mM Dithiothreitol, 25 units Nt.BbvCI nicking enzyme, 5 units Klenow fragment (3'→5' exo-) polymerase, 30 µg T4 Gene 32 single-strand binding Protein (only for the reactions in the second group), 1000 µM Deoxynucleotide solution mix (dNTP) (New England Biolabs, Cat#N0447S), ~0.15 µM template oligonucleotides (50, 70 or 100bp dsDNA), and Nuclease-free water. After vortexing, mixtures were incubated at 37 ºC for 15, 30, 45 or 60 minutes, then, heated at 80 ºC for 20 minutes to deactivate the enzymes and cooled down to room temperature slowly.

"ssBP vs. Hybridization" Experiment. The protocol of this experiment consisted of the following 4 steps:

(1) A reaction mix with a total volume of 50 µl containing 10 mM Tris-HCl (pH 7.9 @ 25 °C), 50 mM NaCl, 10 mM $MgCl_2$, 1 mM Dithiothreitol, 30 µg T4 Gene 32 ssBP, 0.2 µM 30mer ssDNA (Table 2.2), and Nuclease-free water was prepared in a vial. After vortexing, the mix was incubated at 37 ºC for 1 hour. Then, 5 µl of solution was transferred into a new vial to be used for analysis.

(2) 9 µl of 1 µM 30mer-bar ssDNA (the complement of the 30mer, Table 2.2) and 1 µl of 10x NeBuffer 2 were added into the remaining solution from the first step. A mix with a total volume of 55 µl containing 10 mM Tris-HCl (pH 7.9 @ 25 °C), 50 mM NaCl, 10 mM $MgCl_2$, 1 mM Dithiothreitol, 27 µg T4 Gene 32 ssBP, 0.163 µM 30mer and 30mer-bar ssDNA, and Nuclease-free water was obtained. After incubating at 45 ºC for 1 hour, which is a suitable condition for hybridizing 30mer and 30mer-bar complementary strands, 5 µl of solution was transferred into a new vial to be used for analysis.

(3) 9 µl of 1 µM 50mer ssDNA (Table 2.2) and 1 µl of 10x NeBuffer 2 were added into the remaining solution from the second step. A mix with a total volume of 60 µl containing 10 mM Tris-HCl (pH 7.9 @ 25 °C), 50 mM NaCl, 10 mM $MgCl_2$, 1 mM Dithiothreitol, 24.5 µg T4 Gene 32 ssBP, 0.136 µM 30mer and 30mer-bar ssDNA, 0.15 µM 50mer ssDNA and Nuclease-free water was obtained. After incubating at 37 ºC for 1 hour, 5 µl of solution was transferred into a different vial to be used for analysis.

(4) The remaining 55 µl solution was incubated at 65 ºC for 20 minutes to deactivate ssBP molecules.

Analysis of product. 10% .75 mm 10 well polyacrylamide gels were used for electrophoresis to analyze the products of the reactions. To produce a gel, the following protocol was used: 1250 µl of 40% acrylamide (Sigma-Aldrich), 3215 µl of double-distilled water, 500 µl of 10x Tris-Borate-EDTA buffer (89 mM Tris-Borate, 2 mM

EDTA, pH 8.3, Sigma-Aldrich), 35 µl of 10% liquid ammonium pelsulfate (obtained by adding 1 g of ammonium persulfate powder (Sigma-Aldrich) into 10 ml of double-distilled water) and 1.75 µl of TEMED (Sigma-Aldrich) were mixed in a plastic vial. After shaking 10-15 seconds, the liquid mix was loaded into the gap between two glass plates (front and back) attached by being located on a gel caster (Bio-Rad) and comb allowing for 10 wells was placed at the top. Gel was ready to use after waiting for 45-60 minutes for polymerization.

Electrophoresis was performed in the following order: First, 10% polyacrylamide gels prepared earlier were placed into the Mini-PROTEAN® Tetra Cell vertical electrophoresis tank (Bio-Rad) that was connected to a power supply (Sigma-Aldrich). 1x Tris-Borate-EDTA buffer diluted from 10x concentration was loaded into the tank as the electrophoresis running buffer. Then, 1 µl of 1x Bromophenol Blue DNA loading buffer (0.25% w/v Bromophenol with 50% Glycerol, Sigma-Aldrich) was added into 5 µl of each sample and mixed on a shaker table. These solutions were loaded into wells of 10% polyacrylamide gel in the desired order. 20bp dsDNA low ladder (Sigma-Aldrich) with 50% G+C was used as a DNA marker. 154 V potential difference was applied for 35 minutes to run the electrophoresis.

A gel staining solution to stain DNA strands was prepared by diluting 5 µl of 10000x SYBR Gold (Invitrogen) into 100 ml 1x Tris-Borate-EDTA buffer. Gels were carefully removed from their glass plates and soaked in this solution on a mixer for 20 minutes at room temperature. Finally, we took CCD (charge-coupled device) camera

images of the gels containing DNA strands stained by SYBR Gold under a 312 nm UV light (Fisher Scientific). The excitation maxima for this type of stain is ~300nm. The exposure time was 5 seconds for all images. The results were analyzed by using Kodak 1D image analysis system (Fisher Scientific).

2.3. Results and Discussion

Isothermal Linear Amplification. According to our template-dependent design, an ILA reaction starting with *n*-base pairs dsDNA with a nicking site located between the 20^{th} and 21^{st} base pairs (from 5' end) on the lower strand produces *(n-20)*-base pairs ssDNA output. Three different lengths of dsDNA templates (50, 70, and 100bp) purified by QuickStep™2 Purification System (Figure 2.2) were tested in ILA reactions. The following results were obtained:

ILA of 50bp dsDNA template: Four identical ILA reaction mixes not including ssBP were prepared in different vials, then, incubated at 37 °C for 15, 30, 45 and 60 minutes. As shown in Figure 2.3.a lanes 2-5, only 30bp ssDNA molecules that was the expected output was produced at the end of all four cases. This result supports the fact that ILA reactions designed to synthesize short oligonucleotides do not yield nonspecific DNA products although there might be exceptions.

ILA of 70bp dsDNA template: Again, we prepared four reaction mixes not including ssBP and incubated them at 37 °C for 15, 30, 45 and 60 minutes. Figure 2.3.b lane 2 shows that not only 50bp expected output ssDNA but also two other nonspecific DNA products

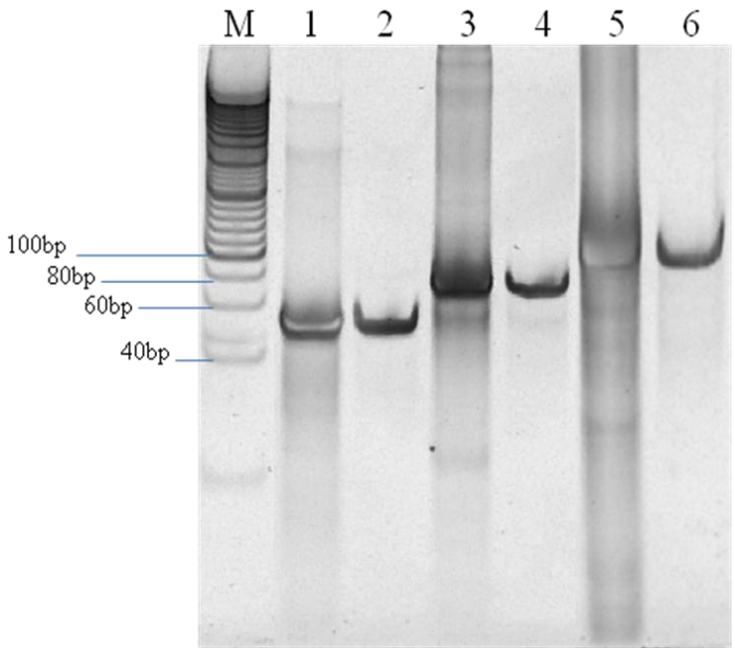


Figure 2.2: Lanes 1&2 refers to 50bp dsDNA template before and after purification step. In a similar manner, Lanes 3&4 and 5&6 pairs correspond to 70 and 100bp dsDNA templates. Lane M of each image contains 20bp dsDNA marker. 5 µl samples are shown in each lane.

(corresponding to two separate dim bands) were produced at the end of 15 minutes incubation. Notice that the amount of those nonspecific products was increased monotonically as the reaction time became longer (lanes 3-5 representing 30, 45, and 60 minutes incubations, respectively). Since it is almost impossible to guess what kind of unexpected reaction caused the formation of those fragment strands, we don't have certain information about the structure or size of those molecules except their mobility. Last two lanes show the result of 60 minutes ILA reaction with ssBP before and after heat deactivation. The output ssDNA, which were captured by ssBP, were stuck at the top of lane 6 before heat deactivation because ssBP is a big enough protein (~35,000 daltons)[14]

so that ssDNA attached to ssBP can't penetrate the gel. After heat deactivation of the ssBP and other enzymes at 80 °C for 20 minutes, 50bp expected output molecules were again present (lane 7). If lanes 5 and 7 (showing the results of 60 minutes ILA reactions without or with ssBP) are compared, it is clearly observed that adding T4 Gene 32 ssBP into the reaction mix prevented the formation of nonspecific DNA molecules, and does not slow down the reaction rate.

ILA of 100bp dsDNA template: In a similar manner to previous cases, ILA reaction mixes without ssBP were prepared, then, incubated at 37 °C for four different lengths of time. The result of 15 minutes reaction seen in Figure 2.3.c lane 2 indicates that at least one nonspecific DNA product other than 80bp expected output ssDNA came out. Not only the number of different nonspecific products but also their amounts were escalated as the reaction time became longer (lanes 3-5). More specifically, as an example 60 minutes reaction yielded 5 or 6 different fragment strands each of which corresponded to one of three obviously observable or other very dim bands (lane 5). In fact, the amount of one of the nonspecific products was more than that of the expected output. The result of 60 minutes ILA reaction with ssBP before and after heat deactivation is shown in last two lanes. As we stated earlier, the output ssDNA, which were captured by ssBP, were stuck at the top of lane 6 before heat deactivation. After incubating the sample at 80 °C for 20 minutes to denature the enzymes including ssBP, 80bp expected output molecules were again present (lane 7). Comparing the results of 60 minutes ILA reactions without or with ssBP (lanes 5 and 7) shows that adding T4 Gene 32 ssBP into the reaction mix stops the fragmentation problem, and does not affect the reaction rate.

Even though it was expected that Klenow fragment polymerase would stop adding nucleotides when the dsDNA structure was completed, for all 3 cases, we observed that the polymerase had a tendency to attach extra nucleotides at 3' side of lower strand, which is the end of the output strand. Determination of the exact number of added nucleotides was impaired by the lack of precision on the dsDNA ladder used in our gels.

Note that the ILA reaction for all three cases slows down noticeably after 45 minutes or so as if an essential component of the reaction is being inactivated as time goes by (Figure 2.4). As we stated earlier, most likely the nicking enzyme is responsible because there are previous reports showing that more nicking enzyme allows the reaction to continue longer[10].

According to our preliminary experiments, if the amount of nicking enzyme is almost equal to or lower than that of the polymerase, then there was no observed reaction. The nicking enzyme/polymerase ratio was set to 5:1 to be able to get a successful reaction. In reality, the balance between the nicking enzyme and DNA polymerase is a very complex function as stated above. However, our research did not seek to optimize this ratio; rather we sought merely to find a method of producing results of ILA without nonspecific ILA products. We chose to use Nt.BbvCI and Klenow Fragment (3'→5' exo) because the reaction temperature for both enzymes is 37ºC, which is also the preferred reaction temperature for T4 Gene 32 ssBP.

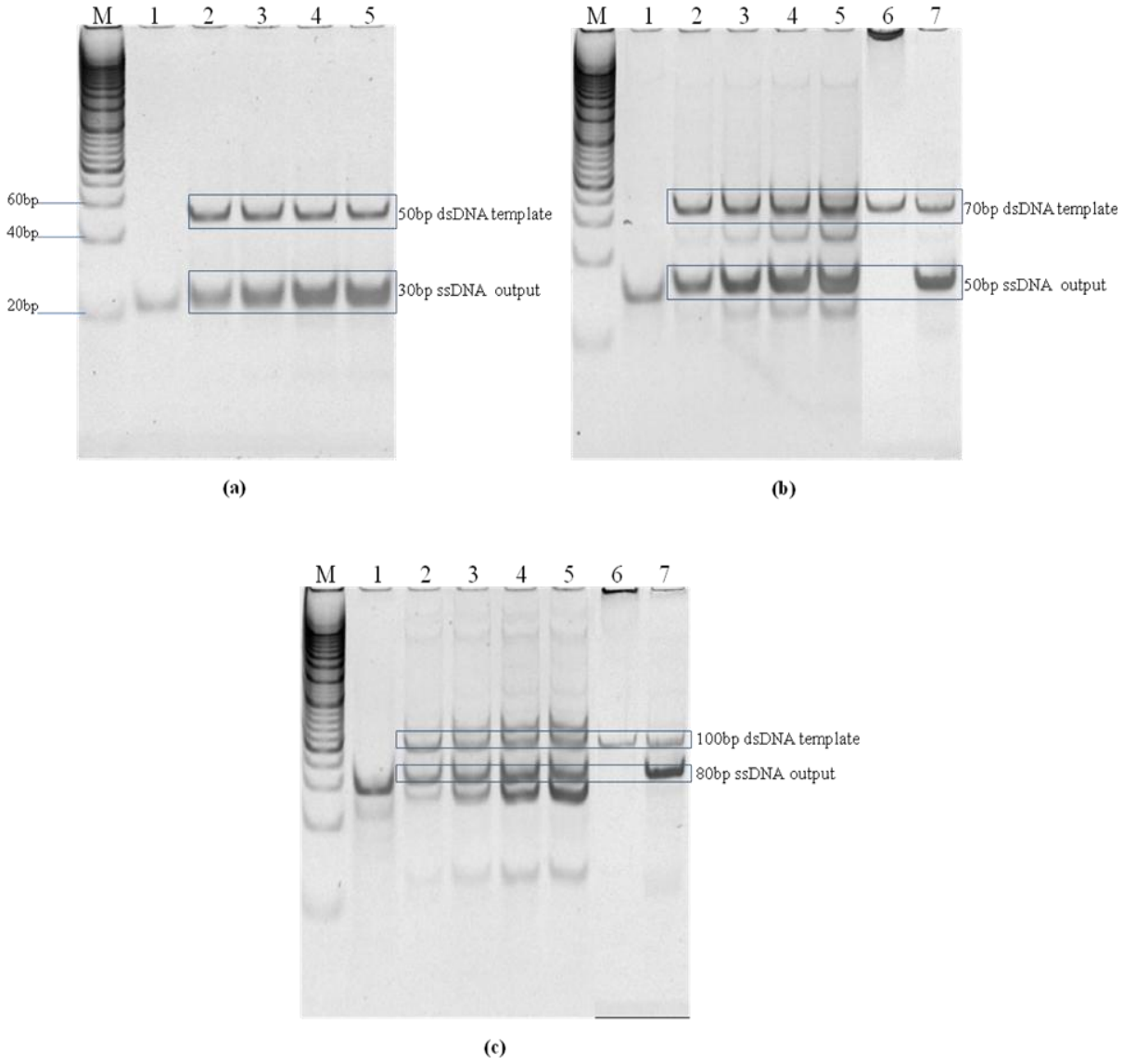


Figure 2.3: 10% polyacrylamide gel images obtained at the end of ILA reactions. (a), (b) & (c) Lane 1 correspond to 1 picomole of the expected 30, 50 and 80bp output ssDNA. (a) Lanes 2-5 indicate the result of ILA reactions of 50bp dsDNA template for 15, 30, 45 and 60 minutes (without ssBP & after heat-deactivation). In a similar manner, (b) & (c) Lanes 2-5 corresponds to the results of ILA reactions of 70 and 100bp dsDNA templates. (b) & (c) Lanes 6-7 indicate the result of ILA reactions of 70 and 100bp dsDNA templates before and after heat deactivation, respectively (with ssBP). Lane M of each image contains 20bp dsDNA marker. 5 µl samples are shown in each lane.

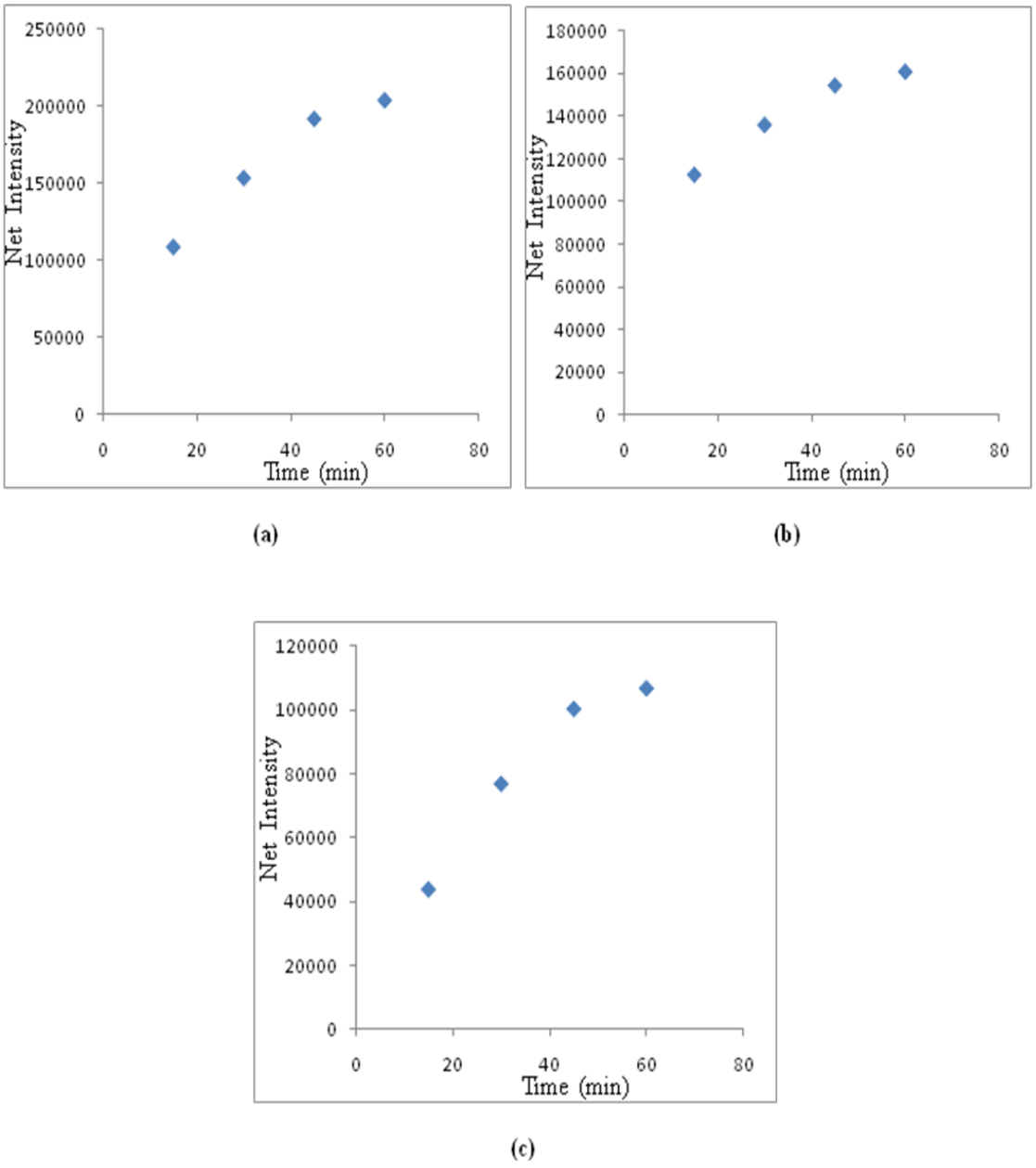


Figure 2.4: Net light intensity (proportional with the amount of amplified output ssDNA found in their corresponding bands) vs. time graphs for ILA reactions of (a) 50bp (b) 70 and (c) 100bp dsDNA templates.

"ssBP vs. Hybridization" Experiment. A mix including 30bp ssDNA (30mer) and ssBP molecules was prepared, then, incubated at 37 °C for 1 hour. The result of this operation is shown in Figure 2.5 lane 4. Since oligonucleotides were captured by ssBP molecules, they appeared at the top of the lane 4. Incubation at 45 °C for 1 hour after adding the complementary strands (30mer-bar) into the mix generated 30bp dsDNA (lane 5). This result indicates that ssDNA molecules, which had been already captured to T4 Gene 32 ssBP, were capable of hybridizing with their complementary strands. To understand whether ssBP was still in active condition, 50 bp ssDNA oligonucleotides (50mer) was added into the mix. As seen at the top of lane 6, those 50bp ssDNA molecules were captured by ssBP. After heat deactivation at 65 °C for 20 minutes, 50bp ssDNA were again present in the solution along with previously formed 30bp dsDNA (lane 7).

