DNA Computing
for
Signal Matching

A Thesis

Submitted in Partial Fulfillment of the

Requirement for the Award of the Degree

MASTER OF ENGINEERING
IN
SOFTWARE ENGINEERING

Under the Supervision
of
Ms. Inderveer Chana Lyall
Sr. Lecturer,CSED
TIET,Patiala

And

Mr. Amardeep Singh
Sr. Lecturer,CSED
TIET,Patiala

Submitted by
Madhu Sharma
(8033110)

Computer Science & Engineering Department
Thapar Institute of Engineering & Technology
(Deemed university)
PATIALA-147004

CERTIFICATE

It is hereby certified that the work which is being presented in the thesis entitled "**DNA Computing for Signal Matching"** in partial fulfillment of the requirement for the award of degree of Master of engineering in software engineering in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology (Deemed University), Patiala, is an authentic record of my own work out under the supervision and guidance Ms. Inderveer Chana Lyall and Mr. Amardeep Singh.

The matter presented in this thesis has not been Submitted by me for the award of any other degree of this or any other University.

Madhu Sharma

This is to certify that the above statement made by the candidate is correct and true to the best my knowledge.

Ms Inderveer Chana Lyall
Sr. Lecturer,
CSED
TIET
PATIALA

Mr Amardeep Singh
Sr. Lecturer,
CSED
TIET
PATIALA

Countersigned by

Mr. R.S. Salaria
Head
CSED
TIET
PATIALA

(Dr. D.S. Bawa)
Dean (Academic Affairs)
CSED
TIET
PATIALA

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to my supervisor Ms.Indeveer chana & Mr. Amardeep Singh for his valuable guidance, encouragement and valuable discussion for this thesis work. Words are inadequate to express the great care and interest taken by him in all aspects of my thesis work.

I express my gratitude to Mr. R.S. Salaria, Head, computer Science & Engineering Department and Mr. Rajesh Bhatia for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank all staff members and my colleagues who were always there at the need of hour and provided with all help and facilities, which I required, for the completion of my thesis work.

Last, but not last I am thankful to all those people who have directly or indirectly helped me during my thesis work.

Madhu Sharma
Roll No.-8033110

ABSTRACT

Electronic computers are only the latest in a long chain of human efforts to use the best technology available for performing computations. Ever since scientists discovered that conventional silicon-based computers have an upper limit in terms of speed, even electronic computers have their limitations: there is a limit to the amount of data they can store, and physical laws dictate the speed thresholds that will soon be reached. They have been searching for alternate media with which to solve computational problems. That search has led us, One of the most recent attempts to break down these barriers is to replace, once more, the tools for performing computations with biological ones instead of electrical one enter DNA computing

DNA computing was grounded in reality at the end of 1994, when Len Adleman of USC announced that he had solved a small instance of a computationally intractable problem using a small vial of DNA. By representing information as sequences of bases in DNA molecules, Adleman showed how to use existing DNA-manipulation techniques to implement a simple, massively parallel random search .DNA computing solve hard computational problem. After adleman experiment, DNA computing apply on so many hard computational problem.

Matching of digital signals is a fundamental problem that arises in many signal processing application. Signals matching problem is also considered as NP-complete problem. This work report apply new approach DNA computing on the matching of query signal into the stored signal. This work report proposes a method for signal matching. It include biological notation and proposes algorithm. I developed the software for simulating the proposed method. This report also discuss existing classical approach A Quick Search Method for Audio and Video Signals Based on Histogram Pruning. Calculate CPU time of proposed method.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Electronic computers are only the latest in a long chain of human efforts to use the best technology available for performing computations. While it is true that their appearance, some 50 years ago, has revolutionized computing, electronic computers mark neither the beginning nor the end of the history of computation.0

Ever since scientists discovered that conventional silicon-based computers have an upper limit in terms of speed, even electronic computers have their limitations: there is a limit to the amount of data they can store, and physical laws dictate the speed thresholds that will soon be reached. They have been searching for alternate media with which to solve computational problems. That search has led us, One of the most recent attempts to break down these barriers is to replace, once more, the tools for performing computations with biological ones instead of electrical one enter DNA computing

DNA computing was grounded in reality at the end of 1994, when Len Adleman of USC announced that he had solved a small instance of a computationally intractable problem using a small vial of DNA. By representing information as sequences of bases in DNA molecules, Adleman showed how to use existing DNA manipulation techniques to implement a simple, massively parallel random search .DNA computing solve hard computational problem. After adleman experiment, DNA computing apply on so many hard computational problem.

Using the four bases of DNA (adenine, thymine, cytosine, and guanine), Adleman encoded a classic “hard” problem known as the Travelling Salesman Problem into strands of DNA and utilized biological properties of DNA to find the answer. My thesis report comprises 5 chapter. Introduction signal matching and DNA computing Approach discussed in chapter number 1. Chapter 2 deal with the DNA fundamentals for providing sufficient knowledge in DNA computing operation and application.

Chapter 3 provide the knowledge of signal processing basics , definition of analog and digital signal. And in this chapter existing approach for signal matching.

Signal matching is basic problem of signal processing. Proposed method for signal processing, algorithm, result comprises in chapter 4. DNA computing simulator for proposed method included in chapter 5.

Chapter 2
DNA Fundamentals

2.1 Beginning of DNA computing

The practical possibility of using molecules of DNA as a medium for computation was first demonstrated by Adleman in 1994 [1]. Adleman's primary intention was to prove the feasibility of bio molecular computation but his work also gave an indication that the emergence of this new computational paradigm could provide an advantage over conventional electronic computing techniques. Specifically, DNA was shown to have massively parallel processing capabilities that might allow a DNA based computer to solve hard computational problems in a reasonable amount of time.

After Adleman successfully solved a directed Hamiltonian path problem using the tools of bimolecular engineering, others followed, applying similar algorithms to other hard computation problems, as well as devising more efficient computational schemes. A number of theoretical models for creating a universal DNA computer have been developed, and mathematical proofs have shown that some models of DNA computing are at least equivalent to a classical Turing machine.
A limited amount of work has been directed at real-life applications and the practical feasibility of DNA computers.

2.2 Concept of DNA computing and DNA computer

The usefulness of developing techniques of DNA computing and ultimately developing working DNA computers can be described as falling into one of the following two general categories[3]:

1. Applications making use of "classic" DNA computing schemes where the use of massive parallelism holds an advantage over traditional computing schemes, including potential polynomial time solutions to hard computational problems.
2. Applications making use of the "natural" capabilities of DNA, including those that make use of informational storage abilities and those that interact with existing and emerging biotechnology.

Classical models of DNA computers derive their potential advantage

- Perform millions of operations simultaneously.
- Generate a complete set of potential solutions.
- Conduct large parallel searches.
- Efficiently handle massive amounts of working memory.

2.3 Advantage of DNA computer over Conventional computer

To Adleman, the following advantages of DNA computing became evident

Speed: Conventional computers can perform approximately 100 MIPS (millions of instruction per second).Combining DNA strands as demonstrated by Adleman, made computations equivalent to 10^9 or better, arguably over 100 times faster than the fastest computer. The inherent parallelism of DNA computing was staggering.

Minimal Storage Requirements: DNA stores memory at a density of about 1 bit per cubic nanometer where conventional storage media requires 10^{12} cubic nanometers to store 1 bit.

Minimal Power Requirements: There is no power required for DNA computing while the computation is taking place. The chemical bonds that are the building blocks of DNA happen without any outside power source. There is no comparison to the power requirements of conventional computers.

