A

Seminar report

On

DNA Computing

Submitted in partial fulfillment of the requirement for the award of degree
of Bachelor of Technology in Computer Science

SUBMITTED TO:

www.studymafia.org

SUBMITTED BY

www.studymafia.org

Acknowledgement

I would like to thank respected Mr…….. and Mr. ……..for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

Preface

I have made this report file on the topic **DNA Computing**; I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude to ……..who assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

CONTENTS

AN INTRODUCTION TO DNA COMPUTING

Introduction:

DNA (Deoxyribose Nucleic Acid) computing, also known as molecular computing is a new approach to massively parallel computation based on groundbreaking work by Adleman. DNA computing was proposed as a means of solving a class of intractable computational problems in which the computing time can grow exponentially with problem size (the 'NP-complete' or non-deterministic polynomial time complete problems).

A DNA computer is basically a collection of specially selected DNA strands whose combinations will result in the solution to some problem, depending on the problem at hand. Technology is currently available both to select the initial strands and to filter the final solution. DNA computing is a new computational paradigm that employs (bio)molecular manipulation to solve computational problems, at the same time exploring natural processes as computational models. In 1994, Leonard Adleman at the Laboratory of Molecular Science, Department of Computer Science, University of Southern California surprised the scientific community by using the tools of molecular biology to solve a different computational problem.

The main idea was the encoding of data in DNA strands and the use of tools from molecular biology to execute computational operations. Besides the novelty of this approach, molecular computing has the potential to outperform electronic computers.

For example, DNA computations may use a billion times less energy than an electronic computer while storing data in a trillion times less space. Moreover, computing with DNA is highly parallel: In principle there could be billions upon trillions of DNA molecules undergoing chemical reactions, that is, performing computations, simultaneously.

History & Motivation:

"Computers in the future may weigh no more than 1.5 tons." So said Popular Mechanics in 1949. Most of us today, in the age of smart cards and wearable PCs would find that statement laughable.

We have made huge advances in miniaturization since the days of room-sized computers, yet the underlying computational framework has remained the same.

Today's supercomputers still employ the kind of sequential logic used by the mechanical dinosaurs of the 1930s. Some researchers are now looking beyond these boundaries and are investigating entirely new media and computational models.

These include quantum, optical and DNA-based computers. It is the last of these developments that this paper concentrates on.
The current Silicon technology has following limitations:

- Circuit integration dimensions
- Clock frequency
- Power consumption
- Heat dissipation.

The problem's complexity that can be afforded by modern processors grows up, but great challenges require computational capabilities that neither most powerful and distributed systems could reach.

The idea that living cells and molecular complexes can be viewed as potential machinic components dates back to the late 1950s, when Richard Feynman delivered his famous paper describing "sub-microscopic" computers.

More recently, several people have advocated the realization of massively parallel computation using the techniques and chemistry of molecular biology. DNA computing was grounded in reality at the end of 1994, when Leonard Adleman, 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. He solved the traveling salesman problem also known as the “Hamiltonian path" problem.

There are two reasons for using molecular biology to solve computational problems.

(i) The information density of DNA is much greater than that of silicon : 1 bit can be stored in approximately one cubic nanometer. Others storage media, such as videotapes, can store 1 bit in 1,000,000,000,000 cubic nanometer.

(ii) Operations on DNA are massively parallel: a test tube of DNA can contain trillions of strands. Each operation on a test tube of DNA is carried out on all strands in the tube in parallel.