2.4. Conclusion

Our results showed that if ILA is utilized to yield short oligonucleotides (≤30bp), nonspecific ILA products may be avoided. However, if the size of the desired output product (ssDNA) is longer (≥40bp), then we observed fragment strands in the solution at the end of ILA reaction. According to our results, it is deduced that the amount of the undesired products (impurities) increases as the size of the expected output ssDNA or the reaction time becomes longer.

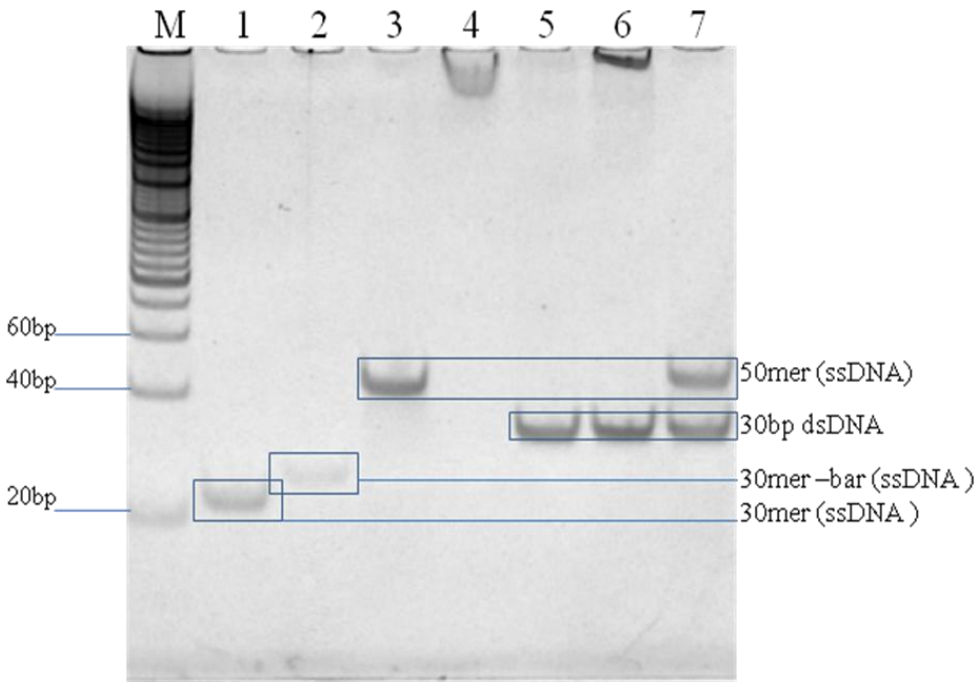


Figure 2.5: 10% polyacrylamide gel image obtained at the end of "ssBP vs. Hybridization" experiment. 1 picomole of 30mer, 30mer-bar, and 50mer are shown in lanes 1-3. Lanes 4-7 indicates the results of following 4 steps: (i) incubating 30mer with ssBP, (ii) addition of 30mer-bar, (iii) addition of 50mer, and (iv) heat deactivation. Lane M contains 20bp dsDNA marker. 5 µl samples are shown in each lane.

According to preliminary experiments, we believe that the fragmentation problem might happen due to the interaction between reaction enzymes and some ssDNA. These ssDNA can come from different sources: they may be attributable to a priori presence in the solution (unhybridized upper or lower strands of dsDNA template although we tried to remove these impurities using QuickStep™2 Purification System), or may exist as a product of the ILA reaction. Another possible source of undesired DNA output product is the result of the proceeding ILA reaction. These errors can occur due to mistakes made by either the polymerase or nicking enzyme. Hence, we decided to add T4 Gene 32 ssBP

into the reaction mix at the beginning to keep indeterminate DNA strands away from getting into an interaction with polymerase and nicking enzyme. As expected, the binding protein maintained the order of reaction, thus, only the targeted output ssDNA was synthesized.

Again, our preliminary experiments show that if the amount of ssBP added into solution isn't enough, then there will be a small amount of fragmentation produced during the ILA procedure. In conclusion, it is important to choose the amount of ssBP carefully in order to achieve a reaction where the amount of ssBP required matches the amount of ssDNA reactant. To calculate the adequate amount of ssBP, each of which can attach to 10 nucleotides[14-15], one has to know the reaction rate per template · sec for a specific nicking enzyme: polymerase ratio so that the amount of expected output can be calculated.

In short, the addition of ssBP to the reaction mixture can help eliminate the production of these undesired fragments, thereby yielding better experimental results. It might be possible to employ this method in an exponential amplification reaction[10] to produce a geometric amplification of oligomers longer than 20bp without having fragmentation problem. This combination may be a possible alternative to PCR.

In addition, the "ssBP vs. Hybridization" experiment shows that ssDNA oligonucleotides previously captured by ssBP molecules preferentially hybridize with their complementary strands rather than being attached to ssBP. This indicates that the structure of a double helix is a more stable than that of ssDNA attached to ssBP.

References

[1] Walker, G. T., et al. "Strand displacement amplification--an isothermal, in vitro DNA amplification technique." *Nucleic Acids Res.* 20 (1992): 1691–1696. Print.

[2] Walker, G. T., et al. "Isothermal in vitro amplification of DNA by a restriction enzyme/DNA polymerase system."*Proc. Natl. Acad. Sci. USA* 89 (1992): 392-396. Print.

[3] Innis, M. A., et al. "DNA sequencing with Thermus aquaticus DNA polymerase and direct sequencing of polymerase chain reaction-amplified DNA." *Proc. Natl. Acad. Sci. USA* 85 (1988): 9436-9440. Print.

[4] Saiki, R. K., et al. "Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia." *Science* 230 (1985): 1350–1354. Print.

[5] Mullis, K., et al. "Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction." *Cold Spring Harb. Symp. Quant. Biol.* 51 Pt 1 (1986): 263–273. Print.

[6] Saiki, R. K., et al. "Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase." *Science* 239 (1988): 487-491. Print.

[7] Fire, A. and Xu, S. Q. "Rolling replication of short DNA circles" *Proc. Natl. Acad. Sci. USA* 92 (1995): 4641–4645. Print.

[8] Liu, D., et al. "Rolling Circle DNA Synthesis: Small Circular Oligonucleotides as Efficient Templates for DNA Polymerases." *J. Am. Chem. Soc.* 118 (1996): 1587–1594. Print.

[9] Lizardi, P. M., et al. "Mutation detection and single-molecule counting using isothermal rolling-circle amplification." *Nat. Genet.* 19 (1998): 225–232. Print.

[10] Van Ness, J., Van Ness, L. K., Galas, D. J. "Isothermal reactions for the amplification of oligonucleotides." *Proc. Natl. Acad. Sci. USA* 100 (2003): 4504-4509. Print.

[11] Villalva, C., et al. "Increased Yield of PCR Products by Addition of T4 Gene 32 Protein to the SMART PCR cDNA Synthesis System." *Biotechniques* 31 (2001): 81-86. Print.

[12] Jeffries, D. and Farquharson, C. "Effects of choice of reverse-transcriptase enzyme and use of T4 gene 32 protein on banding patterns in agarose gel differential display." *Anal. Biochem.* 308 (2002): 192-194. Print.

[13] Rychlik, W., Spencer, W. J. and Rhoads, R. E. "Optimization of the annealing temperature for DNA amplification in vitro." *Nucleic Acids Res.* 18 (1990): 6409-6412. Print.

[14] Alberts, B. and Frey, L. "T4 Bacteriophage Gene 32: A Structural Protein in the Replication and Recombination of DNA." *Nature* 227 (1970): 1313-1318. Print.

[15] Kornberg, A. and Baker, T. *DNA Replication (2nd edition).* San Francisco: W. H. Freeman, 1991. 328. Print.

Chapter 3 - Feasibility of a Hopfield Neural Network using DNA molecules

Abstract

Adleman's 1994 proof that DNA oligomers using specific molecular reactions can be used to solve the Hamiltonian Path Problem suggested the possibility of massively parallel processing power, remarkable energy efficiency and compact data storage ability for this new type of computation. The Boolean architecture of the first DNA computers and the fact that DNA hybridization reactions can be error prone indicates that some form of fault tolerance or error correction would be beneficial in any large scale applications. In this study, we demonstrate the operation of a four dimensional Hopfield associative memory storing two memories as an archetype fault tolerant neural network implemented using DNA molecular reactions. The response of the network compares favorably to a computer simulation and suggests that the protocols could be scaled to a network of significantly larger dimensions.

3.1. Introduction

Adleman[1] first pointed out that Watson-Crick hybridization of pairs of complementary DNA strands makes possible a representation of highly parallel selective operations that is the key to the possible utility of DNA computation. Another architecture for molecular computing using DNA was proposed to show how to solve the famous "satisfaction" problem (SAT) by Lipton[2]. Several groups subsequently showed

that 4, 6 and 9 variable SAT problems can be solved using DNA computation[3-6]. As a landmark, a group led by Adleman[7] was able to solve a 20-variable 3-SAT problem, which is equivalent to finding the only correct solution among over a million possibilities, on a DNA computer. However, for all of the studies summarized so far, small departures from the ideal selectivity of DNA hybridization may eventually lead to significant undesired pairings of strands and thus to difficulties in schemes for implementing large Boolean functions using DNA.

Deaton et al.[8] showed that it should be possible to find a large enough set of mutually non-hybridizing DNA strands to in fact allow digital molecular computation of high complexity with tolerable errors. Another approach to fault-tolerant computation is the use of neural networks, since they don't need the high precision associated with digital computing. Mills, Yurke, and Platzman[9] suggested that neural networks in which the usual axons and neurons are replaced by the diffusion and molecular recognition of DNA might make practical use of the massive parallelism associated with simultaneous hybridization reactions. In this paper, we demonstrate an experimental implementation of an associative memory in the form of a Hopfield neural network[10] using DNA molecules. We note that even if large scale applications turn out to be infeasible, moderate sized DNA neural networks or DNA computers with other architectures could have useful applications to medical diagnostics where the input data might be in the form of DNA strands[11]. An associative or content addressable memory Hopfield network is a convenient benchmark because its properties are very well known and because it is

simply programmed in a single step by forming the sum of the outer products of its component memories[10].

3.2. Hopfield Neural Network and its DNA representation

The elements of memory in the Hopfield network[10] are represented as *d* component vectors $\vec{V} = \sum_{i=1}^{d} V_i \hat{e}_i$ in a space with basis vectors e_i (i=1, 2, …d). The items of memory, a set of vectors $\vec{V}^{(a)}$ (with a=1, 2, …s) representing different experiences, are stored in memory by summing the outer product matrices of the memory vectors:

$$T_{ij} = \sum_{a=1}^{s} V_i^{(a)} V_j^{(a)} \tag{3.1}$$

It has been reported that the removal of all diagonal components, so that $T_{ij} = 0$ for $i = j$, is required along with symmetry, $T_{ij} = T_{ji}$, for unconditional stability of the recall algorithm[12]. A particular experience $V_i^{(b)}$, imperfectly represented by a truncated "clue" vector $U_i^{(b)}=V_i^{(b)}$ for $i \leq q$ and $U_i^{(b)}=0$ for $i>q$, is recalled by iteration of the nonlinear equations[10]

$$X_i^{(1)} = S\left\{\sum_{j=1}^{m} T_{ij} U_j^{(b)}\right\} \tag{3.2a}$$

$$X_i^{(2)} = S\left\{\sum_{j=1}^{m} T_{ij} X_i^{(1)}\right\} \tag{3.2b}$$

and so forth. Here the function $S(x)$ is a saturating function such as $\tanh(\lambda x)$ acting separately on each component of its vector argument. Typically solutions to Eqs (3.2) are found with the small-signal gain λ being adjusted to facilitate convergence. If the stored experience vectors $V_i^{(a)}$ are sufficiently different, i.e. are part of a nearly orthogonal set, the system will settle into a state closely resembling $V_i^{(b)}$.

In order to implement a neural network in a DNA "language", we let each of the amplitudes of the components of the basis vectors be either +1 or -1 as represented by unit concentrations of a ssDNA oligomer or its complement respectively. Note that using the complementary strands to represent negative amplitudes[9] allows the net amplitude of a mixture of positive and negative amplitudes represented by complementary single stranded oligomers to be calculated by the DNA itself if our protocol allows for the removal of any resulting double-stranded DNA. In order to be able to successively perform Oliver's analog matrix algebra[13] using DNA, we must work with two different vector spaces, which will be represented by two independent sets of single stranded DNA (ssDNA) oligomers and their complements, $\{I_i\}$ and $\{O_i\}$, in 1:1 correspondence with the basis vectors $\hat{e}_i$ in the input and output vector spaces, respectively. The oligomers of the set $\{I_i\}$ are chosen to be minimally hybridizing with the set $\{O_i\}$ to prevent unwanted interactions between the inputs and outputs. Thus there is a pair of determined ssDNA oligos (input and output) assigned to each basis vector. Depending on whether $V_i^{(a)}$ is +1 or -1, the i^{th} component of the a^{th} vector $V_i^{(a)}$ will be represented by a specific ssDNA oligo or its complement, respectively. Thus, $\vec{V}^{(a)}$, representing an experience, is nothing

but a combination of various determined ssDNA strands each with the same concentration. In our specific representation we have chosen the $\{I_i\}$ and $\{O_i\}$ as 20 base pair (bp) and 40 bp ssDNA respectively.

Each component of the memory matrix constructed from an experience vector $\vec{V}^{(a)}$ is also a ssDNA oligomer that we produced by ligating the 5' end of the complement of an input ssDNA with the 3' end of the complement of an output ssDNA. For instance, the sequence representing $T_{23}^{(a)}$ is the same as that of $\bar{O}_2\bar{I}_3$ if both $V_2^{(a)}$ and $V_3^{(a)}$ are +1, where the bar means complement. In contrast to theory[12], we find that the diagonal elements of the memory matrix, such as $\bar{O}_1\bar{I}_1$, O_1I_1, $\bar{O}_2\bar{I}_2$, O_2I_2, etc. in DNA "language" do not have to be removed because they will not cause any problem. In fact, each diagonal component helps its corresponding input ssDNA to produce an equivalent version of itself in the output space. This can be conceptualized as one piece of a total experience recalling itself using the memory established by that piece. This is necessary for a small size experiment starting with only one input experience represented by one input ssDNA. However, the diagonal memory matrix oligomers may be removed if desired in an experiment starting with more than one input ssDNA, since an output ssDNA will be produced by other input strands. In our results shown below, we will demonstrate what difference occurs depending on whether or not the diagonal components are included in the memory matrix. Since we use two different sets of DNA oligos, I_i's and O_i's, the $T_{ij} = T_{ji}$ requirement cannot be satisfied literally in terms of the sequences of the DNA strands. However, operationally $T_{ij} = T_{ji}$ in the representation we

have chosen. Finally, our experimental implementation of the saturation function is effected by hybridizing the output oligomers with a standard concentration of their complementary strands, all of which are present in the same amount. This amount has to be chosen carefully for each experimental case and is always taken to be less than the amount of each output oligomer produced at the end. The saturation method used in this study is similar to the one described by Mills et al.[9, 14-15].

3.3. Materials and Methods

The experiences stored in our *d*=4 dimensional Hopfield memory are two different arbitrarily chosen 4-component vectors, 1: (+1, -1, +1, -1) and 2: (-1, -1, +1, +1). These vectors can be thought as two different 4-bit binary information sets. Note that the 1st and 4th basis vectors of both sets are different, whereas the 2nd and 3rd are the same. This design of vectors will not provide any particular advantage to our results except that helps us to study different cases. One might argue that the size of our 4-dimensional problem is very small, but it is in fact sufficient to allow us to take the first steps towards investigating the feasibility of implementing a Hopfield Neural Network using DNA molecules.

As described in the previous section, in DNA "language", if the value of the i^{th} basis vector is +1, a set of 20bp (I_i) and 40bp (O_i) oligos are employed to represent it in the input and output vector spaces, respectively. If the value of a basis vector is -1, the complementary ssDNA oligomers corresponding to this basis vector are used. For instance, I_1 and O_1 represent the 1st basis vector of the 1st vector, which is +1, on the other

hand, $\bar{I}_1$ and $\bar{O}_1$ are used to represent the 1st basis vector of the 2nd vector, which is -1. The explicit oligomers used are indicated by the sequences of randomly chosen ssDNA oligos synthesized by Integrated DNA Technologies (IDT) in Table 3.1. All strands are diluted with nuclease-free water (DEPC-Free) purchased from IDT. At the 5' and 3' ends of input or complementary input strands we have placed GG and CC, respectively, as a partial recognition site of a non-palindromic restriction enzyme, which permits a nicking operation to be performed during the Isothermal Linear Amplification (ILA)[16] step. The other 16 nucleotides found in the middle section of the 20 bp input strands are different for each oligo. Similarly, each 40bp output strand and its complement has TCAGC and GCTGA at the 5' and 3' ends, respectively, and the remaining 30 nucleotides in each strand are different. The enzyme designated Nt.BbvCI is used as the nicking enzyme and was purchased from New England Biolabs (NEB).

The memory matrix of each 4-component vector has to be constructed separately using the following method. The 5' end of each complementary DNA strand of a vector in input space is ligated with the 3' end of each complementary DNA strand of the same vector in the output space. For instance, the complementary sets of strands for the 1st vector (+1, -1, +1, -1) represented by (I_1, $\bar{I}_2$, I_3, $\bar{I}_4$) and (O_1, $\bar{O}_2$, O_3, $\bar{O}_4$), are ($\bar{I}_1$, I_2, $\bar{I}_3$, I_4) and ($\bar{O}_1$, O_2, $\bar{O}_3$, O_4) in the input and output spaces, respectively. If the ligation reaction described above is performed, $\bar{O}_1\bar{I}_1$, $\bar{O}_1I_2$, $\bar{O}_1\bar{I}_3$, $\bar{O}_1I_4$,…, O_4I_4 are produced as the components of the memory matrix corresponding to the 1st vector. The oligomers corresponding to the total memory matrix is equal to the combination of the oligomers for

the individual memory matrices. The set of operations described above is summarized in Table 3.2.

I_1:	5’- GGT AGC AAG CAT CAG TGT CC -3’
$\bar{I}_1$:	5'- GGA CAC TGA TGC TTG CTA CC -3'
I_2:	5'- GGT CAG TAT CAG ACA TCG CC -3'
$\bar{I}_2$:	5'- GGC GAT GTC TGA TAC TGA CC -3'
I_3:	5'- GGC AAG TCT CAG GTA AGT CC -3'
$\bar{I}_3$:	5'- GGA CTT ACC TGA GAC TTG CC -3'
I_4:	5'- GGT TAG CGA CTC AGT TAG CC -3'
$\bar{I}_4$:	5'- GGC TAA CTG AGT CGC TAA CC -3'
O_1:	5'- TCA GCA AGC TAC CGT TGA GTC CGT ATG TCA TCT GAG CTG A -3'
$\bar{O}_1$:	5'- TCA GCT CAG ATG ACA TAC GGA CTC AAC GGT AGC TTG CTG A -3'
O_2:	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG A -3'
$\bar{O}_2$:	5'- TCA GCG ATC TTC CTA AGC GTG TGA TGT GAC TCG AAG CTG A -3'
O_3:	5'- TCA GCA CTG ACA CGA TCA TCT GGA TAC TGA GAC GAG CTG A -3'
$\bar{O}_3$:	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG A -3'
O_4:	5'- TCA GCG TAG ACC ACA GAC TCT CAG ATC GTA CTC AAG CTG A -3'
$\bar{O}_4$:	5'- TCA GCT TGA GTA CGA TCT GAG AGT CTG TGG TCT ACG CTG A -3'

Table 3.1: Input (20bp) and Output (40bp) oligomers with their corresponding complementary strands. Each basis vector in the Hopfield neural network is represented by one input and one output oligomers.

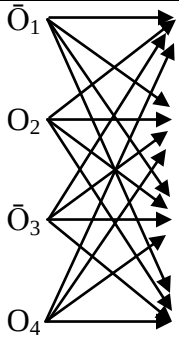
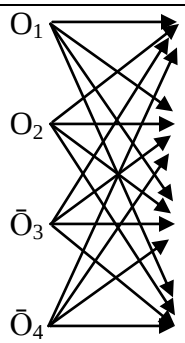
Vectors used in this study	DNA form of vectors in input and output spaces	Complementary DNA strands	Memory matrix building operation	Memory matrix oligos
(+1,-1,+1,-1)	$(I_1,\bar{I}_2,I_3,\bar{I}_4)$ $(O_1,\bar{O}_2,O_3,\bar{O}_4)$	$(\bar{I}_1,I_2,\bar{I}_3,I_4)$ $(\bar{O}_1,O_2,\bar{O}_3,O_4)$		$\bar{O}_1\bar{I}_1,\bar{O}_1I_2,\bar{O}_1\bar{I}_3,\bar{O}_1I_4,$ $O_2\bar{I}_1,\mathbf{O_2I_2,O_2\bar{I}_3},O_2I_4,$ $\bar{O}_3\bar{I}_1,\mathbf{\bar{O}_3I_2,\bar{O}_3\bar{I}_3},\bar{O}_3I_4,$ $O_4\bar{I}_1,O_4I_2,O_4\bar{I}_3,O_4I_4$
(-1,-1,+1,+1)	$(\bar{I}_1,\bar{I}_2,I_3,I_4)$ $(\bar{O}_1,\bar{O}_2,O_3,O_4)$	$(I_1,I_2,\bar{I}_3,\bar{I}_4)$ $(O_1,O_2,\bar{O}_3,\bar{O}_4)$		$O_1I_1,O_1I_2,O_1\bar{I}_3,O_1\bar{I}_4,$ $O_2I_1,\mathbf{O_2I_2,O_2\bar{I}_3},O_2\bar{I}_4,$ $\bar{O}_3I_1,\mathbf{\bar{O}_3I_2,\bar{O}_3\bar{I}_3},\bar{O}_3\bar{I}_4,$ $\bar{O}_4I_1,\bar{O}_4I_2,\bar{O}_4\bar{I}_3,\ \bar{O}_4\bar{I}_4$

Table 3.2: A schematic description summarizes how to produce the memory matrix, which is equivalent to the combination of two separate memory matrices. There are 28 different oligomers in the memory matrix. Highlighted oligomers are common to both sets.