2.4 DNA Basics

2.4.1 What is DNA

DNA is a polymer. The monomer units of DNA are nucleotides, and the polymer is known as a "polynucleotide." Each nucleotide consists of a 5-carbon sugar (deoxyribose), a nitrogen containing base attached to the sugar, and a phosphate group. There are four different types of nucleotides found in DNA, differing only in the

nitrogenous base. The four nucleotides are given one letter abbreviations as shorthand for the four bases.

- A is for adenine
- G is for guanine
- C is for cytosine
- T is for thymine

2.4.2 DNA Structure

This (DNA) structure has two helical chains each coiled round the same axis. Both chains follow right handed helices...the two chains run in opposite directions. The bases are on the inside of the helix and the phosphates on the outside.

The novel feature of the structure is the manner in which the two chains are held together by the purine and pyrimidine bases. The (bases) are joined together in pairs, a single base from one chain being hydrogen-bonded to a single base from the other chain, so that the two lie side by side. One of the pair must be a purine and the other a pyrimidine for bonding to occur .Only specific pairs of bases can bond together. These pairs are: adenine (purine) with thymine (pyrimidine), and guanine (purine) with cytosine (pyrimidine).

In other words, if an adenine forms one member of a pair, on either chain, then on these assumptions the other member must be thymine; similarly for guanine and cytosine. The sequence of bases on a single chain does not appear to be restricted in any way. However, if only specific pairs of bases can be formed, it follows that if the sequence of bases on one chain is given, then the sequence on the other chain is automatically determined.

The bases A and T, and C and G, can bind together, forming base pairs. Therefore every DNA sequence has a natural complement. For example if sequence S is ATTACGTCG, its complement, S', is TAATGCAGC. Both S and S' will come together (or hybridize) to form double stranded DNA. This complementarily makes DNA a unique data structure for computation and can be exploited in many ways.

2.5 DNA operation

DNA computation apply a specific sequence of biological operation to a set of strands. These operations are commonly used by molecular biologists.

2.5.1 Synthesis

A strand of DNA of specific length and sequence can be synthesized in laboratory. This is possible for strands up to a certain length. Longer ’random’ strands are available. They consist of DNA sequences that have been cloned from many different organisms. A solution, with four-nucleotide bases in it, is supplied to the synthesizer. These bases are combined according to a sequence entered by the user. The instrument makes millions of copies of the required oligonucleotides and places them in solution.

2.5.2 Denaturing, Annealing, Ligation

If we heat up a tube of DNA dissolved in water, the energy of the heat can pull the two strands of DNA apart This process is called denaturation when we have denatured the DNA, we have heated it to separate the strands.

Single stranded complementary DNA will spontaneously form a double strand of DNA when suspended in solution. But both the strands should run in opposite directions. This change occurs through the hydrogen bonds between the complementary base pairs. Cooling of single strand solution below 85o_95o C makes strands fuse again.

The process of splicing two pieces of DNA together. In practice, a pool of DNA fragments are treated with ligase ,and all possible splicing products are produced, including circularized forms and end-to-end ligation of 2, 3 or more pieces. Usually, only some of these products are useful, and the investigator must have some way of selecting the desirable ones.

2.5.3 Hybridization separation

An operation used in DNA computation is Separation by hybridization. It involves the extraction of any single strands containing a specific short sequence from a test tube (e.g., extract all strands containing the sequence *TAGACT*). If we want to extract single strands containing the sequence *X*, first many copies of its complement are created. These oligonucleotides are attached to a biotin molecule, which binds in turn to a fixed matrix. When the contents of the test tube are poured over this matrix, strands that contain the sequence *X* will anneal to the complementary strands. Washing the matrix removes all strands that do not anneal, leaving only strands containing *X*. These may then be removed from the matrix.

2.5.4 Replication

When we have denatured the two strands, there's something else we can do replicate the DNA. The key here is that any single-stranded piece of DNA can only hybridize with another if their sequences are complementary. If we have just one strand, we can actually build another strand to match it.

Here's how it's done, either in a test tube or in a live cell:

- The DNA strands are separated (for example, by heating them in a test tube).
- For each strand, we provide a primer, which is a short piece of DNA that sticks to one end of the strand.
- An enzyme is added. This is a specific type of protein called a "DNA polymerase" that can "read" the bases on one strand and can attach the complementary base to the growing strand.
- The polymerase "walks" down the template strand and creates its exact complement as it goes.
- The same thing happens to the other original strand.

2.5.5 Gel-Electrophoresis

The other method of separation is to separate the DNA strand by their size instead of their content. Gel electrophoresis is a key technique for sorting DNA strands by size.

Electrophoresis is the movement of charged molecules in an electric field. Since DNA molecules carry negative charge, when placed in an electrical field they tend to migrate towards the positive pole. The rate of migration of a molecule depends on its shape and electrical charge. If electrophoresis is carried out in a gel (usually made of agarose, polyacrylamide or a combination of the two) the migration rate of a molecule is also affected by its size. This is due to the fact that the gel is a dense network of pores through which the molecules must travel. Smaller molecules therefore migrate faster through the gel, thus sorting them according to size. The DNA was placed in a well cut out of the gel, and a charge applied. Once the gel has been run, it is necessary to visualize the results.

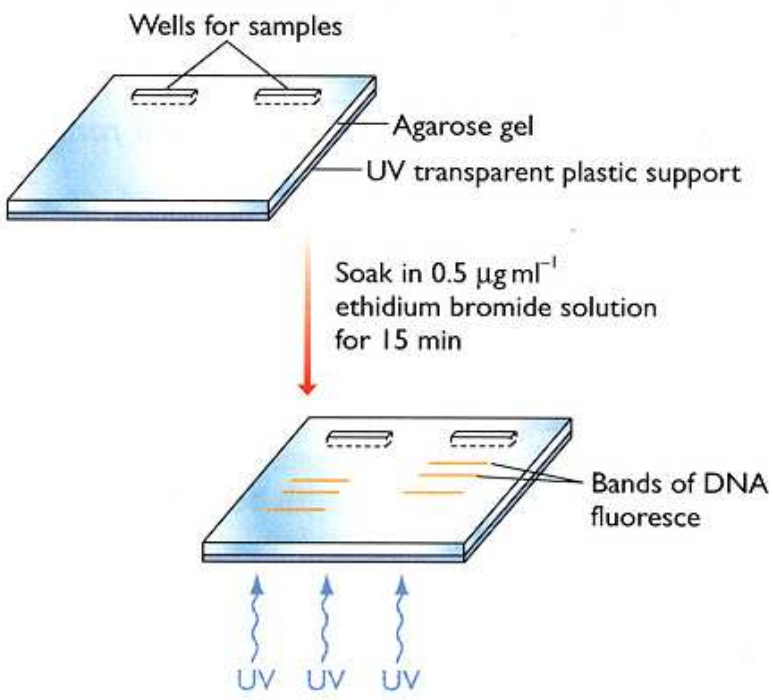

Fig 1:Gel-elctrophorises

2.5.6 Primer extension and Polymer chain Reaction

Polymerase Chain Reaction (PCR) is the basis for a number of extremely important methods in molecular biology. It can be used to detect and measure vanishingly small amounts of DNA and to create customized pieces of DNA.

