

AFRL-RI-RS-TR-2010-072
In-House Final Technical Report
March 2010



SUPERIMPOSED CODE THEORECTIC ANALYSIS OF DNA CODES AND DNA COMPUTING

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2010-072 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/
THOMAS RENZ
Branch Chief

/s/
EDWARD J. JONES, Deputy Chief
Advanced Computing Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) MARCH 2010		2. REPORT TYPE Final		3. DATES COVERED (From - To) April 2007 – September 2009	
4. TITLE AND SUBTITLE SUPERIMPOSED CODE THEORECTIC ANALYSIS OF DNA CODES AND DNA COMPUTING				5a. CONTRACT NUMBER In House	
				5b. GRANT NUMBER N/A	
				5c. PROGRAM ELEMENT NUMBER 61120F	
6. AUTHOR(S) Thomas Renz and Anthony Macula				5d. PROJECT NUMBER 232T	
				5e. TASK NUMBER DN	
				5f. WORK UNIT NUMBER IH	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFRL/RITA 525 Brooks Road Rome NY 13441-4505				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RITA 525 Brooks Road Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) N/A	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TR-2010-072	
12. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2010-0927 Date Cleared: 3-March-2010					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT In this project, a synthetic Deoxyribonucleic Acid, DNA-based memory called ComDMems (Combinatorial DNA Memories) was developed. The research focused on the application and implementation of combinatorial based information theory and group testing to create associative DNA memories and to retrieve information stored in these DNA memories by chemical and electro-chemical means. This research demonstrates that this combinatorial method can feasibly yield billions of covert and synthetic DNA memory strands that carry object and process information. A key component of this innovation is the combinatorial method of bio-memory design and detection that encodes item or process information as numerical sequences represented in DNA. ComDMem is a content addressable memory (CAM) as opposed to a standard random access memory (RAM). A standard RAM goes directly to a physical address and returns the contents. ComDMem achieves CAM when multiple parallel PCR probes, specific for certain pieces of information search, the ComDMem for memories that contain these pieces of information. In this way all memories associated with a concept(s) can be retrieved and decoded in parallel. A visualization model was created and a tiled display system was built to provide high resolution display of large images for verification of probe-strand interactions.					
15. SUBJECT TERMS DNA Codes, Content Addressable Memory, Molecular Memory, Polymerase Chain Reaction, Informatics Visualization					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 39	19a. NAME OF RESPONSIBLE PERSON Thomas Renz
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) N/A

TABLE OF CONTENTS

1.0	SUMMARY	1
2.0	INTRODUCTION	2
3.0	METHODS, ASSUMPTIONS, PROCEDURES	6
3.1	Numerical Sequences Represented in DNA	6
3.2	Polymerase Chain Reaction Laboratory Method	8
3.3	Memory Design and Synthetic DNA Code SynDCode Software	8
3.4	The PCR Signal and PCR Network Graph	9
3.5	Oligo Visualization with 3DViews	13
4.0	RESULTS, DISCUSSION	14
4.1	The Mathematical Model	14
4.2	The Identification Algorithms	17
4.3	Algorithmic Implementation	18
4.4	The Algorithms Applied to Physical Experimental Data	22
4.5	The General Setting, Parameters and Simulated Performance	25
4.6	Visualization with 3DViews	26
5.0	CONCLUSIONS	29
6.0	REFERENCES	34
7.0	ACRONYMS	33

TABLE OF FIGURES

1	A DNA Code	3
2	Scheme for Parallel Search for Multiple Association in DNA	3
3	The DNA Associative Array Relational Table	5
4	A DNA_TC(5,2,10)	6
5	A " <i>longer strand</i> " double-stranded WC ComDTag	7
6	PCR Network Graph	10
7	Covering Strands for Memory in Figure 5	11
8	PCR Results from Covering Strands in Figure 7	12
9	PCR Graphs of Solely Positive PCR Reactions	12
10.	PCR graph G_U with $U=\{11000, 00110, 11100, 11110\}$	16
11.	G^{t-1} to G^t Graph Extension Scheme	19
12.	G^{t-1} to G^t Graph Extension Algorithm 1	20
13.	How Algorithm 1 Leads to ComDMem Decoding	21
14.	Unique Edge Representative Computational Cost Reduction Algorithm 2	22
15.	Record of Actual PCR Results	23
16.	A Portion of the Electrophoresis Gel from the PCR	25
17.	Simulations of Algorithmic Performance	25
18.	MOSAIC Cluster	27

1.0 SUMMARY

In this project, a synthetic Deoxyribonucleic Acid, DNA-based memory called ComDMems (Combinatorial DNA Memories) was developed. This research demonstrates that this combinatorial method can feasibly yield billions of covert and synthetic DNA memory strands that carry object and process information. A key component of this innovation is the combinatorial method of bio-memory design and detection that encodes item or process information as numerical sequences represented in DNA. This DNA data structure can be read by the wet laboratory method polymerase chain reaction (PCR) and then algorithmically decoded to retrieve virtually an unlimited amount of item or process information that has been stored in the combinatorial memories.

ComDMem is a content addressable memory (CAM) as opposed to a standard random access memory (RAM). A standard RAM goes directly to a physical address and returns the contents. A CAM uses the content of the input to direct the search of its entire memory for the specified data word.

ComDMem is a content addressable memory (CAM) as opposed to a standard random access memory (RAM). A standard RAM goes directly to a physical address and returns the contents. ComDMem achieves CAM when multiple parallel PCR probes, specific for certain pieces of information, search the ComDMem for memories that contain these pieces of information. In this way all memories associated with a concept(s) can be retrieved and decoded in parallel.

The system, 3DViews was created to provide visualization of oligo interactions. A tiled display system was built to increase the size of the total display without giving up fine detail.

This Air Force Office of Scientific Research funded project was performed in a collaboration project between Air Force researchers and contractors at the Air Force Research Laboratory in Rome, New York. Air Force researchers focused on the Visualization modeling and 3D Views work described in Section 4.6. This report describes results for the complete AFOSR project.

2.0 INTRODUCTION

In [1]-[7] it has been shown that the hybridization that occurs between a DNA strand and its Watson-Crick complement can be used to perform mathematical computation. This research addresses how the massive parallelism of DNA hybridization reactions can be exploited to construct a DNA based associative memory.

Single strands of DNA are polymers of nucleotide bases adenine (A), cytosine (C), guanine (G) and thymine (T) and thus can be represented by sequences of the letters A, C, G, and T. DNA sequences have an orientation that reflects the asymmetric covalent linking between consecutive bases in the DNA strand backbone; e.g., 5'AACG3' is distinct from 5'GCAA3', but it is identical to 3'GCAA5'.

DNA can be single-stranded (ssDNA) or double-stranded (dsDNA). ssDNA most easily forms into a double-stranded helix with its oppositely directed reverse complement. To obtain the 3'→5' reverse complement of a 5'→3' strand of DNA, substitute A with T and C with G and vice-versa. For example, the 3'→5' reverse complement of 5'TCGCA3' is 3'AGCGT5'. If x is a DNA sequence, then let \bar{x} denote its reverse complement in the opposing 3'→5' direction. For example $\bar{5'TCGCA3'} = 3'AGCGT5'$. Henceforth, strands without strikethrough are 5'→3' and strands with strikethrough are 3'→5'. A dsDNA duplex formed between a strand and its reverse complement is called a Watson-Crick (WC) duplex, e.g., $\begin{array}{c} \text{TCGCA} \\ \bar{\text{TEGCA}} \end{array}$. Note that non-WC duplexes can form and such a formation is called a *cross-hybridization*. Cross-hybridizations are undesirable and there is a need to carefully design the synthetic DNA to ensure that a cross-hybridization never happens. The length of ssDNA or a dsDNA WC duplex is the number of bases or base pairs (bp) respectively in the strand. For example, TCGCA is called a 5-mer (mer is short for *polymer*) and the length of the WC duplex $\begin{array}{c} \text{TCGCA} \\ \bar{\text{TEGCA}} \end{array}$ is 5bp. See Figure 1.