Although it is possible to perform the ligation reaction experimentally, we preferred to order the ligated forms (Table 3.3) from IDT-DNA for convenience in this small dimensional problem. Since the number of oligomers needed to represent the memory matrix grows as the square of the dimension, eventually this option will not be practical for larger problems.

$\bar{O}_1\bar{I}_1$	5'- TCA GCT CAG ATG ACA TAC GGA CTC AAC GGT AGC TTG CTG AGG ACA CTG ATG CTT GCT ACC -3'
$\bar{O}_1I_2$	5'- TCA GCT CAG ATG ACA TAC GGA CTC AAC GGT AGC TTG CTG AGG TCA GTA TCA GAC ATC GCC -3'
$\bar{O}_1\bar{I}_3$	5'- TCA GCT CAG ATG ACA TAC GGA CTC AAC GGT AGC TTG CTG AGG ACT TAC CTG AGA CTT GCC -3'
$\bar{O}_1I_4$	5'- TCA GCT CAG ATG ACA TAC GGA CTC AAC GGT AGC TTG CTG AGG TTA GCG ACT CAG TTA GCC -3'
$O_2\bar{I}_1$	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG AGG ACA CTG ATG CTT GCT ACC -3'
O_2I_2	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG AGG TCA GTA TCA GAC ATC GCC -3'
$O_2\bar{I}_3$	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG AGG ACT TAC CTG AGA CTT GCC -3'
O_2I_4	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG AGG TTA GCG ACT CAG TTA GCC -3'
$\bar{O}_3\bar{I}_1$	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG AGG ACA CTG ATG CTT GCT ACC -3'
$\bar{O}_3I_2$	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG AGG TCA GTA TCA GAC ATC GCC -3'

$\bar{O}_3\bar{I}_3$	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG AGG ACT TAC CTG AGA CTT GCC -3'
$\bar{O}_3I_4$	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG AGG TTA GCG ACT CAG TTA GCC -3'
$O_4\bar{I}_1$	5'- TCA GCG TAG ACC ACA GAC TCT CAG ATC GTA CTC AAG CTG AGG ACA CTG ATG CTT GCT ACC -3'
O_4I_2	5'- TCA GCG TAG ACC ACA GAC TCT CAG ATC GTA CTC AAG CTG AGG TCA GTA TCA GAC ATC GCC -3'
$O_4\bar{I}_3$	5'- TCA GCG TAG ACC ACA GAC TCT CAG ATC GTA CTC AAG CTG AGG ACT TAC CTG AGA CTT GCC -3'
O_4I_4	5'- TCA GCG TAG ACC ACA GAC TCT CAG ATC GTA CTC AAG CTG AGG TTA GCG ACT CAG TTA GCC -3'
O_1I_1	5'- TCA GCA AGC TAC CGT TGA GTC CGT ATG TCA TCT GAG CTG AGG TAG CAA GCA TCA GTG TCC -3'
O_1I_2	5'- TCA GCA AGC TAC CGT TGA GTC CGT ATG TCA TCT GAG CTG AGG TCA GTA TCA GAC ATC GCC -3'
$O_1\bar{I}_3$	5'- TCA GCA AGC TAC CGT TGA GTC CGT ATG TCA TCT GAG CTG AGG ACT TAC CTG AGA CTT GCC -3'
$O_1\bar{I}_4$	5'- TCA GCA AGC TAC CGT TGA GTC CGT ATG TCA TCT GAG CTG AGG CTA ACT GAG TCG CTA ACC -3'

O_2I_1	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG AGG TAG CAA GCA TCA GTG TCC -3'
$O_2\bar{I}_4$	5'- TCA GCT TCG AGT CAC ATC ACA CGC TTA GGA AGA TCG CTG AGG CTA ACT GAG TCG CTA ACC -3'
$\bar{O}_3I_1$	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG AGG TAG CAA GCA TCA GTG TCC -3'
$\bar{O}_3\bar{I}_4$	5'- TCA GCT CGT CTC AGT ATC CAG ATG ATC GTG TCA GTG CTG AGG CTA ACT GAG TCG CTA ACC -3'
$\bar{O}_4I_1$	5'- TCA GCT TGA GTA CGA TCT GAG AGT CTG TGG TCT ACG CTG AGG TAG CAA GCA TCA GTG TCC -3'
$\bar{O}_4I_2$	5'- TCA GCT TGA GTA CGA TCT GAG AGT CTG TGG TCT ACG CTG AGG TCA GTA TCA GAC ATC GCC -3'
$\bar{O}_4\bar{I}_3$	5'- TCA GCT TGA GTA CGA TCT GAG AGT CTG TGG TCT ACG CTG AGG ACT TAC CTG AGA CTT GCC -3
$\bar{O}_4\bar{I}_4$	5'- TCA GCT TGA GTA CGA TCT GAG AGT CTG TGG TCT ACG CTG AGG CTA ACT GAG TCG CTA ACC -3'

Table 3.3: The 28 different 60bp ssDNA oligomers comprising the memory matrix.

In order to perform DNA molecular experiments without complications, such as cross-hybridization of memory oligomers, we have to divide the 28 different 60bp ssDNA oligomers representing the memory into four different groups according to their

types (Table 3.4). Each group, consisting of a partial memory matrix, is stored in different vials and used separately in the experiments.

Type	Memory matrix oligos
O_iI_j	O_1I_1, O_1I_2, O_2I_1, O_2I_2, O_2I_4, O_4I_2, O_4I_4
$\bar{O}_iI_j$	$\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_3I_1$, $\bar{O}_3I_2$, $\bar{O}_3I_4$, $\bar{O}_4I_1$, $\bar{O}_4I_2$
$O_i\bar{I}_j$	$O_1\bar{I}_3$, $O_1\bar{I}_4$, $O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_4$, $O_4\bar{I}_1$, $O_4\bar{I}_3$
$\bar{O}_i\bar{I}_j$	$\bar{O}_1\bar{I}_1$, $\bar{O}_1\bar{I}_3$, $\bar{O}_3\bar{I}_1$, $\bar{O}_3\bar{I}_3$, $\bar{O}_3\bar{I}_4$, $\bar{O}_4\bar{I}_3$, $\bar{O}_4\bar{I}_4$

Table 3.4: The weight memory matrix oligos are divided into 4 groups according to their types to prevent unwanted hybridization reactions among themselves. There are 7 oligos in each group.

We can now start using the memory matrix to operate on the clue vectors. As described in Section II, a clue vector is a truncated version of one of the memory vectors, with some zero elements replacing the vector entries. For instance, (+1, 0, 0, 0) is our first clue vector obtained by truncation of the first memory vector (+1, -1, +1, -1). Mathematically our clue vector components may be thought of as duobinary digits since there are three possible values 0 and ±1. Chemically the clue vector is a mixture of ssDNA oligos including I_1 only. There are 3 other clue vectors used in this study: (0, -1, 0, 0), (+1, 0, 0, -1), (0, -1, 0, +1). The first operation of the memory matrix on a clue vector is to add a definite amount of input oligos representing the clue vector into a sample of the memory matrix solutions stored in the 4 different vials. If the clue input strands are complementary to some of the memory matrix oligomers in a vial, Watson

Crick hybridization will take place. Otherwise, there will be no reaction. If hybridized molecules, containing partially dsDNA, are formed in any of the 4 vials, they will be extended using Klenow Fragment (3' → 5' exo) polymerase in the next step so that fully dsDNA molecules will be formed. T4 Gene 32 single stranded binding protein is also added into the reaction mix to guarantee that the extension reaction properly progresses. Both enzymes were obtained from NEB. A visual description of the molecular reactions described so far is displayed in Figure 3.1.

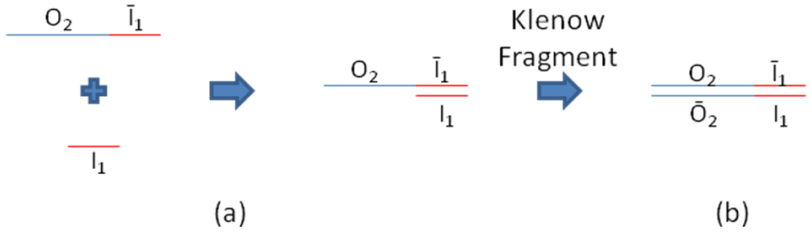


Figure 3.1: (a) $O_2\bar{I}_1$, one of the memory matrix oligos found in the 3rd test vial hybridizes with I_1, indicating the first clue vector, which yields a partially dsDNA structure. (b) Klenow Fragment (3' → 5' exo) extends the lower strand and produces a fully dsDNA structure.

After fully dsDNA oligomers are formed in different vials, we purify each solution by the QuickStep™2 PCR Purification Kit (Single Cartridges-from Edge Systems). All unhybridized input and memory matrix ssDNA oligos, other impurity strands if there are any, Klenow Fragment polymerase and buffer will be removed during this step so that there will be nothing but fully dsDNA remaining in our solutions at the end. Then, we transfer a portion of each solution into a new vial and start producing the

output strands using the ILA (Isothermal Linear Amplification) reaction (Figure 3.2). During this reaction the following steps will occur: (1) Nt.BbvCI (nicking enzyme) cuts the previously extended lower strands of dsDNA oligomers at a point where the extension was started. (2) Klenow Fragment polymerase starts extending the lower strands one more time while it also displaces the previously extended strands which are the output oligomers. (3) Steps 1 and 2 are continuously being repeated so that more output strands of each kind are produced. It is important to note that, similar to what we did in the extension reaction, T4 Gene 32 single stranded binding (ssb) protein is added into the ILA reaction mix to ensure that undesired fragment strands causing difficulty in further steps are not produced[17]. In order to stop the ILA reaction, we heat-denature the polymerase, ssb protein, and nicking enzyme. This action also helps to release ssDNA output strands bound/retained by single-stranded binding protein so that they can be seen on a gel electrophoresis image during the read-out step. Depending on which clue input strands are used, an output strand and its complement can be produced at the same time. In such a case, these output strands hybridize with each other and yield 40bp dsDNA oligomers that have to be removed by gel extraction. Otherwise, the gel extraction is not necessary. Finally, a small amount of the final solution, including the net output strands, will be transferred into 8 different vials, each of which contains a definite amount of one of to output ssDNA oligomers O_1, $\bar{O}_1$, O_2, $\bar{O}_2$, O_3, $\bar{O}_3$, O_4, or $\bar{O}_4$. This definite amount has to be less than the amount of each output oligomer, since it will then act like the saturation function that is described in the theory section above. In a vial, if one of the net output strands finds its complement, the hybridization reaction yields a 40bp dsDNA

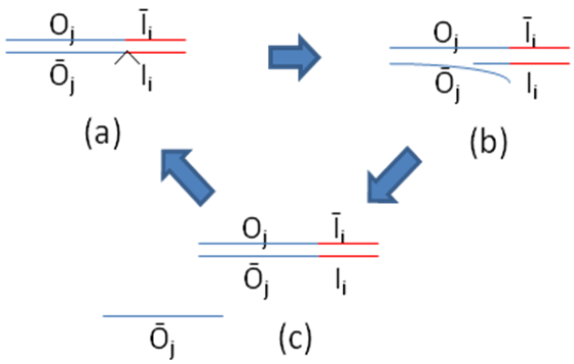


Figure 3.2: Visual description of a cycle of the ILA reaction. (a) dsDNA oligo , representing the double helix form of a memory matrix strand, is cut on its lower strand by Nt.BBvCI, a non-palindromic restriction enzyme. The cut is made at a point where the extension was started in the previous step. (b) Klenow Fragment polymerase starts extending the lower strand while an output strand starts being displaced. (c) The original dsDNA template and an output strand are produced at the end.

oligo whose presence can be determined by the gel electrophoresis method. This read out process will give us the final result, an approximation to the item of experience associated with the given imperfect clue.

As briefly discussed in the Section II, we would also like to demonstrate the difference between the following two cases: the diagonal oligos (O_iIi , $\bar{O}_i\bar{I}_i$) are (1) included or (2) not included in the memory matrix. In order to study this, we operate on 2 different sets of memory matrices, (1) without diagonal (WOD) oligos and (2) with diagonal (WD) oligos, by the first clue vector (+1, 0, 0, 0). However, for the cases starting with other 3 clue vectors, we always work with the memory matrix including the diagonal oligomers. Reactions between the memory matrix oligomers and 4 different clue vectors are summarized in Table 3.5.

EXP #	Clue vector Corresponding input strands Corresponding memory vector	Memory matrix oligos initially stored in 4 different vials until fully dsDNA molecules are produced by extension	Output strands produced at the end of ILA reaction starting with all dsDNA oligos combined in a single vial	Saturated Net output strands Corresponding output vector
1	(+1, 0, 0, 0) I_1 **(+1, -1, +1, -1)**	O_1I_2, O_2I_1, O_2I_4, O_4I_2	-	$\bar{O}_2$, O_3, $\bar{O}_4$ **(0, -1, +1, -1)**
		$\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_3I_1$, $\bar{O}_3I_2$, $\bar{O}_3I_4$, $\bar{O}_4I_1$, $\bar{O}_4I_2$	-	
		$O_1\bar{I}_3$, $O_1\bar{I}_4$, $O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_4$, $O_4\bar{I}_1$, $O_4\bar{I}_3$	$\bar{O}_2$, $\bar{O}_4$	
		$\bar{O}_1\bar{I}_3$, $\bar{O}_3\bar{I}_1$, $\bar{O}_3\bar{I}_4$, $\bar{O}_4\bar{I}_3$	O_3	
2	(+1, 0, 0, 0) I_1 **(+1, -1, +1, -1)**	O_1I_1, O_1I_2, O_2I_1, O_2I_2, O_2I_4, O_4I_2, O_4I_4	-	O_1, $\bar{O}_2$, O_3, $\bar{O}_4$ **(+1, -1, +1, -1)**
		$\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_3I_1$, $\bar{O}_3I_2$, $\bar{O}_3I_4$, $\bar{O}_4I_1$, $\bar{O}_4I_2$	-	
		$O_1\bar{I}_3$, $O_1\bar{I}_4$, $O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_4$, $O_4\bar{I}_1$, $O_4\bar{I}_3$	$\bar{O}_2$, $\bar{O}_4$	
		$\bar{O}_1\bar{I}_1$, $\bar{O}_1\bar{I}_3$, $\bar{O}_3\bar{I}_1$, $\bar{O}_3\bar{I}_3$, $\bar{O}_3\bar{I}_4$, $\bar{O}_4\bar{I}_3$, $\bar{O}_4\bar{I}_4$	O_1, O_3	
3	(0, -1, 0, 0) $\bar{I}_2$ **(+1, -1, +1, -1)** **or** **(-1, -1, +1, +1)**	O_1I_1, O_1I_2, O_2I_1, O_2I_2, O_2I_4, O_4I_2, O_4I_4	$\bar{O}_1$, $\bar{O}_2$, $\bar{O}_4$	$\bar{O}_2$, O_3 **(+0, -1, +1, 0)**
		$\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_3I_1$, $\bar{O}_3I_2$, $\bar{O}_3I_4$, $\bar{O}_4I_1$, $\bar{O}_4I_2$	O_1, O_3, O_4	
		$O_1\bar{I}_3$, $O_1\bar{I}_4$, $O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_4$, $O_4\bar{I}_1$, $O_4\bar{I}_3$	-	
		$\bar{O}_1\bar{I}_1$, $\bar{O}_1\bar{I}_3$, $\bar{O}_3\bar{I}_1$, $\bar{O}_3\bar{I}_3$, $\bar{O}_3\bar{I}_4$, $\bar{O}_4\bar{I}_3$, $\bar{O}_4\bar{I}_4$	-	
4	(+1, 0, 0, -1) I_1 & $\bar{I}_4$ **(+1, -1, +1, -1)**	O_1I_1, O_1I_2, O_2I_1, O_2I_2, O_2I_4, O_4I_2, O_4I_4	$\bar{O}_2$, $\bar{O}_4$	O_1, $\bar{O}_2$, O_3, $\bar{O}_4$ **(+1, -1, +1, -1)**
		$\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_3I_1$, $\bar{O}_3I_2$, $\bar{O}_3I_4$, $\bar{O}_4I_1$, $\bar{O}_4I_2$	O_1, O_3	
		$O_1\bar{I}_3$, $O_1\bar{I}_4$, $O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_4$, $O_4\bar{I}_1$, $O_4\bar{I}_3$	$\bar{O}_2$, $\bar{O}_4$	
		$\bar{O}_1\bar{I}_1$, $\bar{O}_1\bar{I}_3$, $\bar{O}_3\bar{I}_1$, $\bar{O}_3\bar{I}_3$, $\bar{O}_3\bar{I}_4$, $\bar{O}_4\bar{I}_3$, $\bar{O}_4\bar{I}_4$	O_1, O_3	

5	(0, -1, 0, +1) $\bar{I}_2$ & I_4 **(-1, -1, +1, +1)**	O_1I_1, O_1I_2, O_2I_1, O_2I_2, O_2I_4, O_4I_2, O_4I_4	*$\bar{O}_1$*, $\bar{O}_2$, *$\bar{O}_4$*	$\bar{O}_1$, $\bar{O}_2$, O_3, O_4 **(-1, -1 +1, +1)**
		$\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_3I_1$, $\bar{O}_3I_2$, $\bar{O}_3I_4$, $\bar{O}_4I_1$, $\bar{O}_4I_2$	*O_1*, O_3, *O_4*	
		$O_1\bar{I}_3$, $O_1\bar{I}_4$, $O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_4$, $O_4\bar{I}_1$, $O_4\bar{I}_3$	$\bar{O}_1$, $\bar{O}_2$	
		$\bar{O}_1\bar{I}_1$, $\bar{O}_1\bar{I}_3$, $\bar{O}_3\bar{I}_1$, $\bar{O}_3\bar{I}_3$, $\bar{O}_3\bar{I}_4$, $\bar{O}_4\bar{I}_3$, $\bar{O}_4\bar{I}_4$	O_3, O_4	

Table 3.5: (a) Five cases are experimentally studied. (b) There are 4 different types of query vectors. (c) Two different sets of the memory matrix, (1) not including or (2) not including diagonal oligos, are used. (d) All of the output strands that are produced at the end of ILA reactions for each case are shown in the table. If two output products which are complements of each other (ones written by italic font), then they are canceled out as a result of the hybridization reaction. (e) The remaining output strands are saturated so that the net output strands are found.

3.4. Details of the protocol for recall of memory

First, the clue input strands are operated on (hybridized with) the memory matrix oligomers and the output strands are produced as a result of extension reaction. Then, the dsDNA molecules produced at the end of previous step are purified by a gel extraction protocol. Next, the output strands were amplified by Isothermal Linear Amplification (ILA) method. Depending on which strands are used as input strands, an output strand and its complementary might be produced at the same time. In such cases, one more gel extraction process is necessary to separate the net output strands from other undesired DNA oligos found in the solution. Finally, purified net output strands obtained at the end are hybridized with each output strand or its complement to be able to determine what these net output strands are. The details of experimental protocols used in this study for 5 different cases are explained below in the same order as they were performed.

(i) Hybridization between input and memory matrix oligomers. This step is performed in 4 different vials for each case summarized in Table 3.5. For instance, we place 1 µl of each 50 µM 40bp OI type of memory matrix oligomers into the first vial, thus, there will be 4 µl of memory matrix oligomers in Vial-1 for the 1st case. Note that the memory matrix of the first case doesn't include the diagonal oligomers. However, for cases 2-5, there are 7 of µl memory matrix oligomers (OI type) in Vial-1, including diagonal ones. In a similar manner, 3 more vials containing other types of memory oligomers ($\bar{O}I$, $O\bar{I}$ and $\bar{O}\bar{I}$) are prepared for each case. Then, we add 3 µl (in case 1) or 4 µl (in cases 2-5) of each 50 µM 20bp input oligomer depending on the clue vector in to vials. 10 µl 10xT4 DNA Ligase Buffer (500 mM Tris-HCl, 100 mM $MgCl_2$, 10 mM ATP, 100 mM Dithiothreitol, pH 7.5 @25°C, from NEB) containing 0.1M NaCl is also put into each reaction mix to improve the efficiency of hybridization. Finally, just enough nuclease-free water (NFW), varying for different cases, to make the total volumes equal to 100 µl is added. All vials are incubated at 40.1 °C, which is chosen as the average annealing temperature for 1 hour. After the incubation, we get a partially dsDNA structure, 60bp strand on one side with 20bp on the other. In order to remove the high salt buffer we run each sample contained in different vials through Performa® Gel Filtration matrix (from Edge Bio Systems), which efficiently removes >99.9% of salts, buffers, dNTPs and other low molecular weight materials, by centrifuging at 2900 rpm for 2 minutes.