PCR is a process based on a specialized polymerase enzyme, which can synthesize a complementary strand to a given DNA strand in a mixture containing the 4 DNA bases and 2 DNA fragments (primers, each about 20 bases long) flanking the target sequence. The mixture is heated to separate the strands of doublestranded DNA containing the target sequence and then cooled to allow

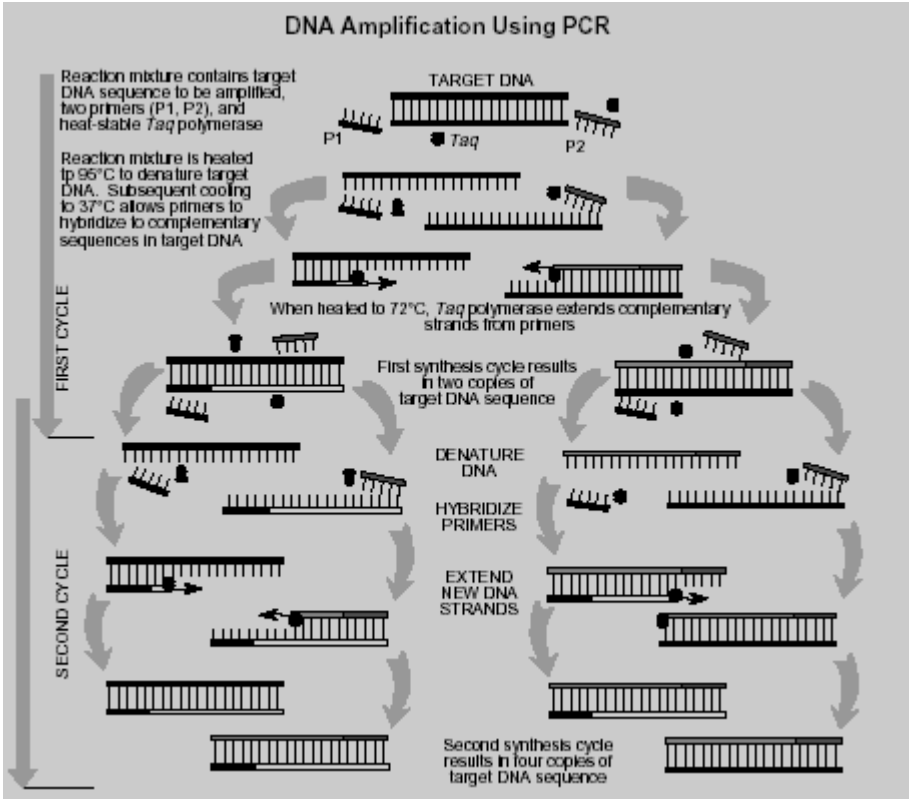

Fig 2 Polymer chain Reaction

(1) The primers to find and bind to their complementary sequences on the separated strands and

(2) The polymerase to extend the primers into new complementary strands. Repeated heating and cooling cycles multiply the target DNA exponentially,

since each new double strand separates to become two templates for further synthesis. In about 1 hour, 20 PCR cycles can amplify the target by a million fold.

2.5.7 Extraction

Given a test tube T_1 and a strand S, it is possible to extract all the strands in T_1 containing S as a subsequence and to separate them from those that don't contain it.

2.5.8 Union

Given two or more test tubes, say $T_1; T_2;: : : ; T_n$, it is possible to put in a new test tube the union of all the strands contained in $T_1; T_2; : : : ; T_n$.

2.5.9 Detection

Confirm presence/ absence of DNA in a given test tube.

2.6 DNA Application

2.6.1 Hamiltonian path problem (HPP)

Given N points, find a path visiting each and every point only once, and starting and ending at given locations.

NP problems are intractable with conventional computers, but can be solved using massively parallel computers. A DNA computer is a type of non-deterministic computer.

The Hamiltonian Path problem was chosen because it is known as NP-complete[3].

Directed graph with two nodes specified as the source and the destination, the Hamiltonian path is one that starts at the source node and ends at the destination node such that each node in the graph appears once and only once in the path. The Hamiltonian path problem is to determine whether there is a Hamiltonian path for a directed graph. Adleman has given solution to this problem using DNA computing in 1994 with seven nodes in the graph.

Consider directed graph of the Hamiltonian path problem. Directed graph map with four cities.

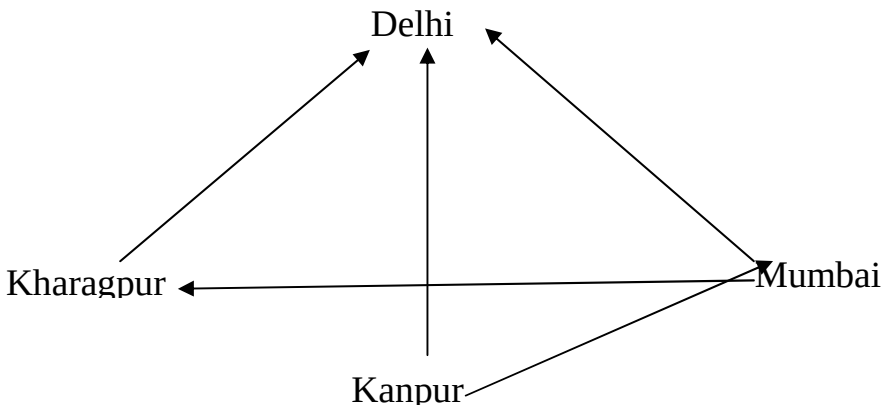

Fig 3: Root of City

Directly path between it is Mumbai to Delhi but not vice versa.

The problem is determine a path exists between at the start city (Kanpur), finish at the end city (Delhi) and traverse through each of the cities exactly once.

In DNA computation, Adleman thought of as each city is assigned a DNA sequence (ATCGTCGA for Kanpur) that can be thought of as a first name (ATCG) followed by a last name (TCGA) of the city.

City Name	***DNA Strand of City Name***	***Complemented DNA Strand***
Kanpur	ATCGTCGA	TAGCAGCT
Mumbai	GTACACTA	CATGTGAT
Delhi	TCAGACGA	AGTCTGCT
Kharagpur	CGATCGAT	GCTAGCTA

Table 1:Encoding of City

Route	***Route no.***
Kanpur-Mumbai	TCGAGTAC
Kanpur-Delhi	TCGATCAG
Mumbai-Delhi	ACTATCAG
Kharagpur-Delhi	CGATTCAG
Mumbai-Kharagpur	ACTACGAT
Mumbai-Kanpur	ACTAATCG

Table 2: Encoding Root

Each strand of DNA has its Watson-Crick compliment. Thus each city has its complimentary DNA name. Kanpur complimentary name becomes for instance TAGCAGCT. After working out these encoding, names and the DNA strand of root number synthesized. The following algorithm will give the Hamiltonian path for a given graph, if it exists in the graph. Given a directed graph with n vertices,

Algorithm:

Step1: Generate a set of random paths through the graph.

Step2: For each path in the set:

a. Check whether that path starts at the start vertex and ends with the end vertex. If not remove the path from the set.

b. Check whether that path passes through exactly *n* vertices. If not, remove that path from the set.

c. For each vertex, check whether that path passes through that vertex. If not, remove that path from the set.

Step3: If the set is nonempty, then report that there is a Hamiltonian path. If the set is empty, report that there is no Hamiltonian path.

Expiation of algorithm:

Step 1: Generate a large number of paths through the graph

The Kanpur to Mumbai root number (TCGAGTAC) and the complimentary name of Mumbai (CATGTGAT) might meet by chance. By design, the first sequence ends with GTAC and the second sequence starts with CATG. As these sequences are complimentary to each other, they will stick/anneal together. If the resulting sequence for Mumbai-to- Kharagpur root number (ACTACGAT), it too, will join with the resulting sequence because the first sequence (TGAT) is complimentary to the beginning of the second (ACTA). In this way the resulting sequence will grow in length, with DNA root numbers together by complimentary DNA city names. The addition of ligase in the mixture will then permanently concatenate the chains of DNA root numbers. Hence the test tube contains molecules that encode random paths through the different cities. Because he began with such a large number of DNA molecules and the problem contains just a handful of cities, there was a virtual certainty that at least

one of the molecules formed would encode the Hamiltonian path. It was amazing to think that the solution to a mathematical problem could be stored in a single strand. For the above graph, the following are some of the paths that are generated after step1.