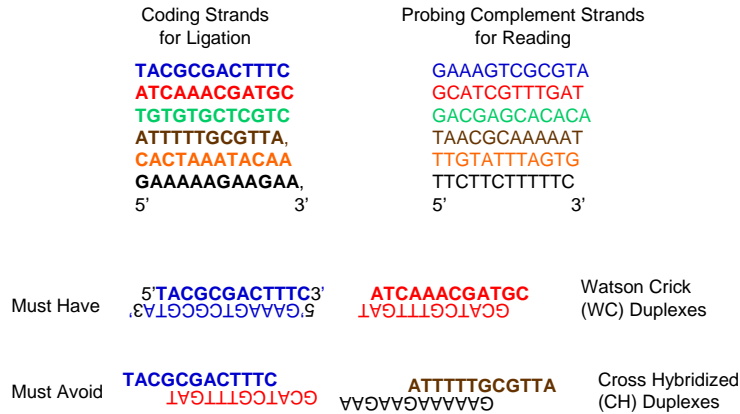


Figure 1: A DNA Code

Hybridization assays offer the possibility of *simultaneously* processing trillions of bits of information. In DNA hybridization assays for biomolecular computing, concatenated DNA strands can be used for multiple purposes. They can be used to store, write, read and retrieve information. Hybridization assays with DNA strands are also used to separate, manipulate, identify and address molecules in many other important experiments beyond biomolecular computing [9]-[11]. Figure 2 shows a CAM concept using DNA for the probe and store.

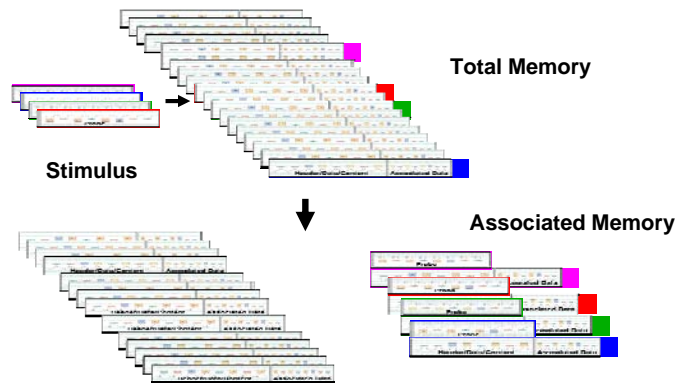


Figure 2: Scheme for Parallel Search for Multiple Associations in DNA

In DNA biomolecular computing, occasions can arise where a sample containing several distinct sequences of DNA needs to be analyzed. For example, each individual sequence in a mixture of DNA could:

- (i) encode a solution to a mathematical problem [12].
- (ii) be stored information associated to an entity [13].
- (iii) be a taggant or label associated to a target [14].

In these cases, the composition of each DNA strand in mixture needs to be determined so that each mathematical solution, memory and/or target can be respectively retrieved. This research shows how a single and parallel battery of reactions performed on a mixed DNA sample containing an arbitrary subset of several double stranded DNA sequences taken can be used to determine the composition of each sequence in the mixture.

Further, this research demonstrates that the combinatorial method employed can feasibly yield billions of covert and synthetic DNA memory strands that carry object and process information. A key component of the innovation is the combinatorial method of bio-memory design and detection that encodes product, item or process information as a numerical sequence represented in DNA. This DNA data structure can be read by the wet laboratory method *polymerase chain reaction* (PCR) (that can also be converted into an electrical signal) and then algorithmically decoded to retrieve virtually an unlimited amount of item or process information that has been stored in the combinatorial memories. In Figure 3, data is encoded using DNA substrands with the whole library strand containing related associations, i.e., "a memory."

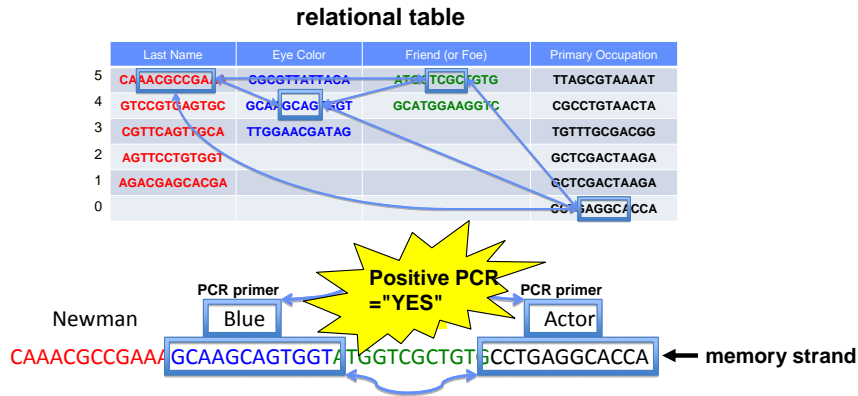


Figure 3: The DNA Associative Array Relational Table

The system 3DViews was created to provide visualization of oligo interactions. A model was created to represent the physicality of oligos, their structure, movement and interactions. A user interface was created to graphically display the model results. For DNA libraries that are large enough to be of interest the model graphics output produces large and detailed images. Current computer screens don't have sufficient pixel densities to display the number of details desired from a simulation. A tiled display system was built to increase the size of the total display without giving up fine detail.

3.0 METHODS, ASSUMPTIONS, PROCEDURES

3.1 Numerical Sequences Represented in DNA

Throughout the remainder of this report all lower case variables are natural numbers, e.g., n , q , s , and t .

A fixed set of $n \cdot q$ relatively short t -mers of ssDNA is called a *t-DNA $n \times q$ table code* and is denoted by $\text{DNA_TC}(n,q,t)$. See Figure 4 for an example of a $\text{DNA_TC}(5,2,10)$ where n is positions along the long strand, q is the number of rows and t is the length of the substrand. The sequences in a given $\text{DNA_TC}(n,q,t)$ are called *table-mers*. A *ssDNA memory library* is the collection of q^n relatively long $n \cdot t$ -mers strands of ssDNA that are concatenated from a fixed $\text{DNA_TC}(n,q,t)$. A member of a ssDNA memory library is called a ssDNA memory.

Key Idea: Any finite numeric sequence can be encoded as a ssDNA (or dsDNA) memory and vice-versa.

For example, using the table-mers from Figure 4, the binary sequence 01101 is encoded as **CGTCCATCGT** **CGCAAGCTGA** AGTGGATGCG TCGGTAAGCG TCGGAGTGCT. This encoding is possible because only certain collections (partitioned by font type) of sequences are allowed to be in each position (e.g., Arial = position 0, Comic = position 1, etc.) and within each collection, distinct strands are assigned distinct numerical values (e.g., **CGTCCATCGT** = 0, **GCAGAAGCCA** = 1 for position 0). It is straightforward to see that table-mers can be used to make a table that in turn can be concatenated to make q^n distinct longer DNA memories encoding each numeric sequence with n digit positions where each digit can range from 0 to $q-1$.

	position 0	position 1	position 2	position 3	position 4
0	CGTCCATCGT	<i>CATTCGCGGA</i>	ACAGTTGCCG	TCGGTAAGCG	GAGCGAACCA
1	GCAGAAGCCA	<i>CGCAAGCTGA</i>	AGTGGATGCG	TGCACGAGAC	TCGGAGTGCT

Figure 4: A $\text{DNA_TC}(5,2,10)$

Each table-mer in DNA_TC(5,2,10) in Figure 4 can be labeled by an ordered pair (position, value). The first coordinate corresponds to the position and the second coordinate corresponds to the value. Font type only indicates position. For example (0,1)= **GCAGAAGCCA**, while (2,0)= ACAGTTGCCG.

For every ssDNA memory there is a corresponding dsDNA memory that is the unique WC duplex that contains the ssDNA memory. See Figure 5.

```
CGTCCATCGT CGCAAGCTGA AGTGGATGCG TCGGTAAGCG TCGGAGTGCT
CGTCCATCGT CGCAAGCTGA AGTGGATGCG TCGGTAAGCG TCGGAGTGCT
```

Figure 5: A "longer strand" double-stranded WC ComDMem

Henceforth, a ssDNA memory is identified with the unique WC dsDNA memory that contains it and the term *DNA memory* henceforth means WC dsDNA memory. For a given DNA_TC(n,q,t) table code M, let MEM_LIB(n,q,n·t) of M denote the collection of q^n possible double-stranded $n \cdot t$ bp memories that can be formed by concatenation, where each DNA memory is identified by its top 5'→3' strand. For example, the DNA memory in Figure 3 is a member of MEM_LIB(5,2,50) of Figure 4.