(ii) Extension reaction. We continue doing experiments for each case in 4 different vials at this step. The extension reaction solution is prepared in the following order. 37.5 µl

NFW, 2.5 µl a type of partially dsDNA oligo (4 different types for each case) from earlier step, 5 µl 10xNEBuffer 2 (500 mM NaCl, 100 mM Tris-HCl, 100 mM $MgCl_2$, 10 mM Dithiothreitol pH 7.9 @ 25°C), 2 µl dNTP (from NEB), 2 µl 10mg/ml T4 Gene 32 single-strand binding protein and 1 µl 5000units/ml Klenow fragment polymerase are mixed in a vial. The mixes are incubated at 37 °C for 30 minutes. At the end, 60bp fully dsDNA oligos are created.

(iii) Purification of extended strands. In order to remove all fragment DNA oligomers (unhybridized or unextended memory matrix and input oligomers), enzymes (polymerase, nicking and single-stranded binding protein), buffer and dNTPs we run our samples through QuickStep™2 PCR Purification Kit (Single Cartridges-from Edge Bio Systems). Simply, SOPE™ (Solid-phase Oligo Protein Elimination) Resin that binds and efficiently removes primers, ssDNA, enzymes and other proteins is added in to each sample. The volume amount of SOPE used for a sample is about one half of the corresponding sample's volume. Mixes are vortexed for 30 seconds, then allowed to sit at room temperature for approximately 3 minutes. Finally, mixes are transferred into different Performa® Gel Filtration vials and centrifuged at 2900 rpm for 2 minutes. Pure 60bp dsDNA solutions are obtained as the product.

(iv) Isothermal Linear Amplification (ILA). This step is needed to increase the quantities of output strands to the point where they can be detected on an electrophoresis gel. At this step, we combine all 60bp dsDNA oligos produced in 4 different vials for each sample together in a single vial. The ILA reaction mix is prepared in the following

order for each sample. 8.375 µl of each of 4 solutions containing 60bp dsDNA, 5 µl 10xNEBuffer 2, 3 µl dNTP (from NEB), 5 µl 10mg/ml T4 Gene 32 single-strand binding protein, 2.5 µl 10000units/ml Nt.BbvCI the nicking enzyme and 1 µl 5000units/ml Klenow fragment polymerase are mixed in a vial, then, incubated at 37 °C for 90 minutes. In reality, depending on how many output strands we would like to produce and other factors, the reaction time and the amount of ss-binding protein may vary for different cases. Preferred amounts used in this study are chosen based on our experiences to achieve successful results. In addition, reaction mixes are incubated 80 °C for 20 minutes to heat-deactivate enzymes so that the ILA completely stops and output DNA strands captured by ss-binding proteins are freed.

(v) Extraction. Gel extraction of the target DNA is required for cases 3 and 5, but not for others, since some output strands and their complements are produced as shown in Table 3.5. These 40bp dsDNA oligomers produced as a result of hybridization at the end of ILA reaction have to be removed from solution because they can produce confusion during the read out process in the next step. This necessity can also be understood as the need to purify the net output 40bp ssDNA strands by a suitable method. We prefer following the standard QIAEX® II gel extraction procedure (by QIAGEN). The diffusion coefficient can be obtained by the Nernst-Einstein relation

$$D = \mu kT/q \tag{3.3}$$

where μ, k, T and q are the mobility, Boltzmann's constant, the temperature and the charge, respectively. In this experiment, the length of DNA that we want to extract is 40

bases, so $\mu \approx 6.2*10^{-5}$ $cm^2/V.s$, $kT \approx 1/37$ eV at 50 °C and $q \approx 40e$. Thus, $D \approx 3*10^{-8}$ cm^2/s and the root mean square of diffusion length, $L_{rms} = (Dt)^{1/2}$, for $t = 1$ hour would be about 0.1 mm. Since the size of gel chunks after grinding is smaller than 0.1 mm, DNA oligomers in the gel chunks can easily diffuse out with the help of the elution buffer (0.5M ammonium acetate, 10mM magnesium acetate, 1mM EDTA, pH 8.0, 0.1 SDS-prepared in our lab) with incubating at 50 °C for an hour.

(vi) Read-out by gel electrophoresis. In this final step, for each experimental case, we transfer 4 µl of final solution containing the net output strands in 8 different vials, each of which already contains .5 µl 2 µM of one of 8 different ssDNA oligomers (O_1, $\bar{O}_1$, O_2, $\bar{O}_2$, O_3, $\bar{O}_3$, O_4 or $\bar{O}_4$). In order to increase the hybridization probability between complementary strands we also add 0.5 µl T4 DNA Ligase Buffer in each vial. Each output strand finds its complement and hybridizes with it during a one hour incubation period at 51 °C. As a result, 40bp dsDNA oligomers are produced in some of the vials for each case. In order to read out this information, we load the 8 resulting samples (1 µl of Bromophenol blue is added to each one) for each case on a different 0.75mm 10 well 10% polyacrylamide gel (produced in our lab) and begin running the electrophoresis using the Mini-PROTEAN® Tetra Cell vertical electrophoresis tank (Bio-Rad). 1x Tris Borate-EDTA (TBE) buffer (Sigma-Aldrich) is used as the running buffer. After electrophoresis, SYBR Gold (Invitrogen) nucleic acid gel stain, which is the most sensitive fluorescent stain available for detecting double or single stranded DNA molecules, is used to stain DNA oligomers. The polyacrylamide gel with DNA is soaked in a stain buffer, which is a mixture of 5 µl of 10000x SYBR Gold gel stain and 100ml 1x

TBE buffer, for 20 minutes at room temperature. Finally, we take images of the gels containing DNA strands stained by SYBR Gold under a 312 nm UV light (Fisher Scientific). The excitation maxima for this type of stain is ~300nm. The exposure time is 5 seconds for all images. The results are analyzed by using Kodak 1D image analysis system (Fisher Scientific).

3.5. Results and Discussion

The gel image results of "hybridization", "extension" and "purification of extended strands" reactions for all 5 experimental cases are seen in Figure 3.3. All three reactions are performed in 4 different vials, depending on the type of memory matrix strands (OI's, OĪ's, ŌI's or ŌĪ's), for each case. Lanes 1-4 of Fig. 3.3a indicates the results of the "hybridization", "extension" before or after heat-deactivation, and "purification of extended strands" steps, respectively, using I_1 as the input strand of the first case and OI's, the first type of memory matrix strands. More specifically, bands seen in lane 1 of Fig. 3.3a represent the OI type of 60bp ssDNA memory matrix strands, none of which hybridizes with I_1, as expected. Thus, there are no 60bp dsDNA strands shown in lanes 2 or 3, showing the results of the "extension" procedure before or after the heat deactivation, respectively. It is important to note that all DNA molecules (not fully double stranded ones) are captured by single-stranded binding protein that is added to the solution during "extension". Since the size of single-stranded binding protein molecules is larger than the holes in 10% polyacrylamide gel, the ssb protein molecules are unable to enter the solution. The band at the top of lane 2 represents the single-stranded binding

proteins attached to non-fully dsDNA strands. However, heating denatures the ssb proteins, and therefore these DNA strands are released and can be observed again at their normal positions on a gel image as shown in lane 3. Finally, all DNA molecules are removed from the solution during the "purification" step since none of them is fully double stranded. This is the reason for the lack of signal in lane 4 of Fig. 3.3a. Similarly, lanes 5-8 of Fig. 3.3a, lanes 1-4 of Fig. 3.3b, and lanes 5-8 of Fig. 3.3b summarize the results of three reactions between I_1 (the input strand for the first experimental case) and other types of memory matrix strands, $O\bar{I}$'s, $\bar{O}I$'s or $\bar{O}\bar{I}$'s, respectively. As expected, there is also nothing left in lane 8 of Fig. 3.3a while there is a band corresponding to 60bp dsDNA in both lanes 4 and 8 of Fig. 3.3b. The results of the other 4 cases are shown in Fig. 3.3c&3.3d, Fig. 3.3e&3.3f, Fig. 3.3g&3.3h, and Fig. 3.3i&3.3j. One can notice that there are some left over DNA strands shown in lanes 4 and 8 of Fig. 3.3c and lane 8 of Fig. 3.3f even though there shouldn't be any. However, this is not anomalous, since these strands are in all likelihood the same as some of the memory matrix strands. Unlike the first case, some memory matrix strands couldn't be removed from our solutions at the end of the "purification" step since the amount of memory matrix strands for these cases is more than that of the first case. We can either run these solutions through the same "purification" method one more time to take out all left over strands, or use them directly in the ILA reaction (which is the next step). These strands will not affect the progress of the ILA. The second option is used in this study. Other than this minor detail, all other results shown in lanes 4 and 8 of the other images correspond to our expectations based on theory.

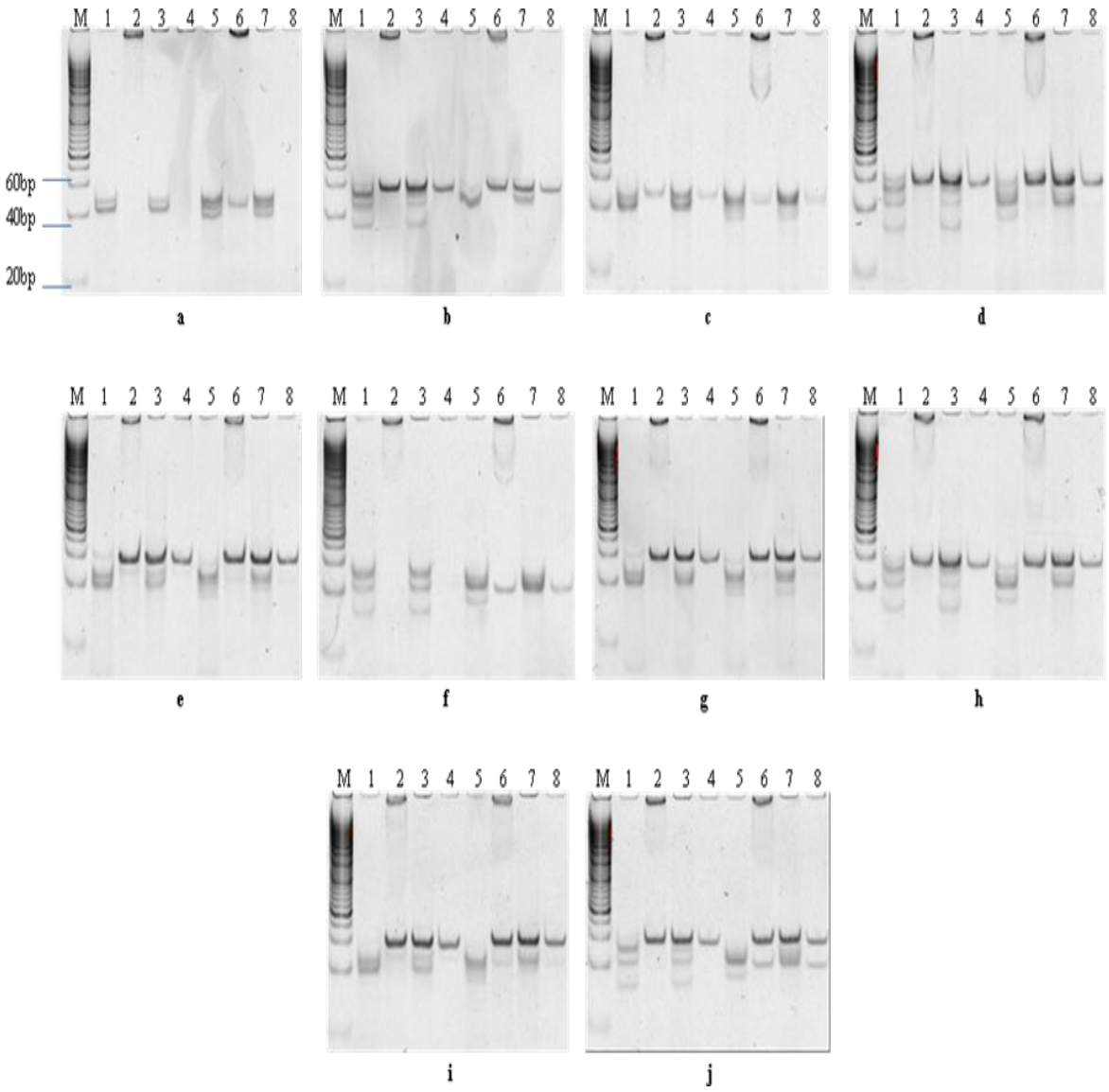


Figure 3.3: Gel images obtained at the end of "hybridization", "extension", and "purification of extended strands" reactions. (a) Lanes 1-4, (a) Lanes 5-8, (b) Lanes 1-4, and (b) Lanes 5-8 indicate the results of "hybridization", "extension" before or after heat deactivation, and "purification of extended strands" reactions between I_1 the input strand representing the first clue vector and 4 different types of memory matrix strands, OI's, O$\bar{I}$'s, $\bar{O}$I's and $\bar{O}\bar{I}$'s, respectively. In a similar manner, (c)&(d), (e)&(f), (g)&(h), and (i)&(j) pairs corresponds to the results of other 4 experimental cases. Lane M of each image contains 20bp dsDNA marker. 5 µl samples are shown in each lane.

The results of the ILA reaction for all 5 experimental cases are shown in Figure 3.4. Lanes 1&2, 3&4, and 5&6 of Fig. 3.4a shows the content of solutions before & after the heat deactivation for cases 1, 2, and 4, respectively. Very dim bands appear at the 60bp dsDNA level in lanes represent the 60bp dsDNA that is used as the template in ILA reactions. 40bp ssDNA output strands are seen between 20bp and 40bp dsDNA marks in lanes 2, 4, and 6. It should be noted that an output strand and its complement aren't generated at the same time for any of these 3 cases since there are no bands representing 40bp dsDNA. Thus, there is a need to extract output strands. However, for cases 3 and 5, the results of which are shown in lanes 1&2 of Fig. 3.4b&3.4c, respectively, there are bands seen at 40bp dsDNA level. This indicates that one or more output strands and their complements are obtained at the end of ILA reaction. Another reason why we know bands appearing at the 40bp dsDNA level in lanes 2 of Fig. 3.4b&3.4c are actually double stranded is because they can't be captured by ss-binding protein, thus, they also appear in lane 1of both images representing the result of ILA reaction before heat-deactivation. So, we need to operate the "extraction" process for these cases. Simply, the solutions that we have after the heat deactivation are loaded on lanes 3-6 of two different 10% polyacrylamide gels (as shown in Fig. 3.4b&3.4c) and the thick bands found between 20 and 40bp dsDNA markers, representing the net output 40bp ssDNA strands, are extracted from both gels. In addition, there are some fragment strands with very slow mobility for cases 3, 4, and 5, however, we aren't very concerned about them since they don't interfere with output strands, which are much faster in the gel.

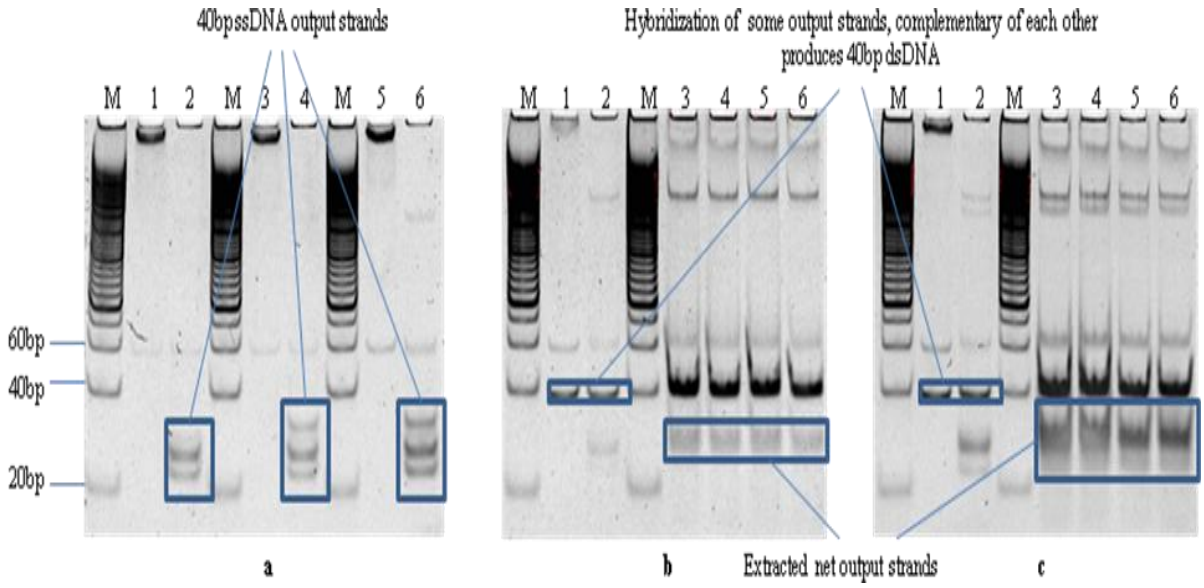


Figure 3.4: Gel images display the results of ILA reactions for all five experimental cases. (a) Lanes 1&2, 3&4, and 5&6 represents the results before & after heat deactivation for cases 1, 2, and 4, respectively. (b) Lanes 1&2 show the results of ILA reaction for case 3. Lanes 3-6 is the gel extraction image for the same case. (c) Lanes 1&2 show the results of ILA reaction for case 5. Lanes 3-6 is the gel extraction image for the same case.