Kanpura*!* Delhi

5*0*TAGCAGCTCATGTGATAGTCTGCT3*0*

3*0*TCGAGTACACTATCAGA5*0*

Kanpura*!* Mumbaia*!* Delhi

5*0*TAGCAGCTCATGTGATGCTAGCTAAGTCTGCT3*0*

3*0*TCGAGTACACTACGATCGATTCAG5*0*

Mumbaia*!* Kharagpur

5*0*CATGTGATGCTAGCTA3*0*

3*0*ACTACGAT5*0*

Kanpura*!* Mumbaia*!* Kharagpur

5*0*TAGCAGCTCATGTGATGCTAGCTA3*0*

3*0*TCGAGTACACTACGAT5*0*

Kharagpura*!* Delhi

5*0*GCTAGCTAAGTCTGCT3*0*

3*0*CGATTCAG5*0*

Kanpura*!* Mumbaia*!* Kanpura*!* Mumbaia*!* Delhi

5*0*TAGCAGCTCATGTGATTAGCAGCTCATGTGATAGTCTGCT3*0*

3TCGAGTACACTAATCGTCGAGTACACTATCAG5*0*

Kanpura*!* Mumbaia*!* Kharagpura*!* Delhi

5TAGCAGCTCATGTGATGCTAGCTAAGTCTGCT5*0*

3*0*TCGAGTACACTACGATCGATTCAG5*0*

Step2: Amplification of paths by PCR:

This important technique requires many copies of two short pieces of DNA as primers to signal the DNA polymerase to start its Watson-Crick replication. The primers used were the last name of the start city (TCGA) and the Watson-Crick compliment of the first name of the end city (AGTC). These two primers worked in correct: The first altered DNA polymerase to copy complements of sequences that had the right start city, and the second initiated the duplication of molecules that encoded the correct end city.

PCR proceeds through thermo cycling, repeatedly raising and lowering the temperature of the mixture in the test tube.

After performing PCR the result was that molecules with both the right start city and end cities were reproduced at an exponential rate. In contrast, molecules that encoded the right start city but an incorrect end city, or vise-versa, were duplicated in a much slower. DNA sequences that had neither the right start nor end were not duplicated at all. Thus by taking a small amount of the mixture after the PCR was completed, he obtained a solution containing many copies of the molecules that had both the right start and end cities, but few if any molecules that did not meet this criterion. After performing PCR, the following are the paths that begin at Kanpur end at Delhi.

Kanpura*!* Delhi

5*0*TAGCAGCTCATGTGATAGTCTGCT3*0*

3*0*TCGAGTACACTATCAGA5*0*

Kanpura*!* Mumbaia*!* Delhi

5*0*TAGCAGCTCATGTGATGCTAGCTAAGTCTGCT3*0*

3*0*TCGAGTACACTACGATCGATTCAG5*0*

Kanpura*!* Mumbaia*!* Kanpura*!* Mumbaia*!* Delhi

5*0*TAGCAGCTCATGTGATTAGCAGCTCATGTGATAGTCTGCT3*0*

3TCGAGTACACTAATCGTCGAGTACACTATCAG5*0*

Kanpura*!* Mumbaia*!* Kharagpura*!* Delhi

5TAGCAGCTCATGTGATGCTAGCTAAGTCTGCT5*0*

3*0*TCGAGTACACTACGATCGATTCAG5*0*

Step3: Gel-Electrophoresis

Next, he used gel electrophoresis to identify those molecules that had the right length All other molecules were discarded. For the above graph, after the completion of gel-electrophoresis the following path would be identified as length 24.

Kanpura*!* Mumbaia*!* Kharagpura*!* Delhi

TCGAGTACACTACGATCGATTCAG

Step4: Watson-Crick affinity-Separation:

To check the remaining sequences obtained in step3, for whether their paths passed through all the intermediary cities, he took advantage of Watson-Crick annealing in a procedure called affinity separation. This process uses multiple copies of a DNA probe molecules that encode the complimentary name of a particular city.

These probes are attached to the microscopic iron balls, each approximately one micron in diameter.

After performing affinity separation he suspended the balls in the tube containing the remaining molecules under conditions that encouraged Watson-Crick pairing. Only those molecules that contain the desired city's name (for example Mumbai) would anneal to the probes.

Then he placed a magnet against the wall of the test tube to attract and hold the metal balls to the side while he poured out the liquid phase containing molecules that did not have the desired city's name. Raising the temperature of the mixture caused the molecules to break free from the probes and re-dissolve in the liquid. Next, he reapplied the magnet to attach the balls again to the side of the test tube, but this time without any molecules attached. The liquid which now contained the desired DNA strands (in our example, encoding paths that went through Mumbai), could then be poured into a new tube for further screening. This process was repeated for the remaining intermediately cities (Kharagpur) in this case. This iterative procedure, which took an entire day to complete in the lab, was the most tedious part of the experiment. Probe molecules are used to locate DNA strands encoding paths that pass through the intermediate cities (Mumbai and Kharagpur). Probe molecules containing the complimentary DNA name of Mumbai (CATGTGAT) are attached to an iron ball suspended in liquid. Because of Watson-Crock affinity, the probes capture DNA strands that contains Mumbai's name (GTACACTA). Strands missing Mumbai's name are then discarded. The process is repeated with probe molecules encoding the complimentary DNA name of Kharagpur. When all the computational steps are completed, the DNA strands left will be those that encode the solution After completion step4, the following sequence would remain in the test tube as that pass through each of the cities once and only once which is thus a Hamiltonian path.

Kanpura*!* Mumbaia*!* Kharagpura*!* Delhi

TCGAGTACACTACGATCGATTCAG

At the conclusion of the affinity separation, he knew that the DNA molecules left in the tube should be precisely those encoding the Hamiltonian paths. Hence if the tube contained any DNA at all, he could conclude that a Hamiltonian path existed in the graph. No DNA would indicate that no such path existed. Fortunately to make this determination he could use an additional PCR step, followed by another gel-electrophoresis operation. To his delight, the final analysis revealed that the molecules that remained did indeed encode the desired Hamiltonian path. After seven days in the lab, the first DNA computation was complete.

2.7 Error in DNA Computation

Depending on the conditions under which the DNA reactions occur, two oligonucleotides can hybridize without exact matching between their base pairs. The mechanism of failures in matching bases depends on reaction conditions, the most important one being the temperature. Both Adleman and Lipton used random encoding under the basic assumption that random encoding method was adequate for their purposes.

Errors occurs in different stages of DNA operation

2.7.1 Extraction

Two Types of error occurs in extraction:

2.7.2 False Positive Errors

False Positive Errors occur when “bad” strands are extracted

2.7.3 False Negative Errors

False Negative Errors occur when “good” strands are not properly extracted

2.7.4 Encoding

A good encoding must satisfy all of these conditions at once in order to be useful for both hybridization reactions and data storage. These conditions are somewhat difficult to satisfy. And consequently it is difficult to compute valid code words.

2.7.5 Hybridization

Hybridization is the action of one oligonucleotide annealing to its complement. For example the 5'-> 3' sequence ATAGC will tend to anneal to its complement GCTAT.
DNA hybridization is not very reliable. Just as exact complements will tend to hybridize, close matches can hybridize as well.