3.2 Polymerase Chain Reaction Laboratory Method

Polymerase chain reaction (PCR) is a technique widely used in molecular biology, forensic science, environmental science, and many other areas [15-16]. Briefly, PCR is a test tube system that exponentially replicates a substrand of a DNA memory that is delimited by two sequence specific recognition sites (e.g., table-mers) which are found at the ends of the substrand to be selectively amplified. By incubating a DNA memory mixture with oligonucleotide recognition site PCR primers and the enzyme DNA polymerase, the presence of a pair of recognition sites on a common substrand of a DNA memory can be determined by whether or not a PCR amplification occurs.

Key Idea: This PCR amplification information can be mathematically exploited to decode layered memories.

A standard method for detection of amplification involves an electrical separation and detection of DNA substrands on a size separation media called a gel. There are other more sensitive and faster (e.g., real-time PCR) methods that automate the entire PCR protocol and can detect amplification. These instruments can very reliably provide the information needed to conduct the mathematical algorithms in a cost effective manner.

3.3 Memory Design and Synthetic DNA Code SynDCode Software

The decoding accuracy of DNA memories by the PCR method depends upon whether or not so-called *false priming sites* exist in the memories. The priming sites for this method are the table-mers used to construct the memories. False priming site sequences can arise if two or more of the table-mers are too similar or if the memory sequence regions that overlap the junctions where table-mers are concatenated are too similar to the original table sequences.

The synthetic DNA code software, *SynDCode* [17] is a tool developed to design synthetic DNA sequences to be used in biologically based information systems (e.g., DNA computing, DNA memory, DNA nanodevices and DNA memories). *SynDCode* allows for the specification of thermodynamic distance and dissimilarity so that the synthetic table-mers (and their complements) do not create false priming sites. The table-mers in Figure 4 were designed by *SynDCode* to be non-complementary and non-cross-hybridizing so that each position in a memory library strand will be (ultra) specific for a unique PCR primer. The fact that *SynDCode* gives non-cross-hybridizing output has been experimentally verified repeatedly in the laboratory. Enhanced *SynDCode* strand design optimization methods were developed in [25-28].

3.4 The PCR Signal and PCR Network Graph

As a small example, consider the table-mers in Figure 4 and all 32 distinct (one for each 0, 1 string of length 5) memories in $\text{MEM_LIB}(5,2,50)$ formed from Figure 4. For the general

memory library $\text{MEM_LIB}(n,q,n-t)$, there are $\frac{n \cdot (n-1) \cdot q^2}{2}$ primer pairs of table-mers and thus the same number of distinct PCR reactions with each memory being positive for exactly $\frac{n \cdot (n-1) \cdot q^2}{2}$ of them. In the above example, $n = 5$ and $q = 2$, so there are 40 distinct PCR reactions to perform with any given memory being positive for 10 of them. Forty may seem like many reactions, but current PCR technology allows for 768 simultaneous reactions (e.g., Applied Biosystems AutoLid Dual 384-Well GeneAmp® PCR System 9700).

Figure 6(a) below is a graphical interpretation, called a *PCR network graph*, of all possible PCR reactions from primer pairs of table-mers from Table 1. The lines connecting the nodes in the graph denote all possible primer pairs. Notice that there are no lines between primer pairs with the same first coordinate (e.g., (4,0) and (4,1)). This is because no single memory can have two distinct table-mers at the same position. In Figure 6(b), the set of bold lines denotes the set of positive PCR reactions for the DNA memory represented by 01101.

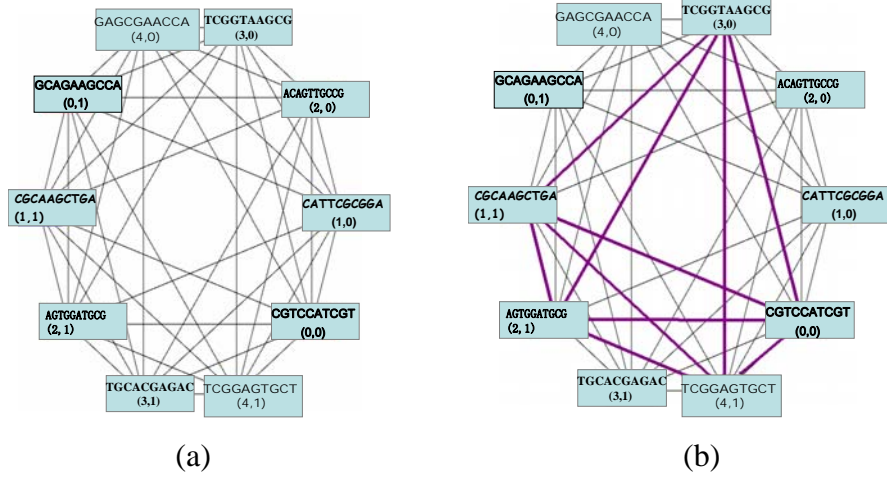


Figure 6: PCR Network Graph

Key Idea: By using smaller DNA fragments that mathematically constitute what is known as a combinatorial cover (hence the name *ComDMem*) the same PCR network graph information can be obtained that would be received from a longer DNA memory.

Let M be a fixed collection of table-mers $\text{DNA_TC}(n,q,t)$. An s -DNA cover of M is a collection of double-stranded WC duplexes concatenations of s table-mers taken from M that yield *exactly* all the same positive PCR reactions that exist for the entire memory library $\text{MEM_LIB}(n,q,n \cdot t)$ for M . A DNA sequence in an s -DNA cover is called a *covering strand*. Note, since the length of such a covering strand is $s \cdot t$ bp, then the s -DNA cover of M is called a $\text{COV_DNA}(n,q,s \cdot t)$ of M . A $\text{COV_DNA}(n,q,s \cdot t)$ of M is also referred to as an s -DNA cover of the memory library $\text{MEM_LIB}(n,q,n \cdot t)$ constructed from M .

Key Idea: By using DNA covers of DNA memory libraries, a virtual memory can be constructed, i.e., *ComDMems*, that behave exactly like real (and longer) memories in the library with respect to their PCR signal. Thus, for $\text{MEM_LIB}(n,q,n \cdot t)$, instead of having to painstakingly construct q^n memories, one can construct approximately q^s strands in $\text{COV_DNA}(n,q,s \cdot t)$ and get the same results by algorithmic mixing to make the *ComDMems*. This amounts to a feasible q^{n-s} fold cost

reduction. For example, with $n = 10$, $q = 2$, $s = 3$, the reduction is approximately 100 fold. Moreover, the physical construction of long DNA memory sequences when $n \cdot t$ is greater than 200 is virtually impossible. Thus, to get massive amounts of data storage capability, $COV_DNA(n, q, s \cdot t)$ must be used.

For example, consider the $MEM_LIB(5, 2, 50)$ constructed from Table 1 and let C be a $COV_DNA(5, 2, 30)$ 3-cover. The four covering strands cs_1 , cs_2 , cs_3 and cs_4 in C that appear below in Figure 7 together constitute a virtual $ComDMem$ memory for the actual memory that appears in Figure 5 above.





$CS_1 =$ CGTCCATCGT CGCAAGCTGA AGTGGATGCG GGTCCATCGT CGCAAGCTGA AGTGGATGCG 	$CS_2 =$ CGTCCATCGT CGCAAGCTGA TCGGAGTGCT GGTCCATCGT CGCAAGCTGA TCGGAGTGCT 
$CS_3 =$ AGTGGATGCG TCGGTAAGCG TCGGAGTGCT AGTGGATGCG TCGGTAAGCG TCGGAGTGCT 	$CS_4 =$ CGTCCATCGT CGCAAGCTGA TCGGTAAGCG GGTCCATCGT CGCAAGCTGA TCGGTAAGCG 

Figure 7: Covering Strands for Memory in Figure 5

The virtual aspect of the collection of the four covering strands can be observed in Figure 8. Each of the four covering strands gives rise to three positive PCR reactions. For example, cs_3 has positive PCR reactions for the primer pairs in the triangle (3,0), (4,1), (2,1) whose lines are shaded with the shorter dashes (---). The triangle of edges that are positive for each covering strand cs_i are shaded according to the line type associated with cs_i in Figure 7. Note that the line between (0,0) and (1,1) appears in three of the four triangles and is thus partially highlighted by three different shadings. Comparing Figure 8 to Figure 6(b), it can be observed that cs_1 , cs_2 , cs_3 and cs_4 in total give the same ten positive PCR reactions as does the single longer memory that they cover.