The gel images obtained at the end of "read-out" reactions for all five experimental cases are presented in Figure 3.5. The first case, which employs (+1, 0, 0, 0) as the clue vector represented by I_1 input strand and the memory matrix without diagonal strands, produces 3 different output strands each of which can hybridize with one of O_2,$\bar{O}_3$, and O_4, and consequently produce 40bp dsDNA strands as seen in lanes 3,6 and 7 of Fig. 3.5b. This implies that our output strands are $\bar{O}_2$, O_3, $\bar{O}_4$. Image analysis by Kodak 1D software shows that the light intensities in arbitrary of 40bp dsDNA bands in lanes 3, 6, and 7 are 284373, 404615, and 323721, respectively. The normalized light intensities over the average value are 0.84, 1.20, and 0.95. By considering that neither O_1 nor $\bar{O}_1$ are

produced as one of the output strands, our results correspond with (0, -0.84, +1.20, -0.95) which is comparable with (0, -1, +1, -1), expected result for the first case shown in Table 3.5. This shows that the query vector can successfully recall its corresponding memory vector (+1, -1, -1, +1), except the first term, but doesn't recall the other vector present in solution (-1, -1, +1, +1). On the other hand, 40bp dsDNA bands are seen in lanes 2, 3, 6, and 7 of Fig. 3.5b presenting the result of the second experimental case that is the same as case 1 except diagonal strands are also included memory matrix solution. This indicates that the output strands are O_1, $\bar{O}_2$, O_3, $\bar{O}_4$. The corresponding normalized light intensities are 1.13, 0.80, 1.19, 0.88, which can be understood as (+1.13, -0.80, +1.19, -0.88) in vector representation. This is another comparable result with (+1, -1, +1, -1) the first memory vector. By combining the two results, we conclude that the reason why the first term is not remembered in case 1 but is in case 2, is because $\bar{O}_1\bar{I}_1$, one of the diagonal memory matrix strands, is not included in the memory matrix solution in case 1 but is in case 2. As a result, we prove that an input strand, if it clearly corresponds with one of the memory vectors, i.e. I_1 corresponds with (+1, -1, +1 , -1) but not (-1, -1, +1, +1), recalls its corresponding memory vector completely only if the diagonal memory strands are included in memory matrix solution in a single iteration. In case 3, we use $\bar{I}_2$ representing (0, -1, 0, 0) as the clue vector and the memory matrix including diagonal strands, which can't distinguish if (+1, -1, +1, -1) or (-1, -1, +1, +1) is the correct recall. All 8 lanes of Fig. 3.5c have dim 40bp dsDNA bands, two of which (in lanes 3 and 6) are darker than others. This result indicates that, most likely, some 40bp dsDNA strands,

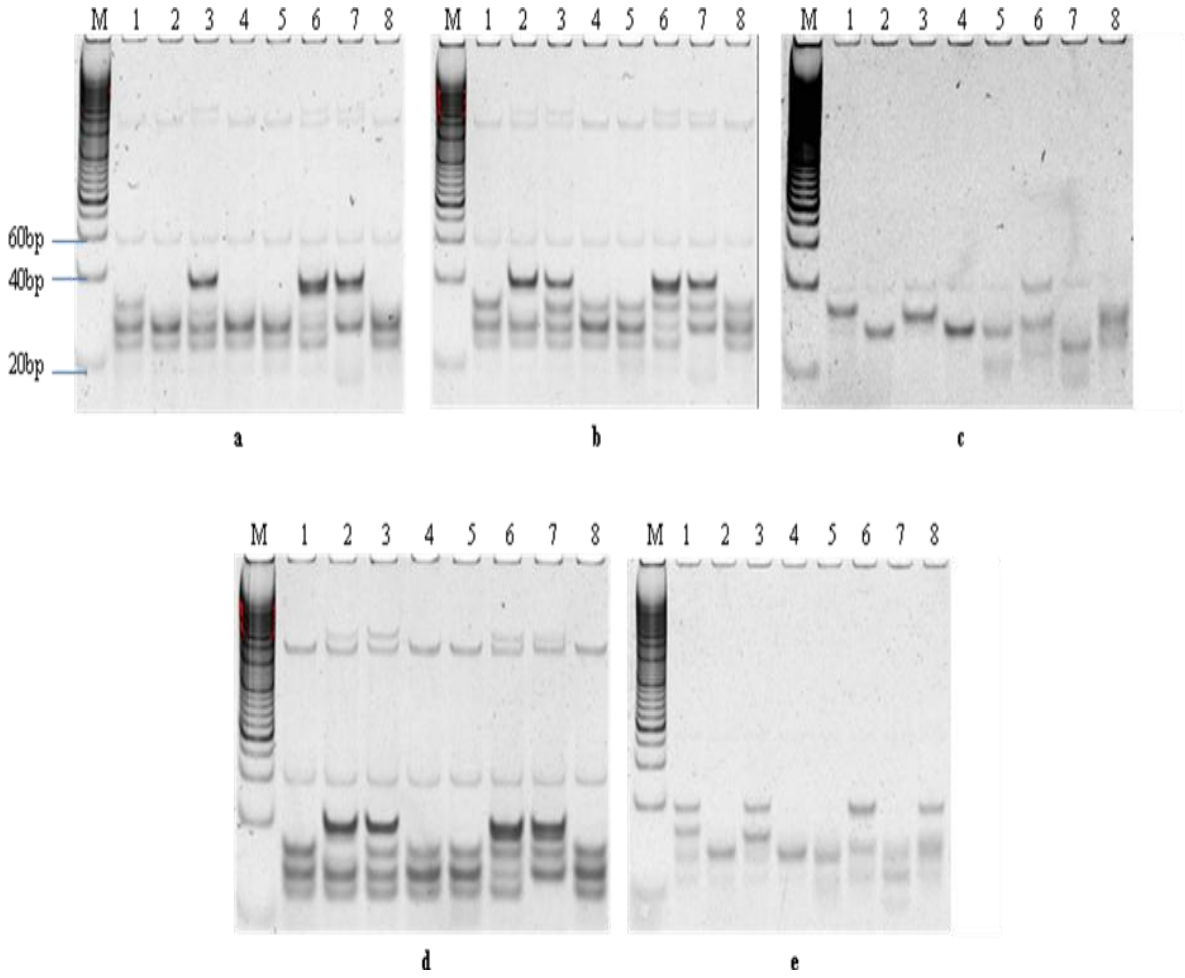


Figure 3.5: The "read-out" gel images for 5 different experimental cases. Lanes 1-8 of each image represent the hybridization between output strands and O_1, $\bar{O}_1$, O_2, $\bar{O}_2$, O_3, $\bar{O}_3$, O_4 or $\bar{O}_4$, respectively. If there is a hybridization reaction, then 40bp dsDNA strands will be produced. For instance, the 40bp dsDNA band in lane 3 of the first image states that one of the output strands of the first experimental case is complementary of O_2, thus, this strand is $\bar{O}_2$. By following the same logic, we can conclude that the output strands are (a) O_2, $\bar{O}_3$, O_4, (b) $\bar{O}_1$, O_2, $\bar{O}_3$, O_4, (c) O_2, $\bar{O}_3$, (d) $\bar{O}_1$, O_2, $\bar{O}_3$, O_4, and (e) O_1, O_2, $\bar{O}_3$, $\bar{O}_4$ for cases 1-5, respectively.

which are produced at the end of the "extension" procedure, are also extracted along with the net output strands during the "extraction" as a result of experimental error. This

actually explains why we have 40bp dsDNA bands with the same low light intensity in lanes 1, 2, 4, 5, 7, and 8 despite the fact that there shouldn't be any (Table 3.5). So, we subtract this low intensity from the light intensities of darker bands in lanes 3 and 6 to obtain the net intensities. After normalization, we get 0.37 and 1.63 corresponding to the number of net output strands $\bar{O}_2$, O_3, respectively. Although the experimental result (0, -0.37, +1.63, 0) is quite different from the expected (0, -1, +1, 0) result (Table 3.5) in terms of magnitudes, they still have the most important common point, which is the fact that only $\bar{O}_2$, O_3 strands have to be found in the final solution. The conclusion of this experimental case is that an input strand that can't distinguish two memory vectors, i.e. $\bar{I}_2$ representing -1 as the second element of a memory vector is common between (+1, -1, +1, -1) and (-1, -1, +1, +1) memory vectors, can only recall the common terms but not others. The 3 experimental cases discussed so far can be thought as a bit simple and extreme since only one input strand is employed in each one, but they are useful in proving the characteristics of an experimental implementation of a Hopfield neural network. The overall outcome of 3 cases is that a query vector consisting one bit of information, represented by an input strand, is enough to recall its corresponding memory vector among others only if the one bit information in the query vector is not common between memory vectors. In addition, unlike what is suggested by theory, having the diagonal strands in the memory matrix solution doesn't cause a problem on results, in fact, they have to be included to be able to make a complete recall in cases where only one input strand and only one iteration are employed.

The results of the fourth and fifth cases, each of which employs a query vector consisting of two nonzero elements, are shown in Fig. 3.5d&3.5e. There are 40bp dsDNA bands in lanes 2, 3, 6, and 7 of Fig. 3.5d, the results of case 4 using (+1, 0, 0, -1) as the query vector. This proves that O_1, $\bar{O}_2$, O_3, and $\bar{O}_4$ are the output strands. The normalized light intensities of each band are assumed to be proportional to amounts of output strands are 1.04, 0.77, 1.13, and 0.95. The vector representation of our result, (+1.04, -0.77, +1.13, -0.95), is comparable with the expected one (+1, -1, +1, -1). On the other hand, 40bp dsDNA bands are evident in lanes 1, 3, 6, and 8 of Fig 3.5e, the results of the fifth case employing (0, -1, 0, +1) as the query vector. This implies that $\bar{O}_1$, $\bar{O}_2$, O_3, and O_4 are the output strands. The normalized light intensities are 0.86, 0.80, 1.34, and 1.00. The vector representation of this result, (-0.86, -0.80, +1.34, +1.00), is comparable with the expected one (-1, -1, +1, +1).

We believe that error between our results and the expected results in terms of numerical magnitudes, except for the third case (63%), is in a reasonable range (0-35%) if various experimental errors, such as the lack of precision due to pipetting errors, possible imperfect hybridization mistakes, partial loss of sample between experimental procedures etc., are considered. Further improvements to perfect the current experimental procedure or using alternative methods are expected to increase the precision of the results.

3.6. Confirmation by Simulation

Since the oligomers used in this experiment are designed to effect computation akin to a Hopfield Neural Network (HNN), it might at first appear that simulation of this experiment *in silico* involves the construction of a memory matrix like that prescribed by Hopfield – the sum of the outer product matrices formed by each vector with itself. However, this is not accurate, since in our representation DNA does not add like a scalar. For example, consider the formation of our memory matrix for the vectors given previously. The two outer product matrices would appear as in Figure 3.6, and their components as they would be summed in the four separate vials are given in Table 3.5.

$$\begin{pmatrix} o1bari1bar & o1bari2 & o1bari3bar & o1bari4 \\ o2i1bar & o2i2 & o2i3bar & o2i4 \\ o3bari1bar & o3bari2 & o3bari3bar & o3bari4 \\ o4i1bar & o4i2 & o4i3bar & o4i4 \end{pmatrix}$$

$$\begin{pmatrix} o1i1 & o1i2 & o1i3bar & o1i4bar \\ o2i1 & o2i2 & o2i3bar & o2i4bar \\ o3bari1 & o3bari2 & o3bari3bar & o3bari4bar \\ o4bari1 & o4bari2 & o4bari3bar & o4bari4bar \end{pmatrix}$$

Figure 3.6: General memory matrices for the DNA vectors used.

Unlike for the usual model for a HNN, the two outer product DNA memory matrices must not be simply added together due to the likelihood of cross hybridization, and any attempt to model the DNA network without taking this into account will give incorrect predictions. Instead, the memory matrix is preserved as four components (O_iI_j, $O_iI_j$bar, $O_i$barI_j $O_i$bar$I_j$bar) which will act separately on the query before adding the results together, at which point strands representing opposite polarities can hybridize and thus cancel in whole or in part. For Experiment #1, it was also necessary to remove the

diagonal components of the memory matrix to match the experiments. Further, duplicates of certain degenerate matrix oligomers were not included in the memory solutions experimentally, so our simulation was run with this constraint in place as well.

For example, consider Experiment #2 above. In this experiment, the clue vector consisted of

```
{i1, 0, 0, 0}
```

To determine the output of the HNN on this clue vector in DNA:

1) Search for the complement to each clue entry by examining the "right hand side" of each memory oligomer for the complement to the clue entry. In this case, the first column of the first matrix is the only set of entries that have the complement ("i1bar").
2) Determine the complement to the "left hand side" of those oligomers which match. In this situation, the left hand sides of the matching oligomers are o1bar, o2, o3bar, and o4. Therefore the complements are o1, o2bar, o3, and o4bar.

These are the outputs from the HNN implemented in DNA. In order to do this more efficiently, we wrote a Mathematica code which allows the entry of arbitrary vectors, constructs the appropriate set of memory matrices, then considers an input clue and generates the appropriate output.

For each of the experiments above we obtained outputs for both simulations and experiments indicated in Figure 3.7. Experimental data is not shown for presaturation values, since different vector components represented by ssDNA cannot be distinguished from each other in gel electrophoresis.

3.7. Conclusion

In this paper, we present the experimental implementation of a small scale Hopfield neural network using DNA molecules for the first time. Although the sizes of the memory vectors are small, they are big enough to study different cases to reveal the important details of this implementation. A series of experimental procedures followed here is capable of producing results which are compatible with expected ones. However, further improvements can be made to improve the effectiveness and precision. In the future, we are thinking of adding an extra procedure, in which the ligated forms of input and output strands representing the memory matrix oligomers can be produced in the lab instead of directly ordering them from IDT. This will reduce the cost of the experiment and allow us to be able to study larger sized problems. In addition, the individual input and output DNA oligomers could be produced by PCR amplification of single random oligomers as described in[18]. Our proposed method can be used to study different types of problems. For instance, two black & white binary images can be transferred into vector forms. By following our experimental procedure, a memory matrix in terms of DNA strands could be constructed. A small portion of one of the images could be interpreted as the query vector and we could operate on the memory matrix oligomers by using the

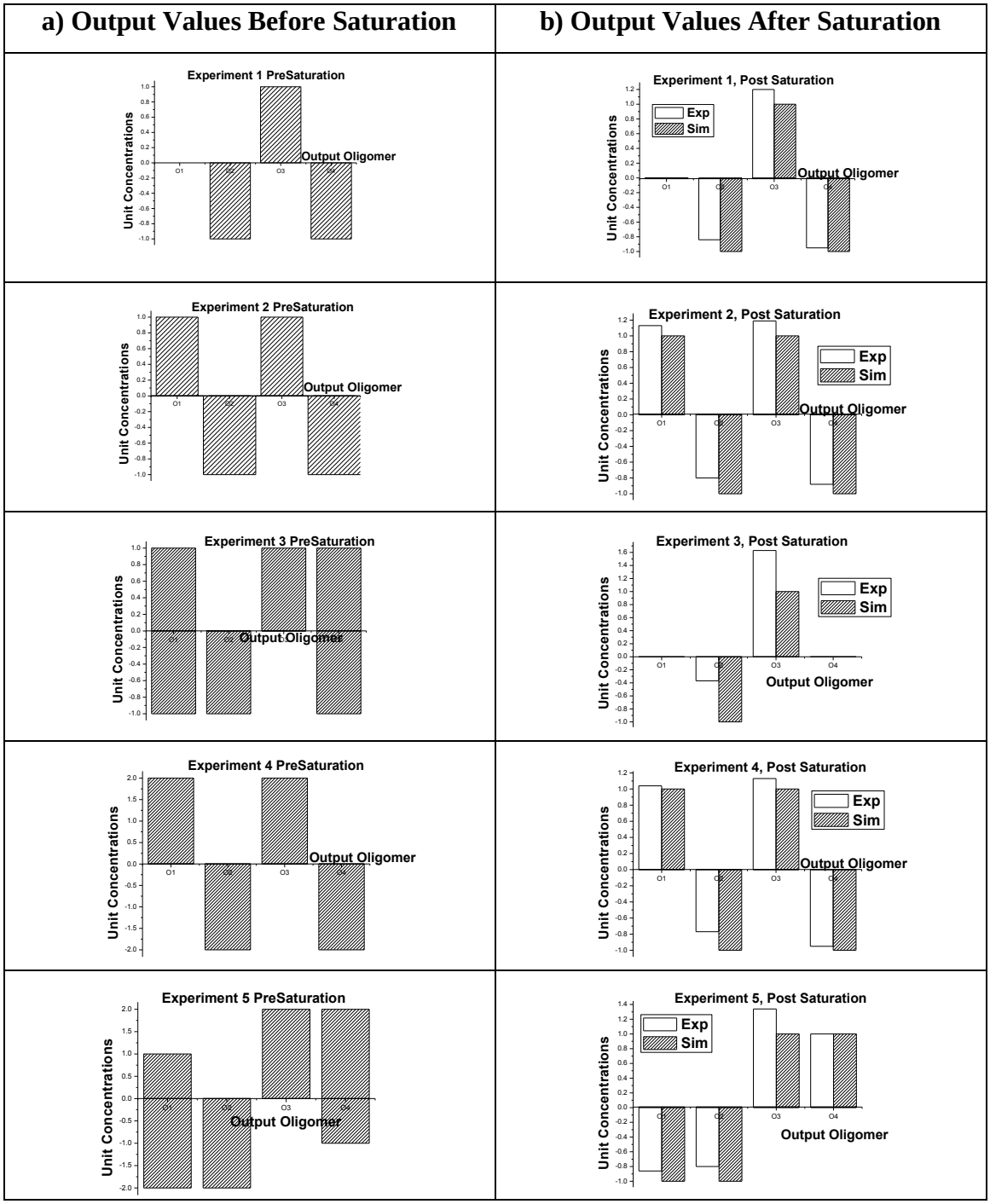


Figure 3.7: Experimental and Computational Results for DNA Vector Outputs a) After ILA reaction (Simulated). Experimental results not shown as mentioned in text above figure and b) After saturation (Simulated and Experimental Results).

input strands corresponding to this query vector. At the experiment's conclusion, the correct image should be recalled. The idea of this future project has been already proposed[15], however, the outline of the experimental procedure suggested there has been modified in the light of experience for this study. Finally, an extended version of the experimental model described here might be used to implement similar types of neural networks or other mathematical problems.

References

[1] Adleman, L. M. "Molecular computation of solutions to combinatorial problems." *Science* 266 (1994): 1021–1024. Print.

[2] Lipton, R. J. "DNA solution of hard computational problems." *Science* 268 (1995): 542–545. Print.

[3] Liu, Q., et al. "DNA Computing on Surfaces." *Nature* 403 (2000): 175-179. Print.

[4] Yoshida, H. and Suyama, A. "Solution to 3-SAT by Breadth First Search." *DNA Based Computers V,* Eds. Winfree, E. and Gifford, D. K. Providence, RI: American Mathematical Society, 1999. 9-20. Print.

[5] Sakamoto, K., et al. "Molecular Computation by DNA hairpin formation." *Science* 288 (2000): 1223-1226. Print.

[6] Faulhammer, D., et al. "Molecular Computation: RNA solutions to chess problems." *Proc. Natl. Acad. Sci. U.S.A.* 97 (2000): 1385-1389. Print.

[7] Braich, R. S., et al. "Solution of a 20-variable 3-SAT problem on a DNA computer." *Science* 296 (2002): 499–502. Print.

[8] Deaton, R., et al. "Reliability and efficiency of a DNA-based computation." *Phys. Rev. Lett.* 80 (1998): 417–420. Print.

[9] Mills Jr., A. P., Yurke, B., and Platzman, P. M. "Article for analog vector algebra computation." *BioSystems* 2 (1999): 175-180. Print.

[10] Hopfield, J. J. "Neural networks and physical systems with emergent collective computational abilities." *Proc. Natl. Acad. Sci. USA* 79 (1982): 2554–2558. Print.

[11] Mills Jr., A. P. "Gene expression profiling diagnosis through DNA molecular computation." *Trends in Biotechnology* 20 (2002): 137-140. Print.

[12] Cohen, M. A., and Grossberg, S. G. "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks." *IEEE Trans. Syst. Man Cybern*. 13 (1983): 815–826. Print.

[13] Oliver, J. S. "Matrix multiplication with DNA." *J. Molec. Evol.* 45 (1997): 161–167. Print.

[14] Mills Jr., A. P., Yurke, B., and Platzman, P. M. "DNA analog vector algebra and physical constraints on large-scale DNA-based neural network computation." *DNA Based Computers V*. Eds. Winfree, E. and Gifford, D. K. Providence, RI: American Mathematical Society, 2000. 65-73. Print.

[15] Mills Jr., A. P. "Experimental aspects of DNA neural network computation." *Soft Computing* 5 (2001): 10-18. Print.

[16] Van Ness, J., Van Ness, L. K., and Galas, D. J. "Isothermal reactions for the amplification of oligonucleotides." *Proc. Mat. Acad. Sci. USA* 100 (2003): 4504-4509. Print.

[17] Karabay, D., Hughes, B. S. T. and Mills Jr., A. P. "Application of T4 Gene 32 SSB Protein to the Improvement of Isothermal Linear Amplification (ILA) of DNA." In Preparation.

[18] Wang, S., et al. "Production of random DNA oligomers for scalable DNA computing." *Biotechnol. J.* 4 (2009): 119–128. Print.

Chapter 4 – Implementation of a 6-bit Hopfield Neural Network using DNA molecules

4.1. Introduction

Our previous study[1] presented in Chapter 3 demonstrated the feasibility of Hopfield Neural Network using DNA molecules. In that study, we skipped the ligation reaction step by which the memory matrix oligomers could be generated. Instead, those ssDNA molecules were synthesized by Integrated DNA Technologies. This is not an efficient way if the fact that there are n^2 memory matrix oligomers for each n-bit memory is taken into consideration. For example, a Hopfield Neural Network[2] designed to store 20-bit memories has more than 400 memory matrix oligomers. Due to a high cost factor, it is not very convenient to order all those molecules instead of generating them from input and output strands. Therefore, the scalability of DNA Based Hopfield Neural Network wasn't actually fully justified.

In this chapter, we demonstrate the results of a new experimental protocol starting with the ligation reaction. In addition, in place of reading out the final result on polyacrylamide gels at the end, a Heller Electronic Microarray/DNA cartridge (Nanogen) is used for this job[3-4]. This new protocol has a potential to handle with bigger size problems. Except for the modifications stated above, the rest of the experimental protocol and the theory behind it are the same as the previous one as outlined in Chapter 3.

4.2. Materials and Methods:

Two randomly chosen two black & white images (Figure 4.1), each of which is composed of 6 pixels, are stored in our *d*=6 dimensional Hopfield memory as the experiences. These images can be thought as two different 6-bit binary information sets. The 3rd, 4th, 5th, and 6th pixels of both images are different, whereas the 1st and 2nd are the same. This design of images will not provide any particular advantage to our results.

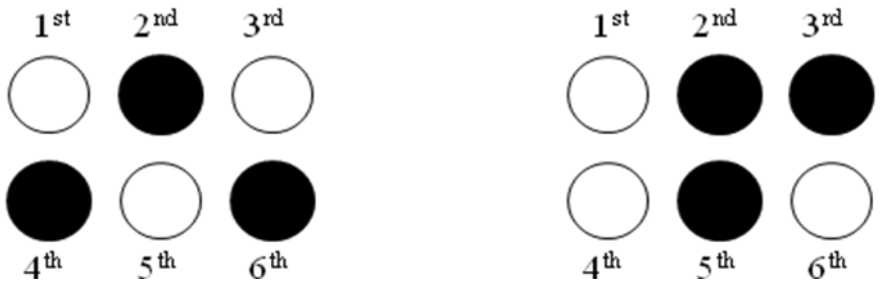


Figure 4.1: Two 6-pixel binary images used as experience to build a Hopfield associative memory.