2.7.6 Bubble match

We have a situation of bubble match when two strands of different length match for the external parts of the strands itself: the longest strand tend to link to the other only for the "tails" creating a bubble in the central part:

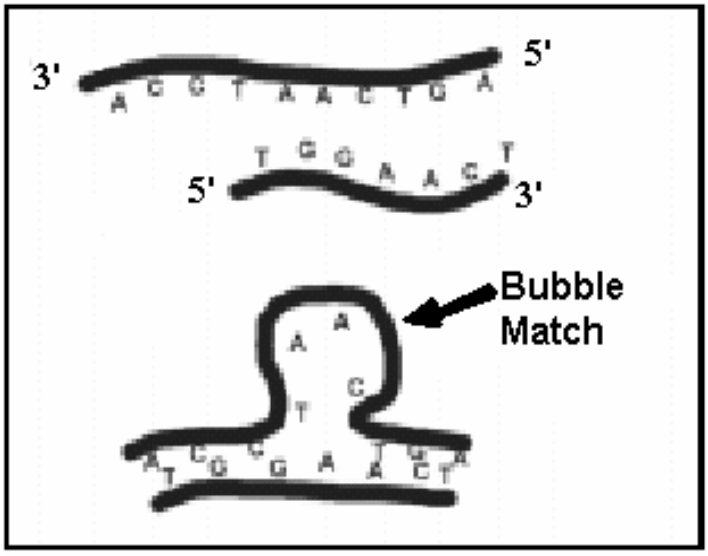

Fig 4 Bubble Match

Creating a simple database of subsequences it is possible to reduce spurious matching within strands' hybridization. Since a complete search may be quite complex and the

result not very significant, to improve tools' performance it is better to reduce the search space for fixed length of sub chains.

2.7.7 Slide Match

This is the normal situation of two single strand matching. Problems arise when strands match for a random number of bases. We could prevent this particular situation designing encoding scheme for the oligonucleotides.

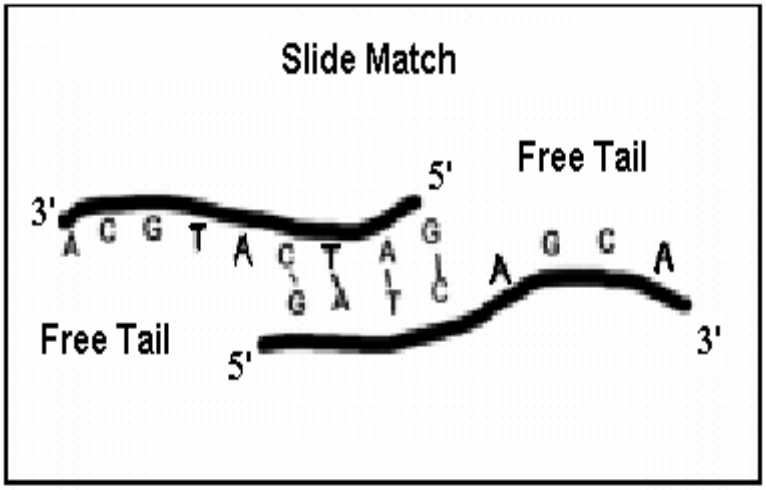

Fig. 5: Slide Match

2.7.8 Undesired Annealing

Most of the causes of errors in biological computations not only in the extraction operation are related to the possibility of undesired annealing: a strand *V* could anneal with one that is similar to *V*, but it is not the right one. In this case, we talk about partial matches. Another kind of undesired matching could happen between two "shifted" strands: for example, a strand *X* could partially anneal with a strand *Y*. Finally, a strand could anneal with itself, losing its linear structure.

Chapter 3
DSP Basic

Signal is a stream of energy that carries information [12]. Certain physical features of the energy stream will be a function of time (e.g., radio, TV, speech). Signal processing is to enhance, recover, or extract the information carried by signals.

Signals can be represented in either analog or digital form. Analog signals are continuous and can be illustrated as a curve which indicates the amplitude of a signal as a function of a continuous, independent variable such as time. Some analog signals repeat with a certain frequency, measured in hertz, and each repeated cycle has a duration, known as its period [12].

When we sample analog signals such as speech, they must first transform the analog signals into a series of corresponding numbers which accurately represent the shape of the analog waveform. The sampling process quantizes the continuous analog signal values into a limited set of discrete binary numbers, each of which comprises a string of binary digits, or bits. Once an analog signal is converted into the digital domain, computers can analyze the frequency spectra and other characteristics of a signal by using a wide variety of digital signal processing algorithms [12].

Digital signals are often obtained as an approximation of analog signals. It is the signal that can be processed by computers. In fact, the use of digital signals is largely promoted or demanded by the revolution of computer technology.

In digital signal processing, the physical meaning of the independent variable $n = \ldots -2,-1,0,1,2\ldots$ is typically ignored. It is treated as a discrete sequence of numbers.

3.1 Digitization of Signal

Converting an analog signal into its digital form is usually the very first step in signal processing. The primary question is how fast the analog signal need to be sampled so that the information in the signal is well preserved.

Most signals can be considered as band limited, i.e., the energy for frequency components higher than a particular frequency is almost zero. This frequency is called a cutoff frequency. To preserve the information of a signal with cutoff frequency f_c. The sampling frequency should be $f_s = 2 f_c . f_s$ is called the Nyquist frequency.

The sampling frequency is so specified because the spectrum of the digital signal is a periodic repetition of the original spectrum with the period equal to the sampling frequency.

Sampling at a frequency lower than the Nyquist frequency will distort the signal and, thus, loose certain information. Information lost due to inappropriate sampling is referred to as aliasing. Sampling at a frequency much higher than the Nyquist frequency, however, will be a waste of resources such as disk space.
In common practice, the sampling frequency is chosen a little bit higher than the Nyquist frequency. A lowpass filter is used to guarantee that aliasing will not occur.

3.2 Quantization of Signal

Quantization is a process of using a set of discrete numbers to approximate the amplitudes of a signal. Apparently, the more the available levels the better the approximation.

The number of available levels is specified using bit as a unit. For example, an 16 bit quantizer has 2^{16} = 65536 levels.

The final number range in the computer depends on the design of the A/D converter. For example, it could be from 0 to 65536 or from -32767 to 32768 for a 16-bit A/D converter. A baseline shift may be necessary.

The inaccuracy due to quantization is referred to as quantization noise. A rule of thumb is that the signal to noise ratio will increase 6 dB for every bit increase of quantization level.

3.3 Time and frequency of a digital signals

It is often necessary to know the physical meaning of a discrete sequence (e.g., the time location of sample 5; the original frequency of a digital sinusoid signal). These information can be computed given the sampling frequency.

The time for sample n is $t_n = n \times Ts$ where $T_s = 1/f_s$ is the time interval between any two samples.

3.4 Signal Processing Applications

1) Speech signals can be synthesized from discrete-time models of the human vocal tract using digital signal processing. The speech synthesizer is a time-varying digital filter which models the response of the articulators (mouth, throat, etc.) to excitation from the vocal cords and air turbulence. This results in quality voice synthesis for computers, and data reduction for the digital voice communications used in talking consumer products. Analysis of speech production can also be performed by examining the speech signal's spectrum and its related properties. These techniques can be used to determine the spectral properties of speech (such as the voiced sounds In vowels or the unvoiced sounds in fricatives) and the distance between pitch pulses in voiced sounds. Sounds can also be analyzed by examining their spectra. Once obtained, this information can be useful in the development of an automatic speech recognition or compression system.

2) Signal processing is central to many defense-related systems, and plays a major role in automatic control, navigation, and target tracking. Radar and sonar systems rely heavily on the application of signal processing technology.

3) In neurophysiology, signal processing techniques are used to mathematically characterize signals obtained by electroencephalograph from the scalp of a subject. Frequency spectrum analysis is used to reveal the possible presence of energy prominence at certain frequencies, which are important to physicians for diagnostic purposes.