Key Idea: From the point of view of the positive PCR reactions, the single longer memory is indistinguishable from the mixture of the covering strands, i.e., the virtual memory ComDMem.

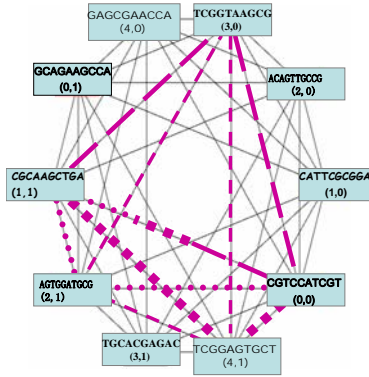


Figure 8: PCR Results from Covering Strands in Figure 7

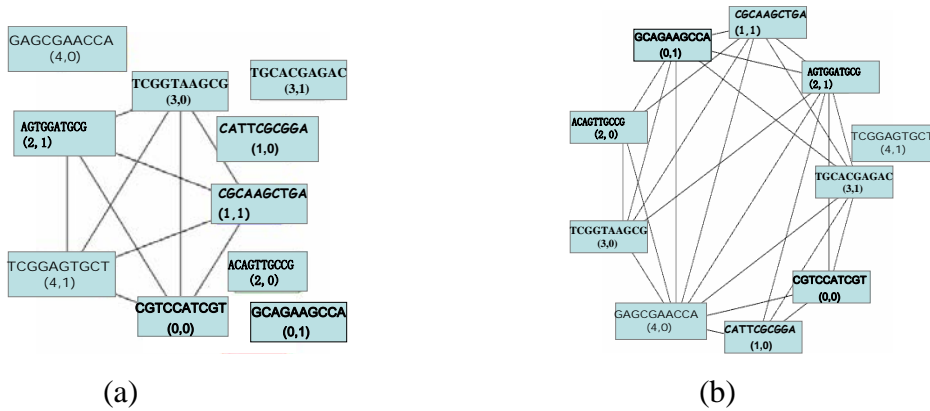


Figure 9: PCR Graphs of Solely Positive PCR Reactions

When a graphical representation of PCR reactions is given, only lines that denote positive PCR reactions need to be given. Using this representation Figure 6(b) becomes Figure 9(a).

Using the binary representation of MEM_LIB(5,2,50), Figure 9(b) gives the positive PCR reactions for the group 11000, 00110, 11100 and 11110 of four layered memories. Note that theoretically, Figure 9(b) would be the same if either the four actual MEM_LIB(5,2,50) sequences 11000, 00110, 11100 and 11110 of 50bp, or the sixteen covering strands of 30bp in COV_DNA(5,2,30) that covered each of the memories 11000, 00110, 11100 and 11110, were combined.

Key Idea: The physical manufacture of all the DNA memories in a MEM_LIB($n, q, n \cdot t$) is an extremely costly, low yield and sometimes impossible endeavor especially for large n and t . With this combinatorial innovation, one can get the same benefits by using COV_DNA($n, q, s \cdot t$) with a feasible q^{n-s} fold reduction in cost

3.5 Oligo Visualization with 3DViews

This task in the AFOSR project helped bridge the gap between the virtual design and expected interaction of short DNA strands and the physical implementation and real interactions in a physical experiment. A physical model of the DNA strand and strand to strand interactions was created. A graphical user interface was created to allow designers to visualize the complex physical structures and interactions of DNA systems. A large scale tiled computer display system was built to provide the large display area with high pixel resolution needed to display the DNA interactions. A tiled display was created because it provides a larger viewing area while not giving up resolution in the number of pixels per area. In this manner, a 5 x 5 monitor tiled array has 25 times as many pixels and 25 times the image resolution as a monitor that is 25 times bigger in area than the original monitor but has the same number of total pixels. Tiled monitor technology is becoming important of object visualization for many complex informatics applications. The completed hardware / software / interaction model system was called 3DViews and is described in the results section.

4.0 RESULTS, DISCUSSION

4.1 The Mathematical Model

For $2 \leq n, q$ let $V_{n,q}$ be the set of all ordered pairs (p, v_p) where $p \in [n]$ and $v_p \in [q]$. An n -set in $V_{n,q}$, $\{(p, v_p)\}_{p \in [n]}$, where the first coordinates are distinct, can be uniquely identified with an element of $[q]^n$ and vice-versa. Under this bijection, each $\tau \in [q]^n$, $\tau = v_0 \dots v_{n-1}$, is identified with the n -set of ordered pairs in $V_{n,q}$, $\tau = \{(0, v_0), \dots, (n-1, v_{n-1})\}$, where the first coordinate designates the position in the sequence and the second coordinate represents the value at that position. For example $\{(2,0), (0,1), (1,3)\}$ corresponds to 130. Henceforth, $[q]^n$ denotes n -sets in $V_{n,q}$ where the first coordinates are distinct.

$E_{n,q}$ denotes the set of all pairs $\{(p_1, v_1), (p_2, v_2)\}$ in $V_{n,q}$ where $p_1 \neq p_2$. Then $E_{n,q}$ is the set of all edges in the q -partite graph $G_{n,q}$ on the vertex set $V_{n,q}$ where the independent sets are collections of vertices with the same first coordinate. Further identify $\tau = \{(0, v_0), \dots, (n-1, v_{n-1})\}$ with the complete subgraph, denoted K_τ , of $G_{n,q}$ on the vertices in $\tau \in [q]^n$.

The correspondence between the mathematical and physical entities is as follows: $V_{n,q}$ is identified with S , $\text{MEM_LIB}(n, q, n \cdot t)$ is identified with $[q]^n$ and $E_{n,q}$ is identified with all possible PCR reactions. This latter identification is less obvious than the others. A pair of primers v_{p_1}, \bar{v}_{p_2} where $0 \leq p_1 < p_2 \leq n-1$ corresponds to a unique PCR reaction. Then identifying $\{v_{p_1}, \bar{v}_{p_2}\}$ with $\{v_{p_1}, v_{p_2}\}$, the identification of $E_{n,q}$ and PCR reactions is observed.

Using these identifications, given a pool of sequences, $U = \{\tau_1, \dots, \tau_k\}$, from $[q]^n$, consider an edge $e = \{(p_1, v_1), (p_2, v_2)\}$ in $E_{n,q}$. Say that e is *positive for U* if and only if there is a $\tau \in U$ such that τ has value v_1 in position p_1 and value v_2 in position p_2 . Considering U as a pool of dsDNA strands taken from $\text{MEM_LIB}(n, q, n \cdot t)$ and considering e as the PCR reaction for primers v_{p_1}, \bar{v}_{p_2} , then e is positive for pool U if and only if an exponential amplification results from exposing the sample U to the PCR reaction v_{p_1}, \bar{v}_{p_2} . Experimentally, this exponential amplification can be observed in many ways. Some of these ways are described as conventional gel based and SYBR green and/or Taqman based real-time PCR [12], [16].

Finally given a pool of sequences from $[q]^n$, $U = \{\tau_1, \dots, \tau_k\}$, let G_U denote the subgraph of $G_{n,q}$ that consists of all the edges positive for U . G_U is the graph-theoretic union of the complete subgraphs $\{K_\tau\}_{\tau \in U}$. If U is considered to be a pool of dsDNA strands taken from $\text{MEM_LIB}(n, q, n \cdot t)$, then G_U is identified with the collection of all positive PCR reactions taken over all possible pairs of primers v_{p_1}, \bar{v}_{p_2} . In either the mathematical or physical setting, the goal is to identify U given G_U . The interesting applications come from the fact that G_U can be obtained from experimentation without the direct knowledge of the contents of U .