A different set of 40bp (I_i) and 60bp (O_i) oligomers are employed to represent each white pixel in the input and output vector spaces, respectively. If the color of a pixel is black, the complementary ssDNA oligomers corresponding to this pixel are used. For instance, I_1 and O_1 represent the 1st pixel of both images, which is white, on the other hand, $\bar{I}_2$ and $\bar{O}_2$ are used to represent the 2nd pixel of both images, which is black.

At the 5’ and 3’ ends of both input and complementary input strands we have placed GGA GCA CCT G and CAG GTG CTC C, respectively, so that ligating every input with output strands by the same 20bp linker could be possible. Note that first and last two nucleotides (GG and CC) of invariant 10-mers at the 5’ and 3’ ends are a partial recognition site of a non-palindromic restriction enzyme (Nt.BbvCI from New England Biolabs, Cat#R0632S, 10 units/μl), which performs a nicking operation during the Isothermal Linear Amplification (ILA)[5] reaction. The other 20 nucleotides found in the middle section of 40 bp input strands are different for each oligo. Since each base pair is chosen from the four bases A, G, C, T, the number of different possible sequences is $\sim 10^{12}$. Considering that the size of our Hopfield memory (d=6) is much smaller than this number, we believe that our input strands have minimal overlap among themselves. On the other hand, each 60bp output strand or its complement has TCA GCA GTT GGT GGC AGG AG and CTC CTG CCA CCA ACT GCT GA at the 5’ and 3’ ends, respectively, and the remaining 20 nucleotides in the middle of each strand are different. Unlike the design of input strands, we had to place not just 10bp (enough for ligation reaction to happen) but 20bp identical sequences at the ends of each output strand so that the same Bitoin-primer and TET-primer ssDNA could hybridize with all output strands. Bitoin-primer and TET-primer are also 20-mers and necessary to be able to get the final read out on Electronic Microarray. The explicit oligomers used are indicated by the sequences of randomly chosen ssDNA synthesized by Integrated DNA Technologies (IDT) in Table 4.1. All strands are diluted with Nuclease-free water (DEPC-Free) purchased from IDT.

I_1:	5’- GGA GCA CCT GGC AGG TCT AAC ACC GAC ATT CAG GTG CTC C -3’
$\bar{I}_1$:	5'- GGA GCA CCT GAA TGT CGG TGT TAG ACC TGC CAG GTG CTC C -3'
I_2:	5'- GGA GCA CCT GAG GTA CTT GAA CGA TGC GAC CAG GTG CTC C -3'
$\bar{I}_2$:	5'- GGA GCA CCT GGT CGC ATC GTT CAA GTA CCT CAG GTG CTC C -3'
I_3:	5'- GGA GCA CCT GTA TGA GAC GCT CTG CTA GAG CAG GTG CTC C -3'
$\bar{I}_3$:	5'- GGA GCA CCT GCT CTA GCA GAG CGT CTC ATA CAG GTG CTC C -3'
I_4:	5'- GGA GCA CCT GGG CTC AGA CAA TAC ATC ACG CAG GTG CTC C -3'
$\bar{I}_4$:	5'- GGA GCA CCT GCG TGA TGT ATT GTC TGA GCC CAG GTG CTC C -3'
I_5:	5'- GGA GCA CCT GGA CAG TTA GTG CAT ACG GCA CAG GTG CTC C -3'
$\bar{I}_5$:	5'- GGA GCA CCT GTG CCG TAT GCA CTA ACT GTC CAG GTG CTC C -3'
I_6:	5'- GGA GCA CCT GAC ATG ACT GCT GTC ATG AGT CAG GTG CTC C -3'
$\bar{I}_6$:	5'- GGA GCA CCT GAC TCA TGA CAG CAG TCA TGT CAG GTG CTC C -3'
O_1:	5'- TCA GCA GTT GGT GGC AGG AGA CCG TTG AGT CCG TAT GTC ACT CCT GCC ACC AAC TGC TGA -3'
$\bar{O}_1$:	5'- TCA GCA GTT GGT GGC AGG AGT GAC ATA CGG ACT CAA CGG TCT CCT GCC ACC AAC TGC TGA -3'
O_2:	5'- TCA GCA GTT GGT GGC AGG AGG TCA CAT CAC ACG CTT AGG ACT CCT GCC ACC AAC TGC TGA -3'
$\bar{O}_2$:	5'- TCA GCA GTT GGT GGC AGG AGT CCT AAG CGT GTG ATG TGA CCT CCT GCC ACC AAC TGC TGA -3'

O_3:	5'- TCA GCA GTT GGT GGC AGG AGC ACG ATC ATC TGG ATA CTG ACT CCT GCC ACC AAC TGC TGA -3'
$\bar{O}_3$:	5'- TCA GCA GTT GGT GGC AGG AGT CAG TAT CCA GAT GAT CGT GCT CCT GCC ACC AAC TGC TGA -3'
O_4:	5'- TCA GCA GTT GGT GGC AGG AGC CAC AGA CTC TCA GAT CGT ACT CCT GCC ACC AAC TGC TGA -3'
$\bar{O}_4$:	5'- TCA GCA GTT GGT GGC AGG AGT ACG ATC TGA GAG TCT GTG GCT CCT GCC ACC AAC TGC TGA -3'
O_5:	5'- TCA GCA GTT GGT GGC AGG AGT TGT GAA CAA TGC TGC CGA CCT CCT GCC ACC AAC TGC TGA -3'
$\bar{O}_5$:	5'- TCA GCA GTT GGT GGC AGG AGG TCG GCA GCA TTG TTC ACA ACT CCT GCC ACC AAC TGC TGA -3'
O_6:	5'- TCA GCA GTT GGT GGC AGG AGC TAT GAG TCA TAG CGA TGA GCT CCT GCC ACC AAC TGC TGA -3'
$\bar{O}_6$:	5'- TCA GCA GTT GGT GGC AGG AGC TCA TCG CTA TGA CTC ATA GCT CCT GCC ACC AAC TGC TGA -3'
Linker:	5’- CAG GTG CTC CTCA GCA GTT GG –3’
Biotin-primer	5’-/Biosg/ TCA GCA GTT GGT GGC AGG AG -3’
TET-Primer	5’-/TET/TCA GCA GTT GGT GGC AGG AG -3’

Table 4.1: Input (40bp) and Output (60bp) oligomers with their corresponding complementary strands as well as Linker, Biotin-primer, and TET-primer ssDNA. Each basis vector in the Hopfield neural network is represented by one input and one output oligomer.

The memory matrix of each 6-pixel image has to be constructed separately using the following method. First, opposite DNA strands of an image in input space are divided into two groups and contained in different vials according to their types (non-complementary or complementary), and the 5' end of those oligomers are phosphorylated by T4 Polynucleotide Kinase[6-7] (New England Biolabs, Cat#M0201S, 10units/μl). Without phosphorylation, the phosphodiester bond can't be formed during the ligation reaction[8]. Then, the 5' end of each opposite DNA strand of an image in input space is ligated with the 3' end of each opposite DNA strand (also divided into two groups according to their types) of the same image in the output space. For instance, the opposite sets of strands for the 1st image are ($\bar{I}_1$, I_2, $\bar{I}_3$, I_4, $\bar{I}_5$, I_6) and ($\bar{O}_1$, O_2, $\bar{O}_3$, O_4, $\bar{O}_5$, O_6) in the input and output spaces, respectively. First, 5' ends of (I_2, I_4, I_6) and ($\bar{I}_1$, $\bar{I}_3$, $\bar{I}_5$) oligomers contained in vials D1 and D2 are phosphorylated. Then, each set of input strands is ligated with each set of (O_2, O_4, O_6) and ($\bar{O}_1$, $\bar{O}_3$, $\bar{O}_5$) using 20bp linker. When the ligation reaction described above is performed, (O_2I_2, O_2I_4, O_2I_6,…, O_6I_6), ($\bar{O}_1I_2$, $\bar{O}_1I_4$, $\bar{O}_1I_6$,..., $\bar{O}_5I_6$), ($O_2\bar{I}_1$, $O_2\bar{I}_3$, $O_2\bar{I}_5$,..., $O_6\bar{I}_5$), and ($\bar{O}_1\bar{I}_1$, $\bar{O}_1\bar{I}_3$, $\bar{O}_1\bar{I}_5$,…, $\bar{O}_5\bar{I}_5$) are produced in 4 different vials as the components of the memory matrix corresponding to the 1st image. Note that each memory matrix oligomer is a 100mer attached to 20bp linker as a partial lower strand. Since there are undesired hybridizations, such as a linker hybridized with only one of two types of strand (input or output), also occur during ligation reaction, we must separate desired ligated strands from others to prevent the confusion that will arise in the following steps by the gel extraction method. After the memory matrix oligomers are obtained for both images, the same type of oligomers (such as OI's for image 1 & 2)

generated from different images are mixed together. Abiding by the outlined method is necessary to perform DNA molecular experiments without complications, such as cross-hybridization of memory oligomers. The oligomers corresponding to the total memory matrix is equal to the combination of the oligomers for the individual memory matrices. The set of operations described above is summarized in Table 4.2.

We can now start operating on the memory matrix by using the input strands that corresponds to a query image. Similar to the description given in Chapter 3, a query image is a truncated version of one of the original memory images, with some unknown pixels. For instance, Figure 4.2 shows our query image (?, B, W, B, ?, ?) obtained by truncation of the first memory image (W, B, W, B, W, B). Mathematically the components of a query image may be thought of as duobinary digits since there are three possible values W, B, and "?". Chemically our query image is a mixture of ssDNA oligos including $\bar{I}_2$, I_3, and $\bar{I}_4$.

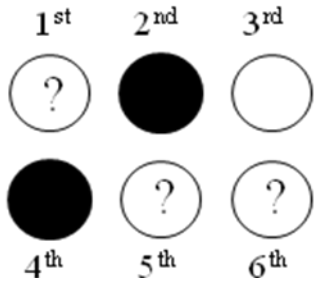


Figure 4.2: The query image with 3 unknown pixels designed to recall the 1st image.

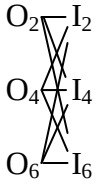
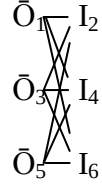
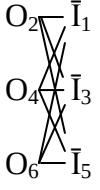
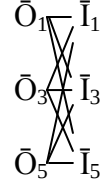
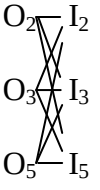
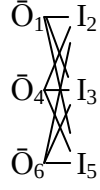
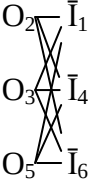
Images used in this study	Image 1 (W,B,W,B,W,B)				Image 2 (W,B,B,W,B,W)			
DNA form of images in input and output spaces	(I_1,$\bar{I}_2$,I_3,$\bar{I}_4$,I_5,$\bar{I}_6$) (O_1,$\bar{O}_2$,O_3,$\bar{O}_4$,O_5,$\bar{O}_6$)				(I_1,$\bar{I}_2$,$\bar{I}_3$,I_4,$\bar{I}_5$,I_6) (O_1,$\bar{O}_2$,$\bar{O}_3$,O_4,$\bar{O}_5$,O_6)			
Opposite DNA strands	($\bar{I}_1$,I_2,$\bar{I}_3$,I_4,$\bar{I}_5$,I_6) ($\bar{O}_1$,O_2,$\bar{O}_3$,O_4,$\bar{O}_5$,O_6)				($\bar{I}_1$,I_2,I_3,$\bar{I}_4$,I_5,$\bar{I}_6$) ($\bar{O}_1$,O_2,O_3,$\bar{O}_4$,O_5,$\bar{O}_6$)			
Opposite input DNA strands are divided into two groups and phosphorylated in different vials (D1-D4)	D1		D2		D3		D4	
	(I_2,I_4,I_6)		($\bar{I}_1$,$\bar{I}_3$,$\bar{I}_5$)		(I_2,I_3,I_5)		($\bar{I}_1$,$\bar{I}_4$,$\bar{I}_6$)	
Memory matrix building operation by ligation reaction								
Memory matrix oligomers produced in different vials (E1-E8)	E1	E2	E3	E4	E5	E6	E7	E8
	O_2I_2 O_2I_4 O_2I_6 O_4I_2 O_4I_4 O_4I_6 O_6I_2 O_6I_4 O_6I_6	**$\bar{O}_1I_2$** $\bar{O}_1I_4$ $\bar{O}_1I_6$ $\bar{O}_3I_2$ $\bar{O}_3I_4$ $\bar{O}_3I_6$ $\bar{O}_5I_2$ $\bar{O}_5I_4$ $\bar{O}_5I_6$	**$O_2\bar{I}_1$** $O_2\bar{I}_3$ $O_2\bar{I}_5$ $O_4\bar{I}_1$ $O_4\bar{I}_3$ $O_4\bar{I}_5$ $O_6\bar{I}_1$ $O_6\bar{I}_3$ $O_6\bar{I}_5$	**$\bar{O}_1\bar{I}_1$** $\bar{O}_1\bar{I}_3$ $\bar{O}_1\bar{I}_5$ $\bar{O}_3\bar{I}_1$ $\bar{O}_3\bar{I}_3$ $\bar{O}_3\bar{I}_5$ $\bar{O}_5\bar{I}_1$ $\bar{O}_5\bar{I}_3$ $\bar{O}_5\bar{I}_5$	**O_2I_2** O_2I_3 O_2I_5 O_3I_2 O_3I_3 O_3I_5 O_5I_2 O_5I_3 O_5I_5	**$\bar{O}_1I_2$** $\bar{O}_1I_3$ $\bar{O}_1I_5$ $\bar{O}_4I_2$ $\bar{O}_4I_3$ $\bar{O}_4I_5$ $\bar{O}_6I_2$ $\bar{O}_6I_3$ $\bar{O}_6I_5$	**$O_2\bar{I}_1$** $O_2\bar{I}_4$ $O_2\bar{I}_6$ $O_3\bar{I}_1$ $O_3\bar{I}_4$ $O_3\bar{I}_6$ $O_5\bar{I}_1$ $O_5\bar{I}_4$ $O_5\bar{I}_6$	**$\bar{O}_1\bar{I}_1$** $\bar{O}_1\bar{I}_4$ $\bar{O}_1\bar{I}_6$ $\bar{O}_4\bar{I}_1$ $\bar{O}_4\bar{I}_4$ $\bar{O}_4\bar{I}_6$ $\bar{O}_6\bar{I}_1$ $\bar{O}_6\bar{I}_4$ $\bar{O}_6\bar{I}_6$

Table 4.2: A schematic description summarizes how to produce the memory matrix, which is equivalent to the combination of two separate memory matrices. There are 68 different oligomers (100-mers with 20bp linker attached as a partial lower strand) in the memory matrix. Highlighted oligomers are common to both sets (E1-E4 & E5-E8). There are four different types of memory matrix oligomers (OI's, $\bar{O}$I's, O$\bar{I}$'s, and $\bar{O}\bar{I}$'s), each of which is generated in different vials, for each image. After the ligation reaction is over, the same type of oligomers obtained for different images are combined (E1&E5, E2&E6, E3&E7, E4&E8).

The first operation of the query image on memory matrix is to add a definite amount of 40bp input oligomers representing the query image into a sample of the memory matrix solutions stored in the 4 different vials. If the query input strands are complementary to some of the memory matrix oligomers in a vial, Watson Crick hybridization will take place and the 20bp linker attached to the memory matrix oligomers will denature. Otherwise, there will be no reaction. If hybridized molecules, containing partially dsDNA, are formed in any of the 4 vials, they will be extended using Klenow Fragment (3' → 5' exo) (New England Biolabs, Cat#M0212S, 5 units/μl) polymerase in the next step so that 100bp fully dsDNA molecules will be formed. A visual description of the molecular reactions described so far is displayed in the upper half of Figure 4.3.

After the 100bp fully dsDNA oligomers are formed in different vials, we separate them from other undesired strands with various lengths by one more gel extraction. Then, we transfer a portion of extracted solution including 100bp dsDNA template only into a new vial and start producing the output strands using the ILA (Isothermal Linear Amplification) reaction (the lower half of Figure 4.3)[1,3]. During this reaction the following steps will occur: (1) Nt.BbvCI (nicking enzyme) cuts the previously extended lower strands of dsDNA oligomers at a point where the extension was started. (2) Klenow Fragment polymerase starts extending the lower strands one more time while it also displaces the previously extended strands which are the output oligomers. (3) Steps 1 and 2 are continuously being repeated so that more output strands of each kind are produced. T4 Gene 32 single-strand binding protein (New England Biolabs,

Cat#M0300S, 10 µg/µl) is added into the ILA reaction mix to ensure that undesired fragment strands causing difficulty in further steps are not produced[1,3].

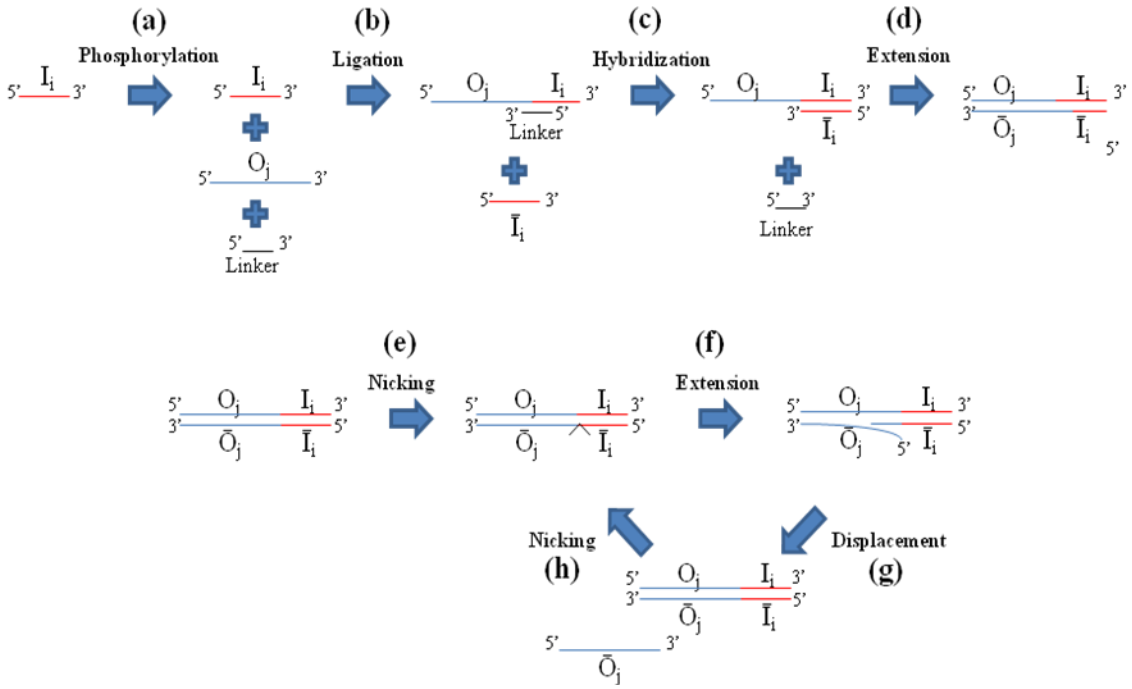


Figure 4.3: Visual description of the experimental procedure. (a) and (b) Memory matrix oligos are generated at the end of phosphorylation and ligation reactions, respectively. (c) Input strand hybridizes with memory matrix oligo. (d) Klenow fragment (3' → 5' exo-) extends the lower strand and produces a fully dsDNA structure (e), (f), (g), and (h) Output ssDNA is produced by ILA reaction.

In order to stop the ILA reaction, we heat-denature the polymerase, ssBP, and nicking enzyme. This action also helps to release ssDNA output strands bound/retained by ssBP so that they can be detected during the read out step. Since, we use a query image represented by 3 input strands, some output strands and their complements will be produced at the same time. Those strands hybridize with each other and yield 100bp

dsDNA oligomers, which don't cause a problem since they won't be detected during the read out.

The read out process on an Electronic Microarray can be summarized as follows: (i) a solution containing TET-primer molecules is added into the final solution obtained by ILA reaction, including the net output strands. TET-primer can hybridize with output strands since the 20bp primer attached to TET molecule is the complement of 20 nucleotides located at the 3' end of output strands. Note that the amount of TET-primers in that solution has to be at least equal or more than the total amount of output strands produced. (ii) A solution with a definite amount of Biotin-primer molecules is mixed with the same definite amount of each of O_1, O_2, O_3, O_4, O_5, O_6, $\bar{O}_1$, $\bar{O}_2$, $\bar{O}_3$, $\bar{O}_4$, $\bar{O}_5$, or $\bar{O}_6$ ssDNA oligomers in different vials. As discussed above, the hybridization reaction also takes place here. The definite amount has to be less than the amount of each output oligomer, since it will then act like the saturation function as described in Chapter 3. Solutions including hybridized structures of Biotin-primer and one of 12 output strands are dispatched on the surface of an Electronic Microarray one at a time. Each type of hybridized DNA strands including biotin molecule are attached on a different streptavidin coated pad by applying a current only on that pad. Note that the positively charged surface created by current is employed to attract the negatively charged DNA strands so that the strong non-covalent bond can be established between biotin and streptavidin molecules afterward. As a result, there will be 12 pads, each of which is covered by a different DNA sequence (Figure 4.4.a). (iii) Finally, the solution prepared in step (i) is also dispatched on the surface of the cartridge. Each output strand attached to TET-

primer will be immobilized by hybridizing with its corresponding complement which is attached to the surface of a pad (Figure 4.4.b). Under the green light, TET molecules fluoresce, thus, we can get the recalled image and its reverse on the Electronic Microarray.