4) Two-dimensional or image processing is used to analyze visual data such as x-rays, aerial photographs, and satellite transmissions from space. With the advent of high-definition television, new image processing algorithms are being developed to enhance picture quality, while new techniques are being investigated to improve the quality of existing transmissions. The potential of high-definition television coupled with image processing technology has opened the way for the all-digital television sets of the future.

Audio and video data from radio, television, databases, or on the Internet has been a source of recent research interest. Among the many studies that have targeted audio or video information search, most have dealt with so-called content-based retrieval by means of indexing and classifying audio or video information. For example, in image or video retrieval tasks. Similarly, in audio retrieval tasks, most works have been based on high level information such as audio content classification, recognized speeches, or transcribed musical pieces.

3.5 Existence Approach for Signal Matching

Matching of digital signals is a fundamental problem that arises in many signal processing applications. It can be used as a search or classification mechanism by quantifying the similarity between signals. search mechanism is essential for database

retrieval, and signal classification is a key system component in data mining applications.

3.5.1 A Quick Search Method for Audio and Video Signals Based on Histogram Pruning:

Quick method of similarity based signal searching to detect and locate a specific audio or video signal given as a query in a stored long audio or video signal.
This method is referred to as time-series active search, offers significantly faster search with sufficient accuracy.

Signal matching problem solve using different-different techniques such as content based retrieval, similarity based searching. In Both method methods

Have some disadvantage such as less accuracy and speed of matching.

New technique A Quick Search [7] Method for Audio and Video Signals Based on Histogram Pruning algorithm is time series active search algorithm.

Step 1: Firstly, the feature vectors are calculated from both the query signal and stored signal. The windows are then applied to both the query-signal and stored-signal feature vectors.

Step 2: The feature vectors over the windows are classified into a certain number of types, and the number of occurrences of each feature type is counted to create the histogram. The window length is the same as the query signal duration.

Step 3 : Similarity between the query-signal histogram and stored-signal histogram is calculated. When the similarity exceeds a threshold value chosen in advance, the query signal is considered to be detected and located in the stored signal.

Step 4; The window on the stored signal is shifted forward in time and the search proceeds. We call this algorithm “time-series active search.”.

Chapter 4
A Proposed Method for Signal Processing

Matching of digital signals is a fundamental problem that arises in many signal processing applications[4]. It can be used as a search or classification mechanism by quantifying the similarity between signals. search mechanism is essential for database retrieval, and signal classification is a key system component in data mining applications. Furthermore, a number of important problems in signal and image processing, like motion and disparity estimation, are matching type problems. One of the key characteristics of the digital signal matching problem is the fact that the matches are typically imprecise, due to the nature of the problem or the presence of noise in the data, which is almost always inevitable. Thus, to solve the matching problem it is necessary to quantify the similarity between signals.

Another key characteristic of the digital signal matching problem is the enormous amount of computation that is typically required to find the optimal solution. It is thus of critical importance to find efficient implementations that can provide optimal or near optimal solutions. An additional requirement is the scalability of the solution.

Signal matching efficiency[7] can be measure two broad parameters.

1. Search speed
2. Search accuracy

4.1 DNA Computing Approach

Digital signal matching problem[4] the matches are typically imprecise, due to the nature of the problem or the presence of noise in the data, enormous amount of Computation, scalability the solution.

However using DNA techniques to solve digital signal matching problems. This thesis report presents a DNA based model for this problem, which makes use of a DNA computing based algorithm to solve the signals matching problem. The proposed model includes the various biological notations developed to solve the problem, the problem formulation framework and the respective algorithm developed to solve the problem.

Proposed method based on the Adleman model for solving travelling salesman problem of graph. And use the ligation and annealing power of DNA computing. Non-specific hybridization is generally not desirable DNA computing. But non-specific hybridization can be very useful (and is actually necessary) in signal processing applications, where an exact match is typically not possible

DNA annealing particularly suited for the matching problem is the fact that it depends only on the concentration of elements and is independent of the total number of elements. This is a very important difference with traditional digital databases, in which the search time depends on the size of the database. It is this property of DNA annealing, combined with its high compactness, which makes DNA an excellent information storage medium.

The main idea of this method, each stored signal represented as a vertex of graph and query signal represented as a complementary edge of the graph.

Each vertex represented unique DNA strand in tube and edge also represent unique DNA strand in another tube. Then both tubes have to merge for the annealing operation. After annealing we get the output of the result in the tube.

4.1.1 Problem Formulation

The Signal matching problem can we expressed taking using two signals. Assuming the presence of two signals S1 and S2 of length N and K<N respectively, we want to find best matches (locations) of the signal S2 in S1.

Number of important problems in signal image processing like motion, presence of noise signal processing and disparity estimation.

4.2 Proposed Method for Signal matching

The signal matching problem can be expressed more naturally in graph theoretic terms. We have stored analog signal which is converted into digital signal. Each signal of one time period represented as a vertex of graph. Let $V = v1, v2 \ldots v_n$ be a set of vertices. Query analog signal which is also converted into digital signal is represented as edge of vertex. Query signal represented as may be only one edge or many edges depends upon the size of query signal. So E={{e_1 }, {$e_{1,}e_2$ } {$e_{1,}e_2 \ldots e_m$}} be a set of edges.

Assign DNA single strand to each vertex and complementary strand for each edge. Each vertex is divide into two equal parts, first part of DNA strand represents the sample number and second part of strand is represent the strength of signal.

Each edge also divide into two equal parts first part of DNA strand represents the sample number and second part of strand is represent the strength of signal. Coding of DNA strand for edge is complemented DNA strand.

4.2.1 Algorithm for proposed method

Step 1: Stored signal $\{x(t) \forall 0 \le t \le T\}$ is converted into a database of signal elements $\{x_i = x(t = (i-1)t_s \; to \; it_s); 1 \le i \le N, Nt_s = T\}$.

Step 2: Each signal element x_i is comprising of (n) number of samples $\{X_j \forall 1 \le j \le n\}$. One particular sample represents the signal strength at an instant.

Step 3: Each sample in a signal element is represented as a vertex of a graph. Assign a DNA strand for each vertex. The DNA strand is divided into two parts

1} First part of DNA S_{1j} strand represents the sample number.

2} Second part of DNA S_{2ij} strand represents strength of the corresponding sample.

3) The DNA strand $S_{ij} = S_{1j} + S_{2ij}$ is put into the tube T1.

4) Thus $\left\{ x_i \rightarrow S_i ; S_i = \sum_{j=1}^{n} S_{ij} \right\}$

Step 4: Repeat step 3 for each signal element.

Step 5: Query signal $\{x_q(t) \forall\, 0 \le t \le t_s\}$ comprising of (n) number of samples $\{X_j \forall 1 \le j \le n\}$ is given.

Step 6: Similar to step 3, the query signal is mapped into strands using complementary strands for sample number and the signal strength. Thus, $\left\{ x_q \rightarrow \overline{S}_q ; \overline{S}_q = \sum_{j=1}^{n} \overline{S}_{qj} \right\}$ and, $\{\overline{S}_{qj} = \overline{S}_{2qj} + \overline{S}_{1(j+1)} \forall 1 \le j \le (n-2)\} \& \{\overline{S}_{q(n-1)} = \overline{S}_{2q(n-1)} + \overline{S}_{1n} + \overline{S}_{qn}\}$. The strands $\overline{S}_{qj}$'s are put into the tube T2.

Step 7: Combine the tube T1 and T2 into T3.

T3←Merge (T1, T2)

Step 8: Apply annealing operation on the tube T3

Anneal (T3).

Step 9: Melt operation for separating single DNA strand.

Melt (T3).

Step 10: Apply length based operation for finding longest DNA strand from tube T3.

Step 11: Exit

4.2.2 Explanation

The proposed method solve the problem in three steps.