Consider the set of strands S given in Figure 4. A description of the actual physical construction of dsDNA library $\text{MEM_LIB}(5, 2, 50)$ appears in [8]. Suppose a pool U taken from $\text{MEM_LIB}(5, 2, 50)$ consists of the duplexes identified by 11000, 00110, 11100 and 11110. Then G_U is given in Figure 10, the graph G_U depicting all the positive PCR reactions from Figure 4.

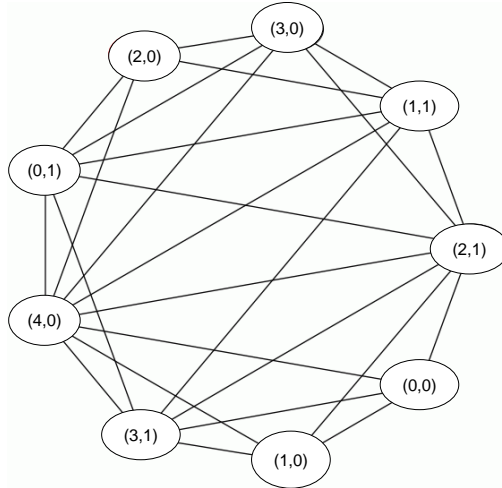


Figure 10: PCR graph G_U with $U=\{11000, 00110, 11100, 11110\}$.

A closer look at an aspect of Figure 4 can aid the discussion. Consider the edge $\{(1,1), (4,0)\}$. From Table 2, this edge corresponds to the PCR reaction primed by $s_{1,1} = \text{TCACACACACACACAATT}$ and the complement of $s_{4,0} = \text{TCTCCTCTCCACTCAAACC}$. This PCR reaction yields amplification because the dsDNA strands 11000 and 11100 are members of U that have the values 1 and 0 in the 1st and 4th positions respectively. (The position count starts with 0.) Note that the PCR reaction primed by $s_{0,0} = \text{CCAAACCTCCACTTTCCAAC}$ and the complement of $s_{2,0} = \text{CCTTTCCTCCATCACCTCAT}$, corresponding to edge $\{(0,0), (2,0)\}$ does not yield an amplification because no strand in $U = \{11000, 00110, 11100, 11110\}$ has the value 0 in both the 0th and 2nd positions.

4.2 The Identification Algorithms

To identify U from G_U , two approaches are taken. The methods are generalizations of those of combinatorial group testing [18-24]. The first is called the *disjunct algorithm* that identifies the strands in $MEM_LIB(n, q, n \cdot t)$ that are surely not in U . The second is called *edge representative decoding* that identifies the strands in $MEM_LIB(n, q, n \cdot t)$ that surely are in U .

Call the disjoint sets of strands identified by these algorithms the *resolved positives* and *resolved negatives*, denoted RP and RN respectively. From the definitions of these sets, then:

$$RP \subset U \subset L_{n,q}(S) - RN . \quad (1)$$

Hence, if $RP = L_{n,q}(S) - RN$, then $U = RP = L_{n,q}(S) - RN$.

The disjunct algorithm is simple to state:

Disjunct Algorithm: Any sequence $\tau \in [q]^n$, thought of as a complete subgraph K_τ in $G_{n,q}$, that has an edge that *does not appear* in G_U is a member of RN .

The disjunct algorithm works because every edge of every $\tau \in U$ corresponds to a positive PCR reaction.

The edge representative decoding is a little more complicated.

Edge Representative Decoding: Any sequence $\tau \in [q]^n$, thought of as a complete subgraph K_τ in $G_{n,q}$, that is also a complete subgraph in G_U and that has an edge that is not contained in any other complete subgraph $K_{\tau'}$ in G_U is a member of RP. In other words, $\tau \in RP$ if and only if K_τ is a complete subgraph of G_U that has an edge that is not contained in any other complete subgraph $K_{\tau'}$ in G_U with $K_{\tau'} \neq K_\tau$.

Edge representative decoding works because every edge in G_U is contained in a complete subgraph K_τ for $\tau \in U$. Thus if an edge in G_U is contained in a unique complete subgraph of G_U , then that subgraph must be K_τ for some $\tau \in U$.

4.3 Algorithmic Implementation

In this section the graph theoretic algorithms on the abstract PCR graph that implement in the PCR data decoding software are given.

Consider information storage as a set of data values, $S = \{s_k\}$. Each data value s_k is a ComDMem and each ComDMem is a set of ordered pairs, i.e., $s_k = \{(i, j)\}_{\substack{0 \leq i < N \\ 0 \leq j < q}}$ where each ordered pair (i, j) is a table-mer. The fundamental question now becomes: how to effectively implement the representative decoding so that S can be found.

The algorithms for reconstructing S are graph-theoretic in nature, so one must reformulate the problem. Let $G^0 = (V^0, E^0)$ be our PCR graph, i.e., vertices are table-mers, edges are positive PCR reactions and a ComDMem is a clique of size N .

Let $G^t = (V^t, E^t)$ be the graph with the following properties:

- 1) Each vertex $v \in V$ has associated with it a set of pairs $\{(i, j)\}$ with $0 \leq i < q$ and $0 \leq j < N$.
- 2) If (i, j) and (k, l) are in the set associated with a vertex $v \in V^t$, then $i \neq k$.
- 3) If the vertices of an edge $(v, w) \in E^t$ have associated sets of pairs p_v and p_w , and $(i, j), (k, l) \in p_v \cup p_w$, then $i \neq k$.

This $G^t = (V^t, E^t)$ graph type is an extension of the PCR graph $G^0 = (V^0, E^0)$. Figure 11 illustrates the extension. Instead of each node having just one pair (i, j) , it has a set of pairs $\{(i, j)\}$. To make viewing easier, each set of pairs is represented by an N -tuple. For example, $\{(2, 0)\}$ is represented by $(*, *, 0)$ which represent an unfolding ComDMem. Each vertex $v \in V^t$ is equivalent to subset of the pairs from a data entry $s_k \in S$, in other words a partial or fuzzy ComDMem. Each edge is equivalent to positive PCR reactions. One can construct a sequence of graphs satisfying the properties above using Algorithm 1 given in Figure 12