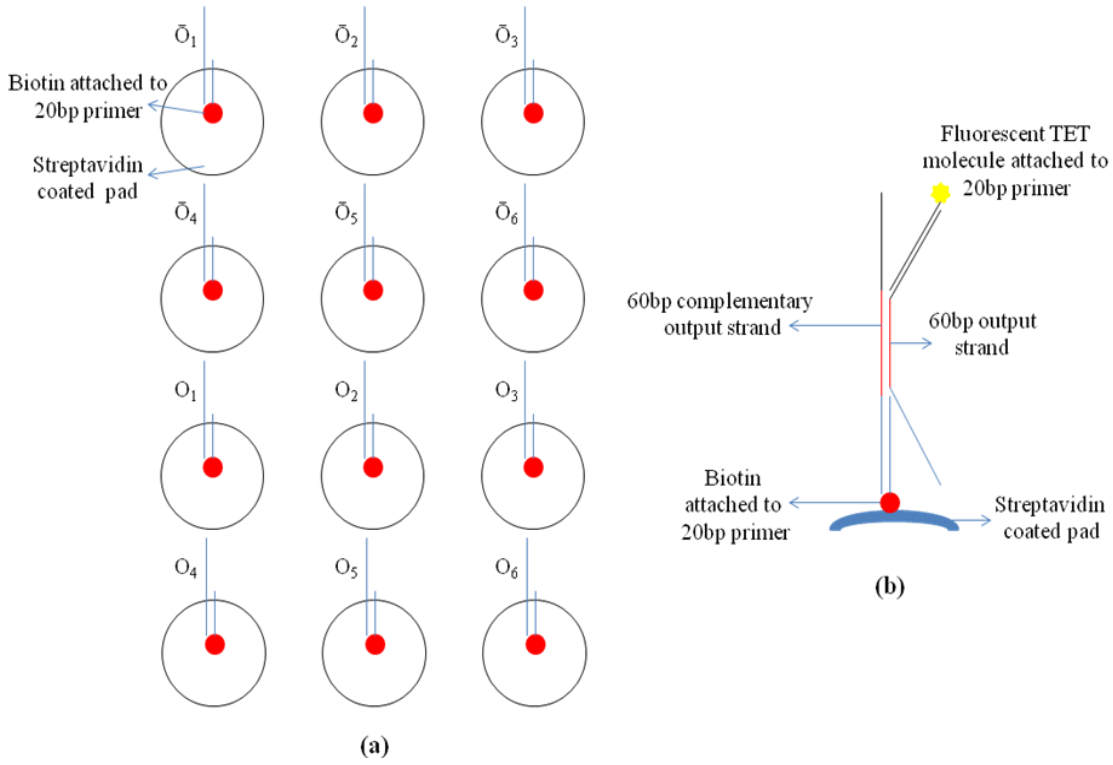


Figure 4.4: Visual description of "read-out" step on Electronic Microaarray. (a) Each of twelve possible output strands is attached on different pads using Biotin-primer. (b) After the solution including net output strands amplified by isothermal linear reaction is dispatched on microarray, each output strand present will be immobilized by hybridizing with its corresponding complementary. Non-complementary and complementary components of the recalled image are obtained in the upper and lower half of 12 pads, respectively. Note that different colors are used to illustrate three different 20bp parts of an output or complementary output strand.

4.3. Details of the protocol for recall of memory

The details of experimental protocols used in this study are explained below in the same order as they were performed.

(i) Phosphorylation. Four different groups of input strands, (I_2, I_4, I_6), ($\bar{I}_1$, $\bar{I}_3$, $\bar{I}_5$), (I_2, I_3, I_5), ($\bar{I}_1$, $\bar{I}_4$, $\bar{I}_6$) in Table 4.2, contained in vials D1, D2, D3, D4, respectively, are phosphorylated. Note that there are 3 oligomers in each vial. Input strands placed in vials D1 & D2 represent the opposite DNA strands of the first image in input space divided into two groups. Similarly, the strands in vials D3 & D4 correspond to the second image. Phosphorylation reactions are carried out in a reaction mixture with a total volume of 50 μl containing 50 mM Tris-HCl (pH 7.5 @ 25 °C), 10 mM $MgCl_2$, 1mM ATP, 10 mM Dithiothreitol, 15 units T4 Polynucleotide Kinase, 3 μM each input strand, and Nuclease-free water. After vortexing, mixtures are incubated at 37 ºC for 30 minutes, then, heated at 65 ºC for 20 minutes to deactivate kinase. The efficiency of phosphorylation reaction by the above protocol, which is derived from previously reported studies[6-7], is expected to be 100%.

(ii) Building the memory matrix by ligation reaction. 40 μl of solution in D1 is evenly transferred into two new vials: E1 & E2. In a similar manner, vials E3 & E4, E5 & E6, and E7 & E8 are formed by solutions found in vials D2, D3 and D4. As described in Table 4.2, four different groups of output strands, (O_2, O_4, O_6), ($\bar{O}_1$, $\bar{O}_3$, $\bar{O}_5$), (O_2, O_3, O_5), ($\bar{O}_1$, $\bar{O}_4$, $\bar{O}_6$), corresponding to the opposite DNA strands of the first and second images in output space are also added into vials E1 & E3, E2 & E4, E5 & E7, E6 & E8,

respectively. Finally, 20bp linker is put into each vial. Ligation reactions are carried out in a reaction mixture with a total volume of 50 µl containing 50 mM Tris-HCl (pH 7.5 @ 25 °C), 10 mM $MgCl_2$, 1mM ATP, 10 mM Dithiothreitol, 600 units T4 DNA Ligase (New England Biolabs, Cat#M0202S, 400 cohesive end units/µl), 1.2 µM each input and output strands, 3.6 µM 20bp linker, and Nuclease-free water. After vortexing, mixtures are incubated at room temperature for 2 hours.

(iii) Purification of memory matrix strands by gel extraction. We load the 8 resulting samples including memory matrix (ligated) oligomers (10 µl of Bromophenol blue is added to each one) on a different 0.75mm 10 well 8% polyacrylamide gel (produced in our lab) and begin running the electrophoresis for 45 minutes at 155 V using the Mini-PROTEAN® Tetra Cell vertical electrophoresis tank (Bio-Rad). 1x Tris Borate-EDTA (TBE) buffer (Sigma-Aldrich) is used as the running buffer. After electrophoresis, SYBR Gold (Invitrogen) nucleic acid gel stain, which is the most sensitive fluorescent stain available for detecting double or single stranded DNA molecules, is used to stain DNA oligomers. The polyacrylamide gel with DNA is soaked on a shaker table in a stain buffer, which is a mixture of 5 µl of 10000x SYBR Gold gel stain and 100ml 1x TBE buffer, for 20 minutes at room temperature. Finally, we take CCD camera images of the gels containing DNA strands stained by SYBR Gold under a 312 nm UV light (Fisher Scientific). The excitation maxima for this type of stain is ~300nm. The exposure time is 5 seconds for all images. The results are analyzed by using Kodak 1D image analysis system (Fisher Scientific).

The memory matrix oligomers shown as a band on gel images are extracted using the standard 11 steps QIAEX® II gel extraction protocol (by QIAGEN). These steps are:

1) Excise the DNA band from the polyacrylamide gel with a clean, sharp scalpel.
2) Weigh the gel slice in a colorless tube. Assuming that the density of the gel is approximately 1 mg/µl, add 2 volumes of Buffer QX1 to 1 volume of gel for DNA strands.
3) Resuspend QIAEX II by vortexing for 30 s. Add 10 µl QIAEX II to the sample and mix.
4) Incubate at 50°C for 30 min to solubilize the polyacrylamide and bind the DNA. Mix by vortexing every 2 min to keep QIAEX II in suspension.
5) Centrifuge the sample for 30 s and carefully remove supernatant with a pipet.
6) Wash the pellet with 500 µl of Buffer QX1.
7) Wash the pellet twice with 500 µl of Buffer PE.
8) Air-dry the pellet until the pellet becomes white.
9) To elute DNA, add 20 µl of 10 mM Tris·Cl, (pH 8.5) and resuspend the pellet by vortexing. Incubate the mix at room temp for 5 min.
10) Centrifuge for 30 s. Carefully transfer the supernatant into a clean tube.
11) Optional: repeat steps 9 and 10 and combine the eluates.

From each solution (E1-E8), approximately 32 µl of new solution is obtained when the process is over. Since the solutions extracted from E1 and E5 contain the same type of memory matrix oligomers (OI's), we combine them in vial F1. In a similar manner,

solutions in vials F2, F3, and F4 corresponding to ($\bar{O}I$'s, $O\bar{I}$'s, and $\bar{O}\bar{I}$'s) are derived from (E2 & E6), (E3 & E7), and (E4 & E8) pairs, respectively. The volume of solution in each newly created vial is 64 µl. The concentration of memory matrix oligomers in these solutions is undetermined because we don't precisely know the efficiency of ligation reaction or gel extraction. These efficiencies are not fixed since they are strongly related to the experimental conditions. Small departures from the original protocol might cause noticeable changes in efficiency values.

(iv) Hybridization between input and memory matrix oligomers. This step is performed in four different vials (F1-F4). $\bar{I}_2$, I_3, and $\bar{I}_4$ input strands referring to the incomplete query image are added into each vial along with 10xNeBuffer 2 (New Englad Biolabs). Hybridization reactions are carried out in a reaction mixture with a total volume of 80 µl containing 10 mM Tris-HCl (pH 7.9 @ 25 °C), 50 mM NaCl, 10 mM $MgCl_2$, 1 mM Dithiothreitol, 0.5 µM $\bar{I}_2$, 0.25 µM I_3 & $\bar{I}_4$, indefinite amount of one of 4 types of memory matrix oliigomers, and Nuclease-free water. After vortexing, mixtures are incubated at 55 ºC (average optimum hybridization temperature for 3 input strands) for 1 hour. Note that the amount of $\bar{I}_2$ is twice as big as the amount of other two input strands (I_3, and $\bar{I}_4$). This is necessary because the color of 2^{nd} pixel for both images is the same whereas the 3^{rd} and 4^{th} pixels are different, thus, the number of memory matrix oligomers including I_2 is twice as big as the number of oligomers including $\bar{I}_3$ or I_4.

(v) Extension reaction. We continue doing the experiment in 4 different vials. Simply, 5 units Klenow fragment polymerase, 3 µl Deoxynucleotide solution mix (dNTP) (New

England Biolabs, Cat#N0447S, 10 mM), and 0. 44 µl 10xNeBuffer (to maintain the buffer concentration at 1x) are added into each vial. After vortexing, mixtures are incubated at 37 ºC for 1 hour. At the end, 100bp fully dsDNA oligomers are created.

(vi) Purification of extended strands by gel extraction. After adding 16 µl of Bromophenol blue into each vial, the solutions including 100bp dsDNA oligomers along with other undesired DNA strands produced during the last two steps are loaded on a different 0.75mm 10 well 8% polyacrylamide gel (produced in our lab) and begin running the electrophoresis for 65 minutes at 155 V. Except the duration of electrophoresis, the rest of the process followed at this step is the same as the one described in step (iii).

(vii) Isothermal Linear Amplification (ILA). This step is needed to increase the quantities of output strands to the point where they can be detected on an Electronic Microarray. At this step, we can combine all 100bp dsDNA oligomers together in a single vial. The total volume of solution in that vial is 128 µl, however, the concentration is undetermined. The same ILA reaction is performed in 2 different vials in parallel to get enough output DNA molecules for the next step. Each 50 µl mixture contains 10 mM Tris-HCl (pH 7.9 @ 25 °C), 50 mM NaCl, 10 mM $MgCl_2$, 1 mM Dithiothreitol, 25 units Nt.BbvCI nicking enzyme, 5 units Klenow Fragment polymerase, 10 µg T4 Gene 32 ssBP , 1000 µM dNTPs, 15 µl solution of pure 100bp dsDNA with indefinite concentration, and Nuclease-free water. After vortexing, mixtures are incubated at 37 ºC for 1 hour, then, heated at 80 ºC for 20 minutes to deactivate the enzymes and cooled down to room temperature slowly. One more time, we incubate the mixtures at 58 ºC

(average optimum hybridization temperature for output strands) for 30 minutes. This final incubation is necessary because some output strands and their corresponding complementary produced during the amplification reaction must hybridize with each other, thus, we can obtain the net amount of output strands.

(viii) Read-out on Electronic Microaarray. As we discussed in previous section, each of O_1, O_2, O_3, O_4, O_5, O_6, $\bar{O}_1$, $\bar{O}_2$, $\bar{O}_3$, $\bar{O}_4$, $\bar{O}_5$, or $\bar{O}_6$ ssDNA oligomers is hybridized with Biotin-primer molecules in different vials. 50 µl mix in a vial contains 100 mM Histidine, 120 mM 1-Thioglycerol, 0.625 µM Biotin-primer, and 0.25 µM one of 12 output ssDNA. The reason why we have Histidine rather than NaCl in the solution is because Histidine molecules provide the best condition on the Electronic Microaarray surface for the next step. After vortexing, mixes are dispatched on a DNA cartridge one at a time. Each type output strand is attached on a different pad by applying 800 nA constant current for 60 seconds. Afterwards, we wash the cartridge with histidine buffer and water twice to remove the oligomers which are not attached to the desired locations. When this process is over, the appearance on the cartridge looks like Figure 4.4.a. Next, the net output stands obtained by ILA reaction is hybridized with TET-Primer molecules in a single vial. 100 µl reaction mix contains 30 mM sodium phosphate, 300 mM NaCl, 1 µM TET-primer, and 69 µl solution of net output strands with indefinite concentration. After vortexing, the mix is dispatched on the cartridge surface in two cycles. In each cycle, as soon as the operation starts the temperature is gradually increased and held at 60 °C for 3 minutes. This high temperature removes any weak secondary structures, such as hairpins or heterodimers, formed by the net output strands or other oligomers previously attached

to different pads. In addition, it also helps to solve the mishybridization problem that might occur between partially complementary sequences of ssDNA oligomers. After 3 minutes, the temperature is slowly decreased to 29 °C at 1 °C per 90 seconds rate. At this temperature, the cartridge is washed with 50 µl high salt buffer including 50 mM sodium phosphate and 500 mM NaCl. Then, the temperature is downgraded to 24 °C for imaging. Tet molecules are excited with a 525nm green light. The Green Integration time of the camera is held constant 12 seconds when the image of the array is taken.

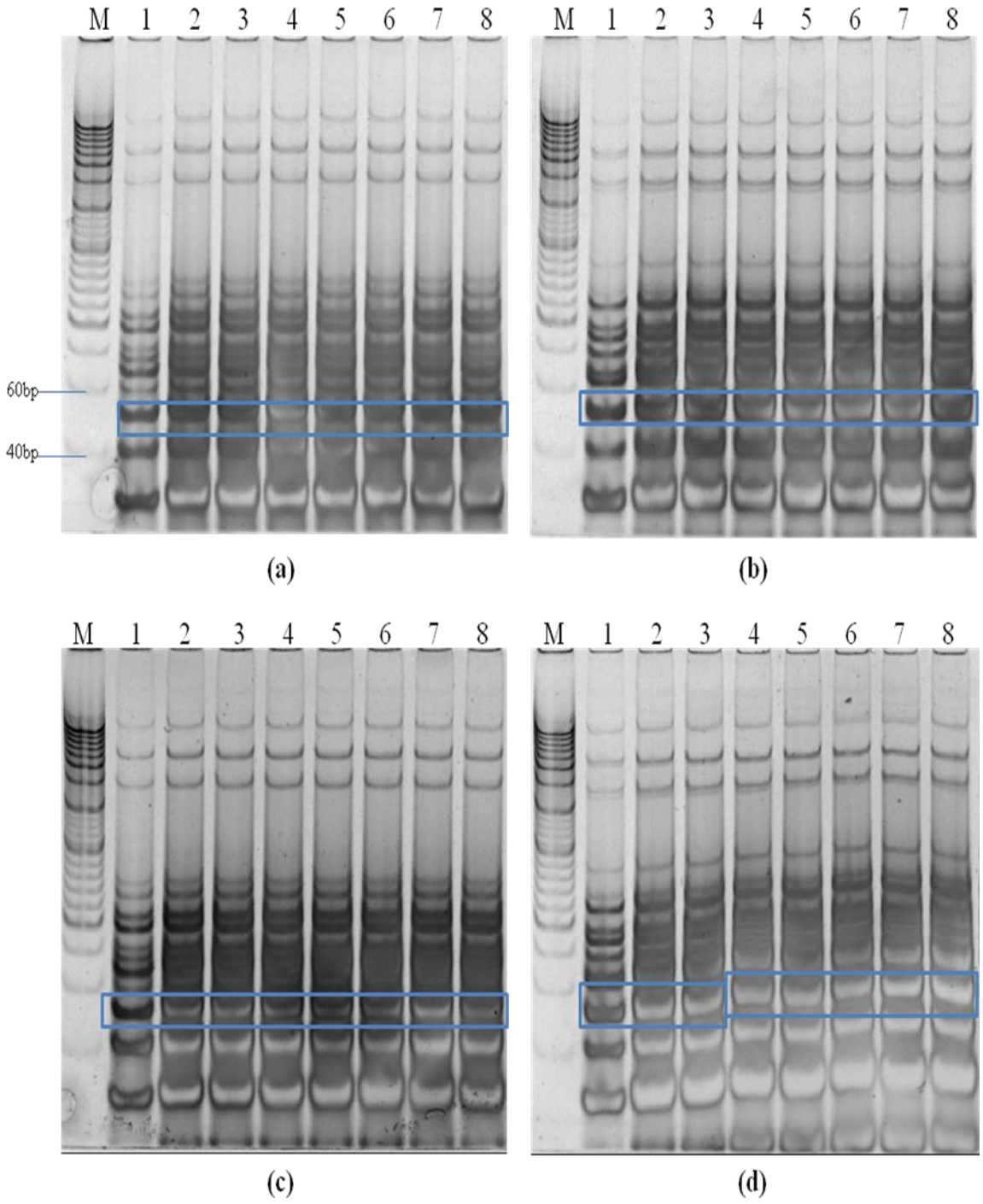
4.4. Results and Discussion

The gel image results of "ligation reaction" representing the memory matrix building operation are seen in Figure 4.5. Images (a)-(h) are related to the results of polyacrylamide gel electrophoresis of 50 µl solutions in vials E1-E8. For each gel, while 3 µl of the corresponding solution is loaded on lane 1 (to obtain a better image showing separate bands), the rest is evenly loaded on lanes 2-8. As stated in previous section, reaction is performed in 4 different vials for each of two initial 6-bit images. Vials E1-E4 and E5-E8 refers to the first and second images. When the ligation reaction is over, solutions in vials (E1 & E5), (E2 & E6), (E3 & E7) and (E4 & E8) contain one of four types of memory matrix strands (OI's, $\bar{O}$I's, O$\bar{I}$'s, or $\bar{O}\bar{I}$'s), respectively.

According to one of our preliminary experiment, in which we ligate only one pair of input and output strands using 20bp linker, the band representing the ligated DNA structure appears at 50bp dsDNA level on an 8% polyacrlamide gel image. Based on this knowledge, we extract the bands encircled by rectangles on images (a)-(h) representing

the ligated DNA strands obtained by 3 different pairs of input and output strands. Note that all images have a similar look. However, the thicknesses of these bands considered being proportional to the amount of ligated DNA molecules vary from one gel to another although they are expected to be the same since the initial amounts of input and output DNA strands and conditions are the same. The differences might be related to pipetting errors, not achieving the same ligation efficiency in each vial or etc. However, as long as there is not a major variation, such as one of the reactions in a vial doesn't yield any result at all or works with a very low efficiency comparing to other cases, this is not a problem since the Hopfield Network model is fault tolerant[9-10], thus, the shortness in recalling mechanism due to missing memory matrix oligomers caused by minor experimental mistakes are covered by others. On the other hand, there are many (at least 10 or even more) other bands showing the result of undesired reactions which use up our input and output strands and form unidentified structures. This shows that the purification of ligated DNA strands is essential and the efficiency of ligation reaction is very low although we can't determine it precisely.