4.2.2.1 Prepare a digital database

Stored signal $\{x(t) \forall 0 \le t \le T\}$ is converted into a database of signal elements $\{x_i = x(t = (i-1)t_s \; to \; it_s); 1 \le i \le N, Nt_s = T\}$. The sampling process quantizes the continuous analog signal values into a limited set of discrete binary numbers.

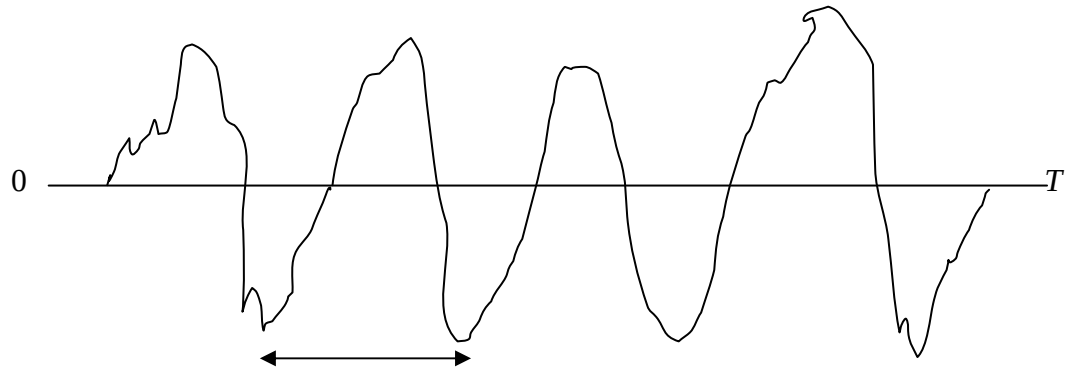

Fig. 6 Stored Signal

Each signal element x_i is comprising of (n) number of samples $\{X_j \forall 1 \le j \le n\}$. One particular sample represents the signal strength at an instant.

Each sample represents vertex of graph. Assign the DNA strand for each vertex. DNA strand divide into two parts, first part of DNA strand S_{1j} represents the sample number and Second part of DNA S_{2ij} strand represents strength of the corresponding sample. All strands put into the tube T1.

After making digital database input Query signal $\{x_q(t) \forall 0 \le t \le t_s\}$ comprising of (n) number of samples $\{X_j \forall 1 \le j \le n\}$ is given.

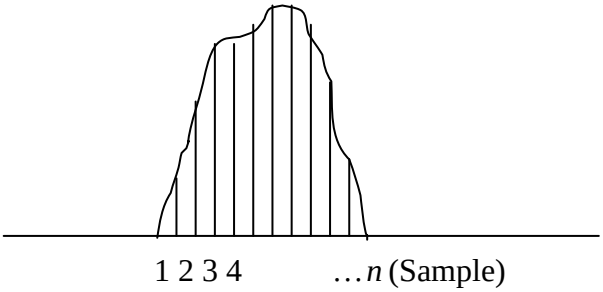

Fig 7. Signal Element

Each sample of query signal assign as a edge of graph. Encoding for edge we are taking complemented DNA strand of second part of first sample and complemented DNA strand of first part of second sample. This process done up to $(n-1)$ signal. Second half complimented DNA strand of n_{th} sample is added in last. All query signal DNA strand put into tube T2.

4.2.2.2 Processing

In this step after preparing tube T1 and T2, combine these tube into tube T3.
After merge operation all DNA strand in tube T3. Then apply anneal operation for matching of signal. After anneal operation we can found signal is matched or not. Then melt operation apply, single DNA strand obtained. And apply the length based operation for separating largest DNA strand from tube T3.

1) Merge operation

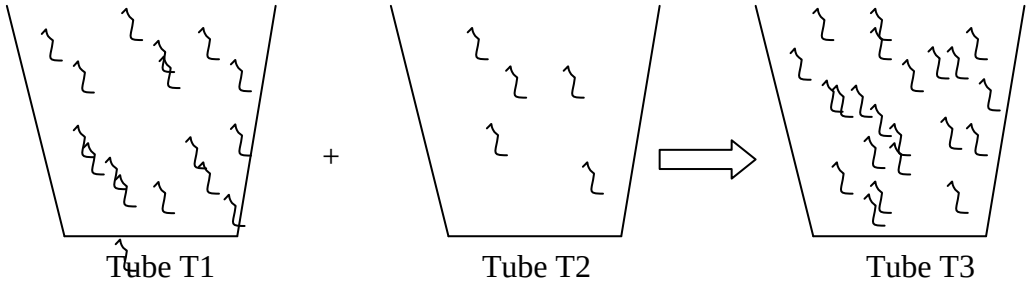

Fig 8 Merge operation

2) Anneal operation

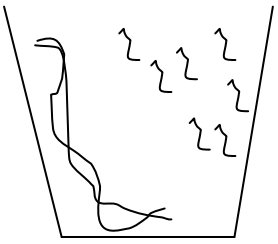

Fig 9: Anneal Operation

3) Melt operation

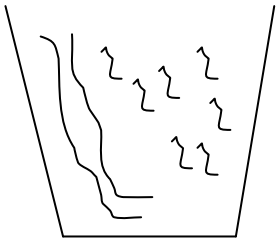

Fig 10: Melt Operation

4) Gel-Electrophoresis operation

Extract the DNA strand of largest number of DNA bases.

4.3 Result

Result obtained of proposed method, from simulated software. Calculate the CPU time from the DNA computing software with fix data base size and CPU time with fix number of sample.

Fixed database size: 100

Number Of Samples	***CPU time (in sec.)***
10	13

9	9
8	6
7	4
6	2

Table 4: CPU Time with fix DB

Fixed number of sample: 8

Database Size	***CPU time (in sec.)***
10	0
15	0
20	0
25	1
30	3
35	4

Table 5: CPU Time of Fixed Number of Sample

Chapter 5
The DNA Computing Software

5.1 The DNA Computing Software

The use of true DNA computers may not be possible, as there don't exist commercially at the moment. It seems perfectly reasonable to try to build software to simulate one. That was the motivating idea for this program. The purpose of this program is only to help designing algorithms for DNA computers. The program uses an ideal model for the DNA molecules. The model doesn't take into account the possibilities of imperfect operations, error correction etc. Those are research questions and need to be taken care of separately. Also the biological, chemical and physical factors affecting the calculations are not covered by this program as they need much more specialized knowledge and expertise in the areas of molecular biology and biochemistry.

This program is a C++ class library that provides the tools to perform DNA calculations using conventional electronic computers. Library functions include interface functions to perform the computations and internal functions to implement the operations. The DNA programs are written in C++ using these classes and used in the linking phase of the compilation to produce the executable version of the DNA program.

5.1.1 Advancement

This section contains various thoughts that have arisen during the development of this program.

5.1.1.1 History

DNA computing began with an experiment published by Adleman in Science in 1994. In the experiment, he solved a simplified instance of a famous NP-complete computational problem called the Travelling Salesman Problem, a Directed Hamiltonian path problem with 7 nodes, which is trivial even for manual computation.

However, his work motivated many researches because he employed DNA as the computation tool. He used DNA strands to encode the problem and biological operations to simulate the computation.

5.1.1.2 Purpose of the program

From the first moment it seemed that there was much theoretical work done on DNA computing, but almost none on the practical side, i e. biochemistry. Adleman's experiment was almost the only actual experiment done. Thus it was only possible to use the ideal, theoretical model in the program.

The second decision was that the simulator should have functions to simulate only such operations that were possible in practice. Both the time complexity and space complexity are still of a remarkable degree but tolerable to simulate small instances of the problems involved in DNA computing.