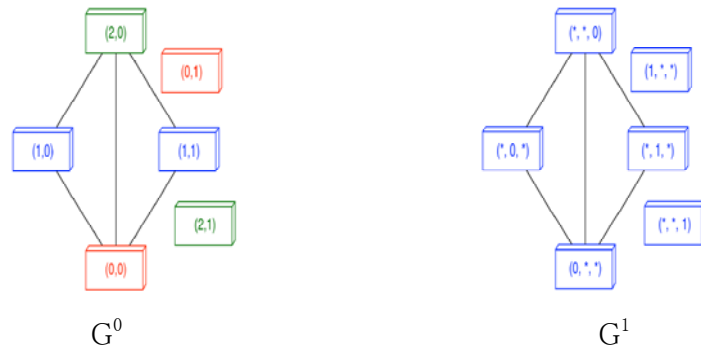


Figure 11: Gt-1 to Gt Graph Extension Scheme


```

Let  $p[w]$  be the sets of pairs associated with node  $w$ .
//Each edge in  $G^{t-1}$  generates a node in  $G^t$ 
for all  $(u^{t-1}, v^{t-1}) \in E^{t-1}$  do
    Create a new node  $w^t \in V^t$ .
     $p[w^t] \leftarrow p[u^{t-1}] \cup p[v^{t-1}]$ 
end for
//Small cliques in  $G^{t-1}$  generates edges in  $G^t$ 
for all  $(u^{t-1}, v^{t-1}) \in E^{t-1}$  do
    // $N_1 \subseteq V^{t-1}$  are the neighbors of  $u^{t-1}$ .
     $N_1 \leftarrow \emptyset$ 
    for all  $w_1^{t-1} \in V^{t-1}$  such that  $(u^{t-1}, w_1^{t-1}) \in E^{t-1}$  do
         $N_1 \leftarrow N_1 \cup \{w_1^{t-1}\}$ 
    end for
    // $N_2 \subseteq V^{t-1}$  are the neighbors of  $v^{t-1}$ .
     $N_2 \leftarrow \emptyset$ 
    for all  $w_2^{t-1} \in V^{t-1}$  such that  $(v^{t-1}, w_2^{t-1}) \in E^{t-1}$  do
         $N_2 \leftarrow N_2 \cup \{w_2^{t-1}\}$ 
    end for
    // $N \subseteq V^{t-1}$  are the neighbors of both  $u^{t-1}$  and  $v^{t-1}$ .
     $N \leftarrow N_1 \cap N_2$ 
     $w^t \leftarrow$  node in  $V^t$  created from  $(u^{t-1}, v^{t-1})$ 
    for all  $w^{t-1} \in N \subseteq V^{t-1}$  do
        // $u^{t-1}, v^{t-1}, w^{t-1} \in V^{t-1}$  form a 3-clique.
         $u^t \leftarrow$  node in  $V^t$  created from  $(u^{t-1}, w^{t-1})$ 
         $v^t \leftarrow$  node in  $V^t$  created from  $(v^{t-1}, w^{t-1})$ 
         $E^t \leftarrow E^t \cup (u^t, w^t)$ 
         $E^t \leftarrow E^t \cup (v^t, w^t)$ 
        for all  $x^{t-1} \in N - \{w^{t-1}\} \subseteq V^{t-1}$  do
            if  $(w^{t-1}, x^{t-1}) \in E^{t-1}$  then
                // $u^{t-1}, v^{t-1}, w^{t-1}, x^{t-1} \in V^{t-1}$  form a 4-clique.
                 $x^t \leftarrow$  node in  $V^t$  created from  $(x^{t-1}, w^{t-1})$ 
                 $E^t \leftarrow E^t \cup (x^t, w^t)$ 
            end if
        end for
    end for
end for
end for

```

Figure 12: G^{t-1} to G^t Graph Extension Scheme Algorithm 1

By starting the sequence with the original PCR graph G^0 , Algorithm 1, this is a means of finding all ComDMem cliques in G . The idea is that each edge in G^{t-1} generates a node in G^t . Two nodes have an edge in G^{t-1} if its constituent edges from G^t form a clique of size 3 or 4. Figure 13 illustrates the application of Algorithm 1.

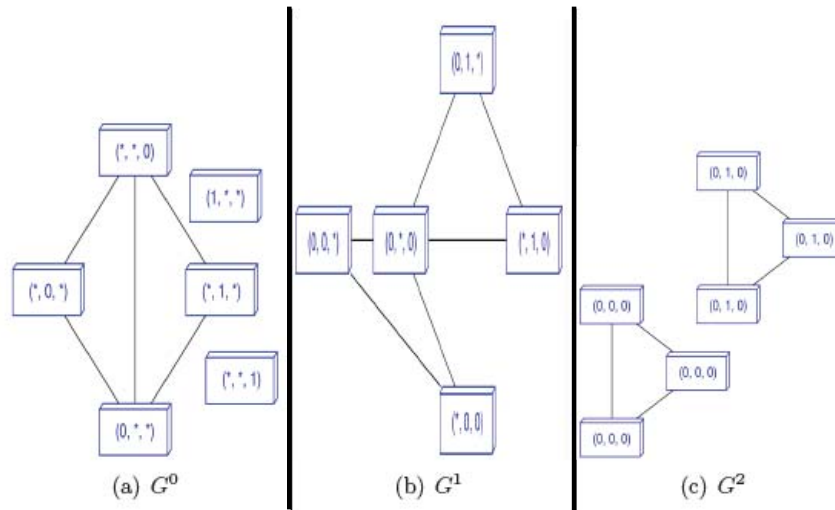


Figure 13: How Algorithm 1 Leads to ComDMem Decoding

Successive application of Algorithm 1 would eventually lead us to all the cliques in the original graph G^0 , but at great computational cost. Instead, by applying the unique edge representative method, one can take advantage of the fact that G^0 was constructed as the graph-theoretic union of cliques of size N e.g., ComDMems. Examine the edge between $(*, 0, *)$ and $(*, *, 0)$ in G^0 from Figure 13. The nodes and the intersection of their neighbors form exactly one data entry $(0, 0, 0)$. Since each edge comes from at least one data entry, this means that a ComDMem has been found. This edge searching algorithm, presented in Algorithm 2, allows us to find entries in S early in the sequence of graphs G^0, G^1, \dots . Data entries found this way in G^0 must be in the original data set S .

```

Let  $p[w]$  be the sets of pairs associated with node  $w$ .
 $S_e^t \leftarrow \emptyset$ 
for all  $(u^t, v^t) \in E^t$  do
   $P \leftarrow p[u^t] \cup p[v^t]$ 
  for all  $w^t \in V^t$  such that  $(u^t, w^t), (v^t, w^t) \in E^t$  do
     $P \leftarrow P \cup p[w^t]$ 
  end for
  if  $|P| = N$  and  $(i, j), (l, m) \in P \implies i \neq l$  then
     $S_e^t \leftarrow S_e^t \cup P$ 
  end if
end for

```

Figure 14: Unique Edge Representative Computational Cost Reduction Algorithm 2

4.4 The Algorithms Applied to Physical Experimental Data

To exhibit the above algorithms, actual dsDNA experiments were performed on MEM_LIB(5,2,50) for Figure 4. A description of the physical construction of this MEM_LIB(5,2,50) appears in [8]. From MEM_LIB(5,2,50), the four sequences that were selected and taken as U are given in Figure 10. To actually select strands from this library, a cloning method was used. The library was amplified with outside primers, the amplified product was cut with BamHI and HindIII, the expected fragment was purified and then ligated into the vector pBluescript [8]. Four of a total of 12 isolated clones were selected to be the pooled sample U. Before these four strands (clones) were pooled, the individual dsDNA were sequenced to determine which library members were actually selected. To exhibit the experimental design and analysis, an incidence matrix is useful and is given in Figure 15. In the actual experiments, essentially no PCR errors occurred and the empirical outcomes seen in Figure 15 were in 100% agreement with the theoretical outcomes that can be founded in the last two rows of Table 15. This is a testament to the SynDCode design method. A portion of gel output of the actual PCR experiments performed on this U is given in Figure 16.

		SEQUENCES																																POSITIVE PCR	NEGATIVE PCR					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32							
		m(5)	m(9)	RP(1)	m(14)	m(2)					m(3)					RP(7)	m(16)	m(30)	m(16)	RP(31)	m(16)	RP(12)	m(16)																	
PCR REACTIONS	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(1,0)}					
	2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(1,1)}			
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	{(0,1),(1,0)}			
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	{(0,1),(1,1)}		
	5	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(2,0)}		
	6	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(2,1)}		
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	{(0,1),(2,0)}		
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	{(0,1),(2,1)}		
	9	1	1	0	0	1	1	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(3,0)}	
	10	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(3,1)}	
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	{(0,1),(3,0)}		
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1	1	1	{(0,1),(3,1)}		
	13	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(4,0)}	
	14	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(0,0),(4,1)}	
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	{(0,1),(4,0)}		
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	{(0,1),(4,1)}		
	17	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	{(1,0),(2,0)}		
	18	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	{(1,0),(2,1)}		
	19	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	{(1,1),(2,0)}	
	20	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	{(1,1),(2,1)}		
	21	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	{(1,0),(3,0)}		
	22	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	{(1,0),(3,1)}		
	23	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	{(1,1),(3,0)}		
	24	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	{(1,1),(3,1)}		
	25	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	{(1,0),(4,0)}		
	26	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	{(1,0),(4,1)}		
	27	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	{(1,1),(4,0)}		
	28	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	{(1,1),(4,1)}		
	29	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	{(2,0),(3,0)}		
	30	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	{(2,0),(3,1)}		
	31	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	{(2,1),(3,0)}		
	32	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	{(2,1),(3,1)}		
	33	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	{(2,0),(4,0)}		
	34	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	{(2,0),(4,1)}		
	35	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	{(2,1),(4,0)}		
	36	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	{(3,1),(4,0)}		
	37	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	{(3,0),(4,0)}		
	38	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	{(3,0),(4,1)}		
	39	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	{(3,1),(4,0)}		
	40	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	{(3,1),(4,1)}		

Figure 15: Record of Actual PCR Results

The sequences in MEM_LIB(5, 2, 50) are given vertically as labels for the columns of the incidence matrix in Table 3 and are numbered 1-32. The sequences in U are distinguished by bold faced fonts and are in columns 7, 25, 29, 31. The PCR reactions, i.e., the edges in $G_{5,2}$, correspond to the rows and are numbered 1-40. The edge labels are given in either the positive or negative PCR columns depending upon whether the given edge is positive or negative for U. Every entry in the matrix corresponds to a pair (PCR reaction, sequence). There is a 1 in a given

entry (i,j) if and only if the sequence j is (theoretically) positive for PCR reaction i. Using our mathematical representation, each entry corresponds to a pair (edge, complete subgraph) and there is a 1 in that entry if the given edge is contained in given complete subgraph. The disjunct algorithm uses only the negative PCR reactions which are listed in the last column and the edge representative decoding algorithm uses only the positive PCR reactions which are given in the penultimate column. In the actual experiment, whose raw results can be seen in Figure 5, the pooled dsDNA sample is separately exposed to all forty pairs of PCR primers.