The gel images indicating the results of "extension reaction" are presented in Figure 4.6. Images (a)-(d) are related to the results of polyacrylamide gel electrophoresis of 84.44 µl solutions in vials F1-F4, which are designated to represent 4 different types of memory matrix strands, OI's, $\bar{O}$I's, O$\bar{I}$'s, and $\bar{O}\bar{I}$'s, respectively. For each gel, the entire amount of corresponding solution is evenly loaded on lanes 1-8. Because of the first


M 1 2 3 4 5 6 7 8
60bp
40bp
(a)
M 1 2 3 4 5 6 7 8
(b)
M 1 2 3 4 5 6 7 8
(c)
M 1 2 3 4 5 6 7 8
(d)

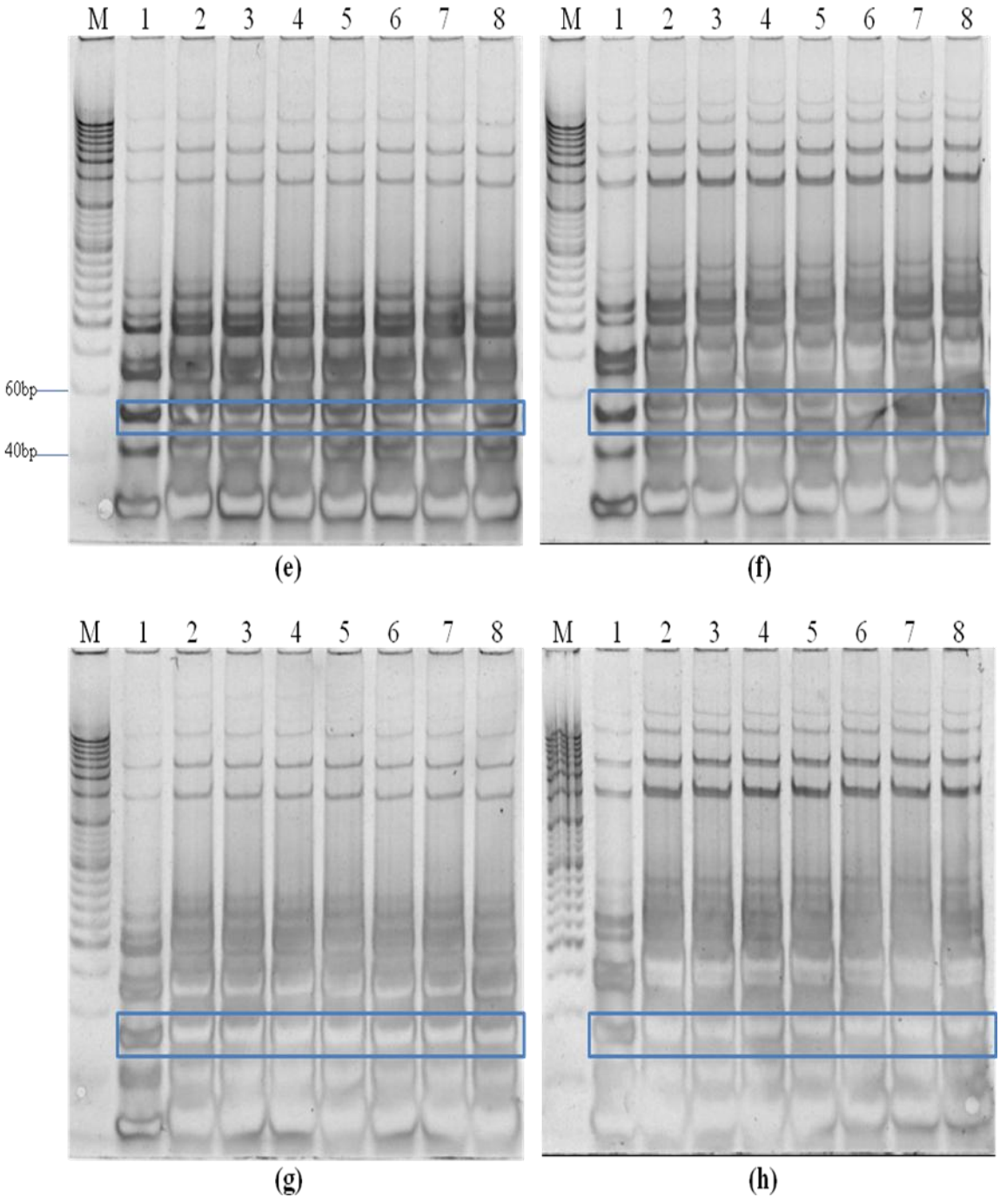


Figure 4.5: 8% polyacrylamide gel images obtained at the end of ligation reactions. (a)-(h) corresponds to samples E1-E8. The bands encircled by blue rectangles represent ligated DNA molecules. Lane M contains 20bp dsDNA marker.

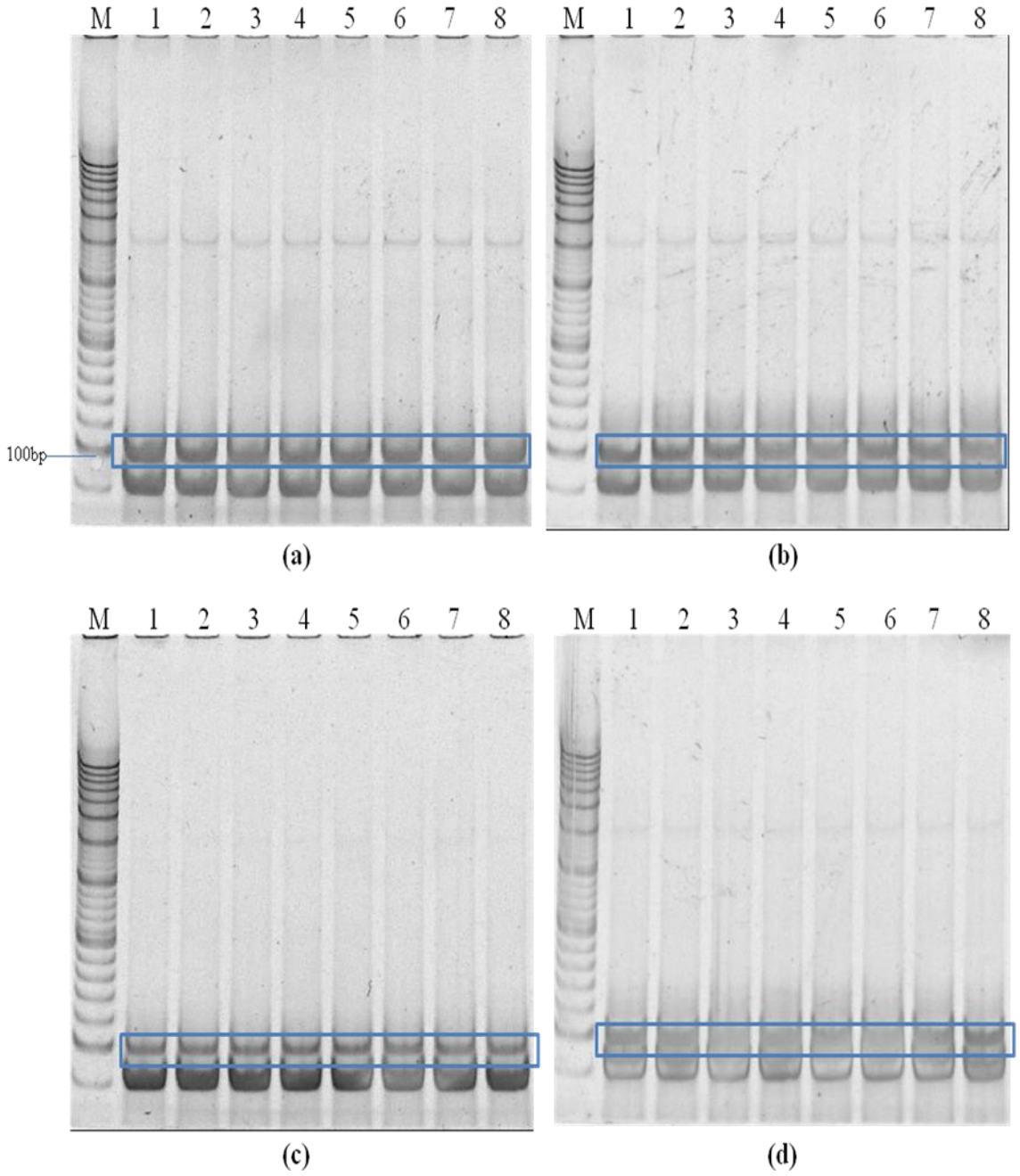


Figure 4.6: 8% polyacrylamide gel images. (a)-(d) indicates the result of extension reaction for 4 different types of memory matrix strands, OI's, $\bar{O}$I's, O$\bar{I}$'s, and $\bar{O}\bar{I}$'s, respectively. The bands encircled by blue rectangles represent 100bp dsDNA molecules. Lane M contains 20bp dsDNA marker.

purification step mentioned just above, the results of extension reaction are relatively cleaner comparing to earlier step. After 65 minutes electrophoresis, two types of DNA strands are left in gels. According to our design, 100bp dsDNA molecules should be produced as the expected output strands at the end of extension reaction. The upper bands representing desired products on each gel encircled by a rectangle are extracted. Note that the bands at 100bp level on the third and fourth gels are thinner than those on the first and second ones. This is an expected result because input strands ($\bar{I}_2$ and $\bar{I}_4$) representing two thirds of the incomplete query image can hybridize with 9 possible memory matrix oligomers in each of vials F1 and F2 whereas the other input strand (I_3) can hybridize with only 3 memory matrix oligomers in each of vials F3 and F4.

Finally, the resulting image taken in "read-out on Electronic Microaaray" step is seen in Figure 4.7. Because of our design described in Figure 4.4.a, both the recalled image and its reverse are obtained during this step. As expected, the first 6-bit image, from which the query image was created, is successfully restored on the cartridge as the result of a series of DNA experiments. Image analysis by Nanogen 400 Data Viewer software gives below light intensity values for 12 pads on the cartridge.

1st: **14808.8**	2nd: 1292.9	3rd: **12281.5**
4th: 1442.2	5th: **13685.1**	6th: 1732.5
7th: 838.7	8th: **12943.4**	9th: 1246.6
10th: **12776.2**	11th: 1122.3	12th: **14246.6**

Notice that there are two groups of numbers: (i) 6 highlighted values (> 12000) are in the first group and correspond to white pixels, (ii) other 6 values (< 2000) are in the second group and correspond to black pixels. The average light intensity of higher values in the first group is at least 6 times bigger than that of lower values in the second group. As we stated earlier, the surfaces of 1^{st}, 3^{rd}, 5^{th}, 8^{th}, 10^{th} and 12^{th} pads are initially covered by O_1, O_3, O_5, $\bar{O}_2$, $\bar{O}_4$ and $\bar{O}_6$, respectively. If we combine this information with the fact that those pads are bright, we can conclude that $\bar{O}_1$, $\bar{O}_3$, $\bar{O}_5$, O_2, O_4 and O_6 strands attached to TET-primer molecules are present on the cartridge. In other words, they are the net output strands produced by DNA molecular reactions as expected and refer to the first 6-bit memory.

Ideally, no light should come from 2^{nd}, 4^{th}, 6^{th}, 7^{th}, 9^{th} and 11^{th} pads since such a result would imply that $\bar{O}_2$, $\bar{O}_4$, $\bar{O}_6$, O_1, O_3 and O_5 ssDNA molecules are also found in the net output solution as opposed to our expectation. However, such a result might occur as a result mishybridization reactions or the fact that some oligomers might have been placed on non-corresponding pads by experimental error. For instance, while O_1 strands are placed on the 1^{st} pad, some of these strands might also be placed on 2^{nd} and 4^{th} pads since they are close by the 1^{st} one. Thus, we consider these low light intensity values as the background noise and ignore their presence.

Finally, the normalized light intensities of 6 luminous pads (14808.8, 12281.5, 13685.1, 12943.4, 12776.2 and 14246.6) over the average value are 1.10, 0.91, 1.02,

0.96, 0.95 and 1.06, respectively. This result shows the success of our implementation of saturation function as described in “Materials and Methods” section.

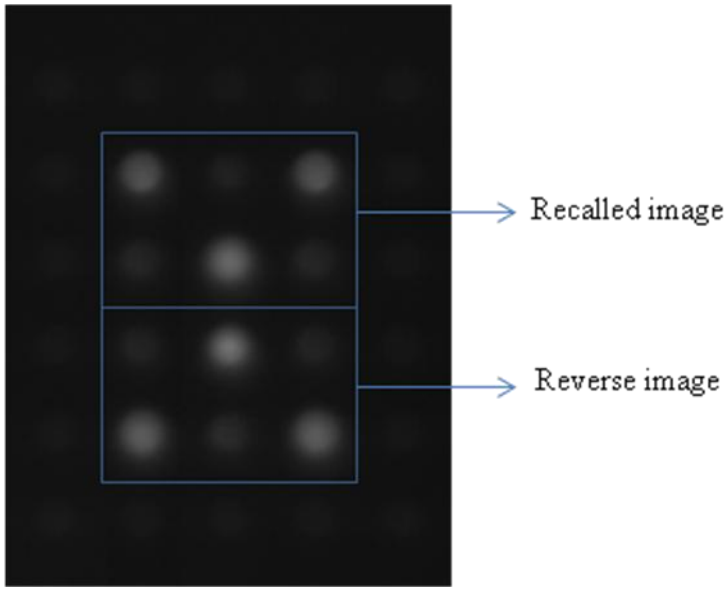


Figure 4.7: The “read-out” image obtained by Electronic Microarray technology.

4.5. Conclusion

In this chapter, we present the experimental implementation of a 6-bit Hopfield neural network using DNA molecules. Although the size of the memory images is small, our study indicates the scalability of this kind of implementation since a complete set of experimental procedures including the memory matrix building operation by “ligation reaction” is carried out for the first time.

In this study, two 6-bit images are used as experiences and a DNA based memory matrix representing Hopfield associative memory is constructed. A query image derived from the first memory image by removing 3-bit of information is employed to operate on the memory to recall the corresponding image (the first one). In reality, as we discussed in Chapter 3, using a query image including 1-bit information (as long as this bit is not common in both original memory images), is enough for a successful recalling. However, such a case is not possible in mathematical representation of Hopfield Neural Network. The advantage of DNA based Hopfield Netural Network over the theoretical one is the fact that there is no memory lost in DNA based Hopfield Neural Network whereas such a problem happens in theoretical model. In DNA implementation, since the memory matrix oligomers are produced and contained in different tubes according to their types, no fully or partially complementary strand are mixed in the same solution and this prevents the loss of memory.

For 6-bit implementation, three different 150 picomole input strands are phosporylated in each vial in the first step. Instead, we could start with 500 different 1 picomole input strands (referring to 1000-bit Hopfield Neural Network built by two 1000-pixel images each of which is assumed to be consisted of 500 white 500 black pixels). For this hypothetical case, if the exact same procedure used in this study is followed, 500 different 0.4 picomole input strands will be ligated with the same amount of 500 corresponding output strands . Considering the fact that each 0.4 picomole input strand will be hybridized with 500 different output strands equally and the overall efficiency of “ligation reaction” and the first “extraction” steps is approximately 10%, we

conclude that 25000 different $8x10^{-5}$ picomole ligated DNA oligomers can be generated in each of 4 different vials. If 100 different input strands representing a 100-pixel query image (suppose that there are 50 white and 50 black pixels and none of these pixels is common in both original 1000-pixel memory images) is operated by memory matrix oligomers, ideally, $4x10^{-3}$ picomole of each type of output strand, the sum of which correspond to the recalled image, will be produced. However, since the efficiency of "Hybridization between input and memory matrix oligomers", "extension reaction" and the second "extraction" steps is assumed to be also 10%, this number will go down to $4x10^{-4}$. Finally, 4 picomole of each output strand can be produced by performing 10 parallel ILA reactions, the conditions of which can be adjusted to achieve > 10^3-fold amplification[11-12], in different vials. Note that 4 picomole of DNA molecules is detectable on the Electronic Microarray. This hypothetical case indicates the possibility of building 1000-bit Hopfield associative memory. However, such a big size experiment might require further improvements to be made to improve the effectiveness and precision.

As an alternative of ordering 1000 different input and output strands required for above case, the individual input and output DNA oligomers could be produced by PCR amplification of single random oligomers[13] to reduce the expense. In addition, an automated system composed of mechanical and electronic parts can be developed to govern the reactions[14]. If a method to conduct many different parallel reactions, each of which works as a subroutine to process one type of computation, as described here is found DNA implementation of more complex problems can be possible in the future.

References:

[1] Karabay, D., Hughes, B. S. T. and Mills Jr., A. P. "Feasibility of a Hopfield Neural Network using DNA molecules." *J. Comput. Theor. Nanosci.* Accepted for publication.

[2] Hopfield, J. J. "Neural networks and physical systems with emergent collective computational abilities." *Proc. Natl. Acad. Sci. USA* 79 (1982): 2554–2558. Print.

[3] Liu, Q., et al. "DNA Computing on Surfaces." *Nature* 403 (2000): 175-179. Print.

[4] Akin, H. E., et al. "Electronic Microarrays in DNA Computing." *J. Nanosci. Nanotechnol.* Accepted for publication.

[5] Karabay, D., Hughes, B. S. T. and Mills Jr., A. P. "Application of T4 Gene 32 SSB Protein to the Improvement of Isothermal Linear Amplification (ILA) of DNA." In Preparation.

[6] Richardson, C.C. *The Enzymes* Ed. Boyer, P. D. San Diego: Academic Press, 1981. 229-314. Print.

[7] Berkner, K.L. and Folk, W.R. "Polynucleotide kinase exchange reaction: quantitave assay for restriction endonuclease-generated 5phosphoroyl termini in DNA." *J. Biol. Chem.* 252 (1977): 3176-3184. Print.

[8] Engler, M.J. and Richardson, C.C. *The Enzymes* Ed. Boyer, P. D. San Diego: Academic Press, 1982. 3-4. Print.

[9] Deaton, R., et al. "Reliability and efficiency of a DNA-based computation." *Phys. Rev. Lett.* 80 (1998): 417–420. Print.

[10] Mills Jr., A. P., Yurke, B., and Platzman, P. M. "Article for analog vector algebra computation." *BioSystems* 2 (1999): 175-180. Print.

[11] Walker, G. T., et al. "Strand displacement amplification--an isothermal, in vitro DNA amplification technique." *Nucleic Acids Res.* 20 (1992): 1691–1696. Print.

[12] Van Ness, J., Van Ness, L. K., Galas, D. J. "Isothermal reactions for the amplification of oligonucleotides." *Proc. Natl. Acad. Sci. USA* 100 (2003): 4504-4509. Print.

[13] Wang, S., et al. "Production of random DNA oligomers for scalable DNA computing." *Biotechnol. J.* 4 (2009): 119–128. Print.

[14] Johnson, C. "Automating the DNA Computer: Solving n-Variable 3-SAT Problems." *LCNS* 4287 (2006): 360-373. Print.

Chapter 5 – General Conclusion

In Chapter 2, we showed that adding T4 Gene 32 single-strand binding protein (ssBP) in the Isothermal Linear Amplification (ILA) reaction mix prevents the formation of fragment strands when the reaction is over. According to our results, the amount of fragment strands is monotonically increases as the length of the output strand and the reaction time increase. The amount of ssBP used in a reaction mix has to be chosen carefully so that all output strands produced during the reaction can be captured by ssBP molecules. Otherwise, the success of solving the fragmentation problem can't be guaranteed. Additionally, we show that ssDNA oligonucleotides previously captured by ssBP molecules preferentially hybridize with their complementary strands rather than being attached to ssBP by the "ssBP vs. Hybridization" experiment.

In Chapter 3, the feasibility of a Hopfield Neural Network (HNN) using DNA molecules as computation substances was proven. 5 different cases were studied, each of which was needed to show different properties of DNA based HNN. Our results indicated that removing the memory matrix oligomers representing the diagonal elements of the Hopfield memory matrix, which are set to zero in mathematical model, is not essential. In fact, they have to be included to be able to make a complete recall in cases where only one input strand and one iteration are employed. In addition, 2-bit query (clue) vectors were used in two different cases, in each of which one of the original memory vectors was successfully restored. Finally, we showed that a query vector consisted of only one

basis vector (1-bit of information), which is in common in both original vectors, can't distinguish if the first or second image is the correct recall.

In Chapter 3, we extended our work presented in the previous chapter by adding the "ligation reaction" and "read-out on an Electronic Microarray" features into earlier version of experimental procedure. Our aim was to prove the scalability of DNA based HNN. Specifically, randomly chosen two 6-bit images were stored in a 6-bit DNA HNN. Then, a 3-bit query image generated from the first original memory image was operated on the memory matrix oligomers. As expected, the first image was successfully restored on an Electronic Microaarray. In addition, we described a sketch of experimental procedure of a hypothetical case to show the possibility of building 1000-bit DNA HNN in the future. However, to be able implement such a big size problem, further improvements and modifications on the experimental algorithm may be needed to improve the effectiveness. One day, if an automated system that is capable of running many experimental procedures, each of which can be assumed as a subroutine, in parallel is developed, different type of more complex problems might be solved using DNA.