Despite these restrictions, the program was designed to be as modular as possible to allow implementation of new models easily. Thus C++ was chosen as the language for developing simulator.

5.2.2 Simulator Components

5.2.2.1 Tube

Design Concept

Tube was designed having an actual test tube in mind. Operations class tube permits include operations performed also for ordinary test tubes, eg. Pouring the contents of a tube into another one. In our restricted model it is only needed to know if there is one or more given strands in the tube, so every strand appears only once in the tube. Of course this model has its limitations but it is sufficient for our needs.

Input/ Output functions

tube()

The Constructor of the class: Construct an empty tube with no strands present in it.

Return Value: None

Read()

Prints out all the strands in the tube

5.2.2.2 Strand

Design Concept

The ‘Strand’ has been designed with actual strands of DNA in mind. Its data structure facilitates the initialization, combination and separation of strands and inspecting them in various ways. Most of the functions are often needed by the end-user. The complete strand is consisting of following parts in the program

Input/ Output functions

Strand()

Constructor of the class. Constructs an empty strand.

Read()

Read the elements in the strand using element indices.

5.2.2.3 Anneal operation

The process in which the two strands connect to each other is called annealing the strands. Thus the strand can be any combination of single and double strand elements.

Anneal(const int len)

Combines the matching strands in the tube with each other

Arguments: optional argument len gives the maximum length of resulting strands

2.3.1 Output of Program

enter no of samples... 3

enter db_sz... 5

enter random smaple (<=db_sz)3

after addition of V...

Strand: 1

1 7

Strand: 2

3 9

Strand: 3

1 13

Strand: 4

1 19

Strand: 5

1 25

Strand: 6

1 31

Strand: 7

5 11

Strand: 8

3 15

Strand: 9

3 21

Strand: 10

3 27

Strand: 11

3 33

Strand: 12

5 17

Strand: 13

5 23

Strand: 14

5 29

Strand: 15

5 35

after addition of E...

Strand: 1

1 7

Strand: 2

-19 -3

Strand: 3

-21 -5 -23

Strand: 4

3 9

Strand: 5

1 13

Strand: 6

1 19

Strand: 7

1 25

Strand: 8

1 31

Strand: 9

5 11

Strand: 10

3 15

Strand: 11

3 21

Strand: 12

3 27

Strand: 13

3 33

Strand: 14

5 17

Strand: 15

5 23

Strand: 16

5 29

Strand: 17

5 35

Anneal

after annealing...

Strand: 1

1 7

Strand: 2

-19 -3

Strand: 3

-21 -5 -23

Strand: 4

-21 6 24

Strand: 5

-19 4 15

Strand: 6

-19 4 9

Strand: 7

-19 4 33

Strand: 8

-19 4 27

Strand: 9

-19 4 21

Strand: 10

-19 4 22 -5 -23

Strand: 11

-19 4 22 6 24

Strand: 12

3 9

Strand: 13

1 13

Strand: 14

1 19

Strand: 15

1 25

Strand: 16

1 20 -3

Strand: 17

1 20 4 9

Strand: 18

1 20 4 15

Strand: 19

1 20 4 21

Strand: 20

1 20 4 22 -5 -23

Strand: 21

1 20 4 22 6 24

Strand: 22

1 20 4 27

Strand: 23

1 20 4 33

Strand: 24

1 31

Strand: 25

5 11

Strand: 26

3 15

Strand: 27

3 21

Strand: 28

3 27

Strand: 29

3 22 -5 -23

Strand: 30

3 22 6 24

Strand: 31

3 33

Strand: 32

5 17

Strand: 33

5 23

Strand: 34

5 29

Strand: 35

5 35

Melt

after melting...

Strand: 1

1 25

Strand: 2

1 19 3 9

Strand: 3

1 7

Strand: 4

-19 -3

Strand: 5

-21 -5 -23

Strand: 6

-19 -3 -21 -5 -23

Strand: 7

1 19

Strand: 8

1 13

Strand: 9

1 19 3 21 5 23

Strand: 10

1 19 3 21

Strand: 11

1 19 3 15

Strand: 12

1 19 3 33

Strand: 13

1 19 3 27

Strand: 14

3 33

Strand: 15

3 21

Strand: 16

3 9

Strand: 17

1 31

Strand: 18

3 15

Strand: 19

3 27

Strand: 20

3 21 5 23

Strand: 21

5 23

Strand: 22

5 17

Strand: 23

5 11

Strand: 24

5 35

Strand: 25

5 29

Length

after Abslen.

Strand: 1

1 19 3 21 5 23

after Abslen (indexed res)

Strand: 1

7 8 9

Jai shree Krishna !!!!!!!!!

Conclusion and Future scope

DNA computing can be accurately described as a collection of new computing approaches rather than a single technique. Each of these different approaches within biomolecular computing can be associated with different potential applications that may prove to place them at an advantage over conventional methods. The advances made in DNA computing will be beneficial to several areas of scientific computing. Advancements in DNA computing may also serve to enhance understanding of both the natural and computer sciences. For these reasons, and due to dependence on various fields of computer science, mathematics, natural science, and engineering, continued interdisciplinary collaboration is very important to any future progress in DNA computing.

The present study focussed on the application of existing DNA computing methodologies to the problem of signal processing. It is demonstrated that the modification of coding schemes and judicious selection of various sub-processes of this new computing paradigm describes for the first time identification of a signal by an artificial non-electronic scheme. This proves the feasibility of DNA-based algorithms for implementation on electronic computers as well as the applicability of the biochemical process for solution of signal processing problems.

References

[1] Adleman, L.,1994. Molecular computation of solutions to combinatorial problems. Science, vol 266.

[2] Brenneman and A. Condon, "Strand design for bimolecular computation," (Survey Paper), University of British Columbia, Vancouver, Canada, to appear 2001.

[3] DNA COMPUTING-GRAPH ALGORITHMS G. P. Raja Sekhar Department of Mathematics, Indian Institute of Technology.

[4] S.A. Tsaftaris, A.K. Katsaggelos, T.N. Pappas and E.T. Papoutsakis, "DNA computing from a signal processing viewpoint", *IEEE Signal Processing Magazine,* vol. 21, no. 5, pp. 100-106, September 2004.

[5] S.A. Tsaftaris, A.K. Katsaggelos, T.N. Pappas and E.T. Papoutsakis, "How can DNA-Computing be applied in Digital Signal Processing?", *IEEE Signal Processing Magazine,* vol.21, no. 6, pp. 57-61, November 2004

[6] S.A. Tsaftaris, A.K. Katsaggelos, T.N. Pappas and E.T. Papoutsakis, "DNA Based Matching of Digital Signals", in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 581-584, Montreal, Quebec, Canada, 17-21 May 2004.

[7] A Quick Search Method for Audio and Video Signals Based on Histogram Pruning Kunio Kashino, *Member, IEEE,* Takayuki Kurozumi, and Hiroshi Murase, *Senior Member, IEEE,* 348 IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 5, NO. 3, SEPTEMBER 2003

[8] A. Narayanan and S. Zorbalas, DNA algorithms for computing shortest paths, *Genetic Programming 1998,* Koza, J. R. *et al.* (eds.), Morgan Kaufmann, pp. 718-723, 1998.

[9] Adleman, L.,1996. On Constructing a Molecular Computer. 1st DIMACS workshop on DNA based computers. Princeton. In DIMACS series. vol.27(1996) . pp 1-21.

[10] DNA Computing Web Pages (http://www.liacs.nl/home/pier/webPagesDNA/)

[11] DNA Computing: A Primer (http://www.arstechnica.com/reviews/2q00/dna/dna-1.html

[12] Digital Signal Processing, Principles Algorithms, and Application John G.Proakis.