Using Table 15 and focusing on the disjunct algorithm, sequences 9-16 are in RN by virtue of PCR reaction 2 because each of the sequences 9-16 contain PCR reaction 2 as an edge and PCR reaction 2 was negative for the given U. Thus columns 9-16 are labeled rn(2) which is meant to denote that these sequences are in RN by virtue of PCR reaction 2 being negative. Other PCR reactions may also indicate that these sequences are in RN, but PCR reaction 2 is the first in our ordering to do so. Similarly, sequences 17-24 are labeled rn(3), sequences 1-4 are labeled rn(5), sequences 5-6 are labeled rn(9), sequence 8 is labeled rn(14), sequence 26, 28, 30, 32 are labeled rn(16) and sequence 22 is labeled rn(30). Thus $RN = \{1-4, 5-6, 8, 9-16, 17-24, 22, 26, 28, 30, 32\}$.

Using Figure 15 and focusing on the edge representative decoding, sequence 7 is identified as being in RP, because the complete graph K_τ , $\tau = 00110$ is the column 7 label, is the only complete subgraph of G_U that contains the edge $\{(0,0), (1,0)\}$ which denotes the positive PCR reaction 1. Thus column 7 is labeled RP(1) which is meant to denote that this sequence is in RP by virtue of PCR 1 being positive. Other PCR reactions may also indicate that this sequence is in RP, but PCR reaction 1 is the first in our ordering to do so. Similarly sequences 25, 31 and 29 are respectively identified by the positive PCR reactions 7, 12, and 31 and columns 25, 31 and 29 are respectively labeled RP(7), RP(12), RP(31). Since $RP = L_{5,2}(S) - RN$, then

$$U = RP = L_{5,2}(S) - RN.$$

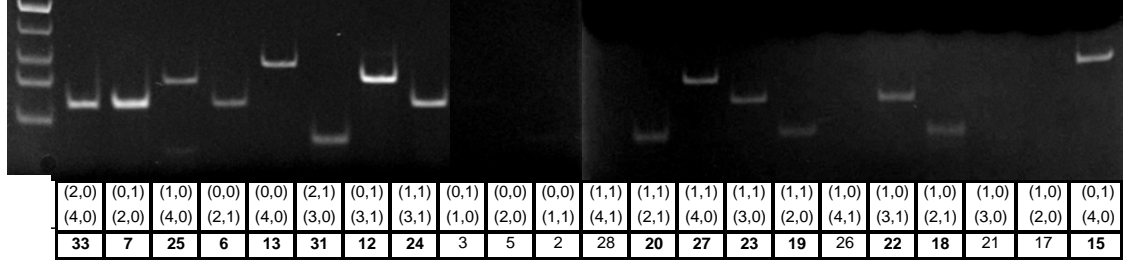


Figure 16: A Portion of the Electrophoresis Gel from the PCR

Figure 16 gives a portion of the electrophoresis gel from the PCR reactions whose positive and negative results are recorded in Figure 15. The lanes where bands can be seen are positive PCR reactions for the encoded primers given at the bottom of the lane. The row number of Figure 15 that corresponds to a lane appears in Figure 16 directly below the encoded primers for the given lane. In all, there were forty separate PCR reactions being primed by all forty primer pairs, each with the same dsDNA sample U. Each reaction occurred in a separate well with each well corresponding to distinct a lane in the gel.

4.5 The General Setting, Parameters and Simulated Performance

In general, the size of MEM_LIB($n, q, n \cdot t$) is q^n and the number of PCR reactions for this library is $\frac{n(n-1)q^2}{2}$. In Figure 17, the outcome of simulated performance is given and compared.

Memory Size, # pairwise associations ($q^n - 6^9$)	$\sim 10^7$
Simultaneous Records Accessed (picked)	20
Simultaneous Associations Accessed @15	300
Accuracy	97%
SynDCode Strands Required (nq)	54
Length of DNA Memory Strands	180
Number of DNA Library Strands	$\sim 10^7$
Number of PCR Reaction Wells	324
Number of PCR Reactions	2916
Computer Clock Cycles Required	$O(10^9)$

DNA data structure requires	Standard computer requires
$O(n^2q^2)$ PCR reactions, $O(nq^2)$ wells	$O(n^2q^n)$ clock cycles

Figure 17: Simulations of Algorithmic Performance

4.6 Visualization with 3DViews

The system 3DViews was created to provide visualization of oligo interactions. A model was created to represent the physicality of oligos, their structure, movement and interactions. A user interface was created to graphically display the model results. For DNA libraries that are large enough to be of interest the model graphics output produces large and detailed images. Current computer screens don't have sufficient pixel densities to display the number of details desired from a simulation and the objects would be too small to see if they did. A tiled display system was built to increase the size of the total display without giving up fine detail.

The oligo shape was modeled as an elongated ellipsoid with short axis a and long axis b . For gross movement this approximation is justified by the rigidity of short oligos and the shape of the polar charge. Oligo movement was modeled by a Brownian motion 3 dimensional random walk. The one dimensional diffusion coefficient D for the ellipsoid shape with 3 independent directions is:

$$D = k_B T \frac{\ln(2b) - \ln(a)}{6\pi\eta b}$$

Where T is temperature, k_B is Boltzmann's constant, and η is the viscosity of the medium. The random walk motion is modeled by assuming the oligo is on a three dimensional lattice and may move a step distance dl in a step time dt . In m time steps, the oligo will move n grid points with equal probability. In random walk, the Brownian motion is approximated by:

$$D = \frac{(n dl)^2}{6 m dt}$$

From these two equations, motion of a group of oligos was mapped through space by the motion model.

Reactions between two or more oligos that land on the same grid point were modeled by assuming a diffuse solution with a Boltzmann distribution in the probability of oligos landing on a grid point. The reaction between multiple oligos landing on the same point was modeled by the Boltzmann distribution for interaction states where the probability P_j of state j is:

$$P_j = \frac{\exp\left(\frac{-\Delta G_j}{k_B T}\right)}{Z}, \text{ where}$$

$$Z = \sum_j \exp\left(\frac{-\Delta G_j}{k_B T}\right)$$

T is the temperature, k_B is Boltzmann's constant, and ΔG_j is the difference between the free energy of the state j. The oligo model and design tool SynDCode was used to approximate ΔG_j [17].

Rendering the model output was a significant issue due to the large number of objects to be displayed on the computer screen. High resolution was needed to view individual hybridization reactions. To view the various kinetics permutations, a large number of grid points were needed. A tiled display system, Mobile Stream Processing Cluster, MOSAIC was built to aid in visualization of the system. The finished modeling cluster and display system is shown in Figure 18. A set of nine 1920 x 1200 pixel monitors were tiled 3 x 3 on a stand which also holds the computer cluster and power supplies. The result was a continuous 5760 x 3600 pixel display.



Figure 18. MOSAIC Cluster

To run the visualization model and drive the display, three 8 core Apple Mac Pros were used with 32GB of RAM each. Red Hat Enterprise Linux v.5x was used for the operating system. Each Mac Pro was given three ATI Radeon graphics cards, one for each monitor in the tile display. The computers were connected with 10Gb Ethernet.

The oligo interaction model run on the cluster creates a continuous series of OpenGL calls that represents the graphical output of the model. The distributed graphics processing application Chromium was used to render the graphical output across the nine displays in real time. The result was a high fidelity physical model of the diffusion and interaction thermodynamics of a large set of oligos and a 9x improvement in resolution in display of the model output.

5.0 CONCLUSIONS

This project developed a synthetic DNA-based associative memory called ComDMems that unlike conventional silicon based associate memories provides for a high degree of input parallelization that allows for a significant reduction in required data structure queries.

This innovation combines mathematics and molecular biology. First, it uses mathematics to design the synthetic DNA that makes the storage of information in ComDMem possible. Then it uses the specificity of DNA strand recognition and the wet laboratory method of polymerase chain reaction (PCR) to store information and to generate a signal. Finally, it uses mathematics to decode the PCR signal and identify the ComDMem signatures and reveals the information and associations they contain.

By using mathematical combinations of short "covering strands" in place of each single and longer memory strand, covert ComDMems can encode a vast amount of information in a more efficient way and that this encoded information can be retrieved only by an authorized user. A uniform method of covering strand construction that minimizes the number of covering strands and theoretically and experimentally mimics the behavior of the longer memories strands was given. This project demonstrated a method of decoding the PCR output that minimizes the number of PCR reactions for given number or distribution of superimposed or associated ComDMems.

These synthetic ComDMems are feasibly functional at concentrations that are below the parts per billion level. Thus, they could not be reverse engineered because their detection would only be possible with prior knowledge of the memory specific DNA sequences required for PCR amplification. Hence, ComDMem synthetic DNA memories are highly covert. ComDMems can encode item or process information as a numerical sequence in DNA, are highly covert, are capable of carrying virtually an unlimited number of data fields, and are deeply super imposable and thus associative.

In general, the decoding of associative DNA memory has been an intractable problem for processes requiring deeply superimposed memories. However, ComDMems are constructed in a sophisticated combinatorial manner so that the decoding of such deeply associative memories is feasible. Thus, beyond being covert and information-rich, our DNA memories can enable design of efficient, scalable and technically useful libraries of synthetic DNA for use in high performance associative memory.

The MOSAIC cluster has been successfully transitioned for use in three projects to date. It is home to Distributed Quantum Computing simulation work where multi-thread and parallel processing are blended to reduce latency and maximize information exchange between the systems. It is also the main demonstration platform for the SWATHBUCKLER project which requires the use of the MOSAIC's nine high-definition displays to view wide area Synthetic Aperture Radar data. Finally, it supports an Air Force Research Laboratory neuromorphic computing camera project which will eventually use the nine tile display to view different algorithmic approaches of computing in a neuromorphic design.

6.0 REFERENCES

1. Adleman, L. M., "Molecular Computation of Solutions to Combinatorial Problems," *Science*, **266**, 1994, pp. 1021–1024.
2. Head, T. and Gal, S. "Aqueous Computing: Writing Into Fluid Memory," *Bulletin of the European Association for Theoretical Computer Science*, **75**, 2001, pp. 190-198.
3. Frutos, A. G. et al. "Demonstration of a Word Design Strategy for DNA Computing on Surfaces," *Nucleic Acids Research*, **25**, 1997, pp. 4748 -4756.
4. Murphy, D., "Gene Expression Studies Using Microarrays: Principles, Problems, and Prospects," *Advances in Physiology Education*, **26**, 2002, pp. 256–270.
5. Winfree, E., et al. "Design and Self-Assembly of Two-Dimensional DNA Crystals," *Nature*, **394**, 1998, pp. 539–544.
6. Braun, E., et al. "DNA-Templated Assembly and Electrode Attachment of a Conducting Silver Wire," *Nature*, **391**, 1998, pp. 775–778.
7. Whitesides G. M. and Boncheva, M., "Beyond Molecules: Self-Assembly of Mesoscopic and Macroscopic Components," *Proc. Natl. Acad. Sci.*, **99**, 2002, pp. 4769–4774.
8. Gal, S., Monteith, N., Macula, A. J., "Successful Preparation and Analysis of a 5-site 2-Variable DNA Library", *Natural Computing*, 8 , 2009, 333 - 347.
9. Brenner, S., "Methods for Sorting Polynucleotides Using Oligonucleotide Tags", U.S. Patent No. 5,604,097, 1997
10. Brenner, S. et al., "Gene Expression Analysis by Massively Parallel Signature Sequencing (MPSS) on Microbead Arrays", *Nat. Biotechnol.*, 18, 2000, pp. 630-634.
11. Cai, H., P. White, D. Torney, A. Deshpande, Z. Wang, B. Marrone, and J. Nolan, "Flow Cytometry-Based Minisequencing: A New Platform for High Throughput Single Nucleotide Polymorphism Scoring", *Genomics*, 66, 2000, pp. 135-143.
12. Ibrahim, Z, et al. "A New Readout Approach in DNA Computing Based on Real-Time PCR with Taqman Probes", (C. Mao and T. Yokomori Eds.), *DNA 12: Lecture Notes in Computer Science 4287*, 2006, pp. 350-359.
13. Yamamoto, M., et al., "Large-Scale DNA Memory Based on the Nested PCR", *Natural Computing*, 7, 2008, pp. 335-346.

14. Hall, B., et al., "Survival and Polymerase Chain Reaction-Based Detection of Nucleic Acid Taggant Markers During Bacterial Growth and Sterilization", *Analytica Chimica Acta*, 475, 2003, pp. 67-73.
15. Mullis, K., et al., "The Polymerase Chain Reaction", Birkhäuser, 1994, Boston
16. Valasek, M. A., Repa, J. J., "The Power of Real-Time PCR". *Advan. Physiol. Edu.* 29, 2005, pp. 151-159.
17. M. A. Bishop, A. J. Macula, T. E. Renz, SynDCode Suite, 2006, <http://syndcode.geneseo.edu>.
18. Du, D. Z. and Hwang, F. K., "Combinatorial Group Testing and Its Applications", 2nd ed. World Scientific, 2000. Singapore.
19. Macula, A. J., "A Simple Construction of d -Disjunct Matrices with Certain Constant Weights", *Discrete Mathematics*, 162, 1996, pp. 311-312.
20. Macula, A. J., "Probabilistic Nonadaptive Group Testing in the Presence of Errors and DNA Library Screening", *Annals of Combinatorics*, 3, 1999, pp. 61-69.
21. Macula, A. J., "Probabilistic Nonadaptive and Two-Stage Group Testing with Relatively Small Pools and DNA Library Screening, *Journal of Combinatorial Optimization*, 2, 1999, pp. 385-397.
22. A. Macula, et al., "Nonadaptive and Trivial Two-Stage Group Testing with d^c -Disjunct Matrices, Entropy Search, and Complexity", *Bolyai Studies*, 16, 2007, pp. 71-84, Springer
24. A. Macula and L. Popyack., "A Group Testing Method for Finding Patterns in Data", *Discrete Appl. Math.* 144, 2004, 149-157.
25. A. Macula, et al., "PCR Nonadaptive Group Testing of DNA Libraries for Biomolecular Computing and Taggant Applications", *Discrete Mathematics, Algorithms and Applications*, Volume: 1, Issue 1, March 2009, pp.59 - 69
26. A. Macula, et al., "Random Coding Bounds for DNA Codes Based on Fibonacci Ensembles of DNA Sequences", 2008 IEEE Proceedings of International Symposium on Information Theory, pp. 2292 - 2296
27. A. Macula, et al., "New, Improved, and Practical k -Stem Sequence Similarity Measures for Probe Design", *Journal of Computational Biology*, 5, June 2008, pp. 525-34.

28. A. Macula, et al., “Random Coding Bounds for DNA Codes Based on Fibonacci Ensembles of DNA Sequences”, 2008 IEEE Proceedings of International Symposium on Information Theory, pp. 2292 – 2296

7.0 ACRONYMS

CAM	Content Addressable Memory
DNA	Deoxyribonucleic Acid
dsDNA	double stranded DNA
MOSAIC	Mobile Stream Processing Cluster
PCR	Polymerase Chain Reaction
RAM	Random Access Memory
ssDNA	single stranded DNA
WC	Watson – Crick
A	Adenine
C	Cytosine
G	Guanine
T	Thymine