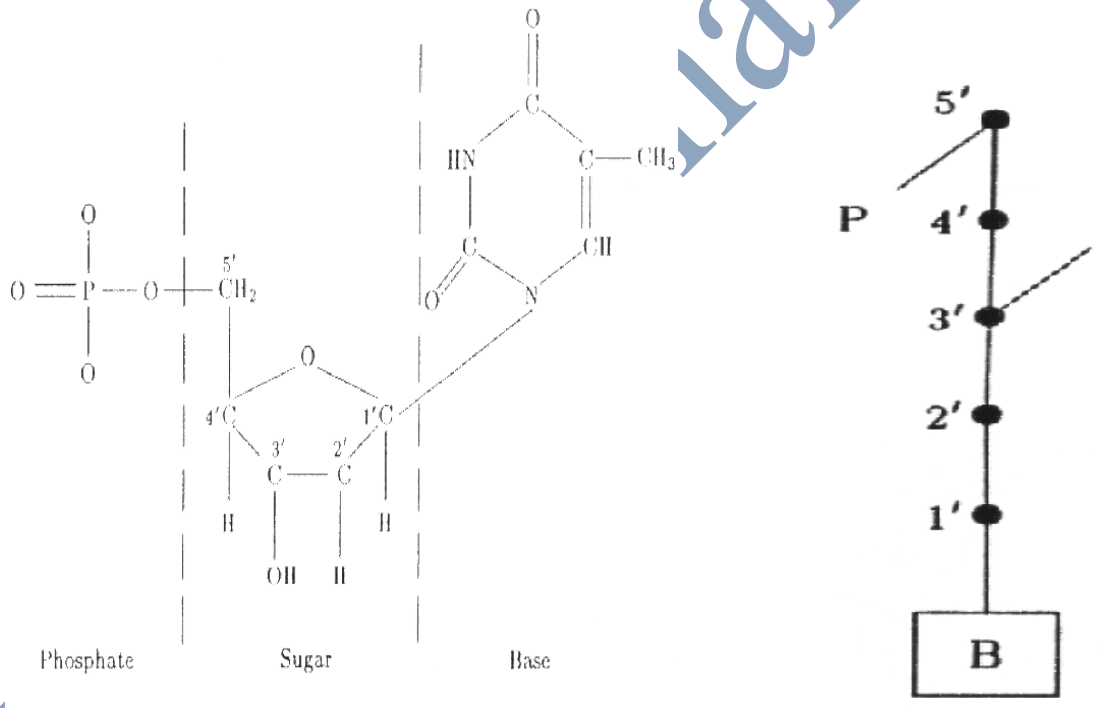
DNA Fundamentals

DNA (deoxyribonucleic acid) is a double stranded sequence of four nucleotides; the four nucleotides that compose a strand of DNA are as follows: adenine (A), guanine (G), cytosine (C), and thymine (T); they are often called bases. DNA supports two key functions for life:

- coding for the production of proteins,
- self-replication.

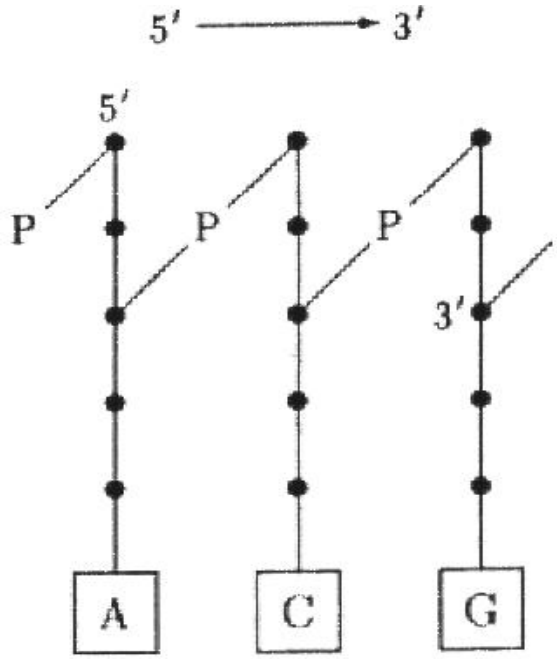
Each deoxyribonucleotide consists of three components:

- a sugar — deoxyribose
  - five carbon atoms: 1´ to 5´
  - hydroxyl group (OH) attached to 3´ carbon
- a phosphate group
- a nitrogenous base.

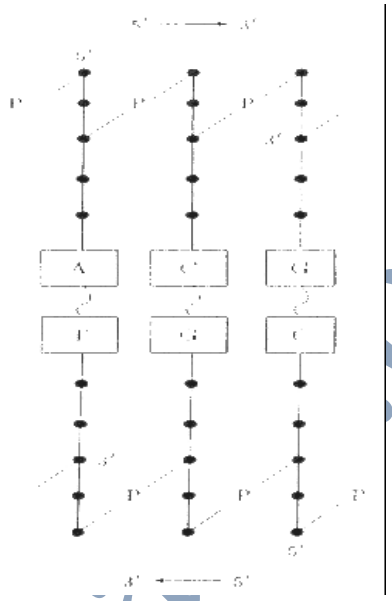


The chemical structure of DNA consists of a particular bond of two linear sequences of bases. This bond follows a property of Complementarity: adenine bonds with thymine (A-T) and vice versa (T-A), cytosine bonds with guanine (C-G) and vice versa (G-C). This is known as Watson-Crick complementarity.

The DNA monomers can link in two ways:

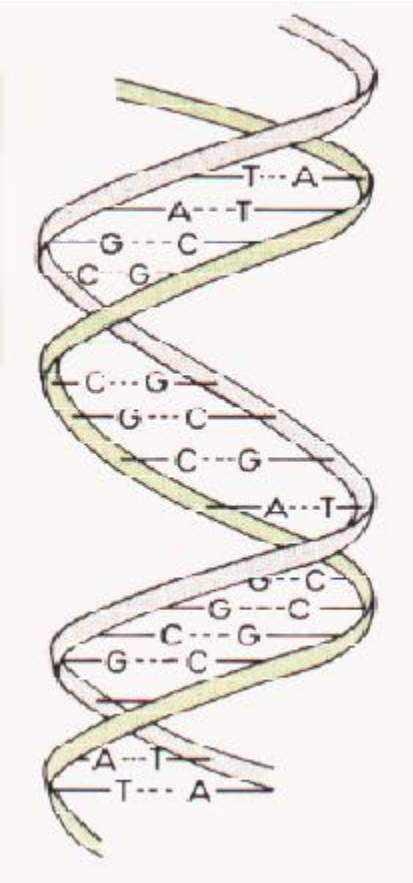


Phosphodiester bond



Hydrogen bond

The four nucleotides adenine (A), guanine (G), cytosine (C), and thymine (T) compose a strand of DNA. Each DNA strand has two different ends that determine its polarity: the 3'end, and the 5'end. The double helix is an anti-parallel (two strands of opposite polarity) bonding of two complementary strands.



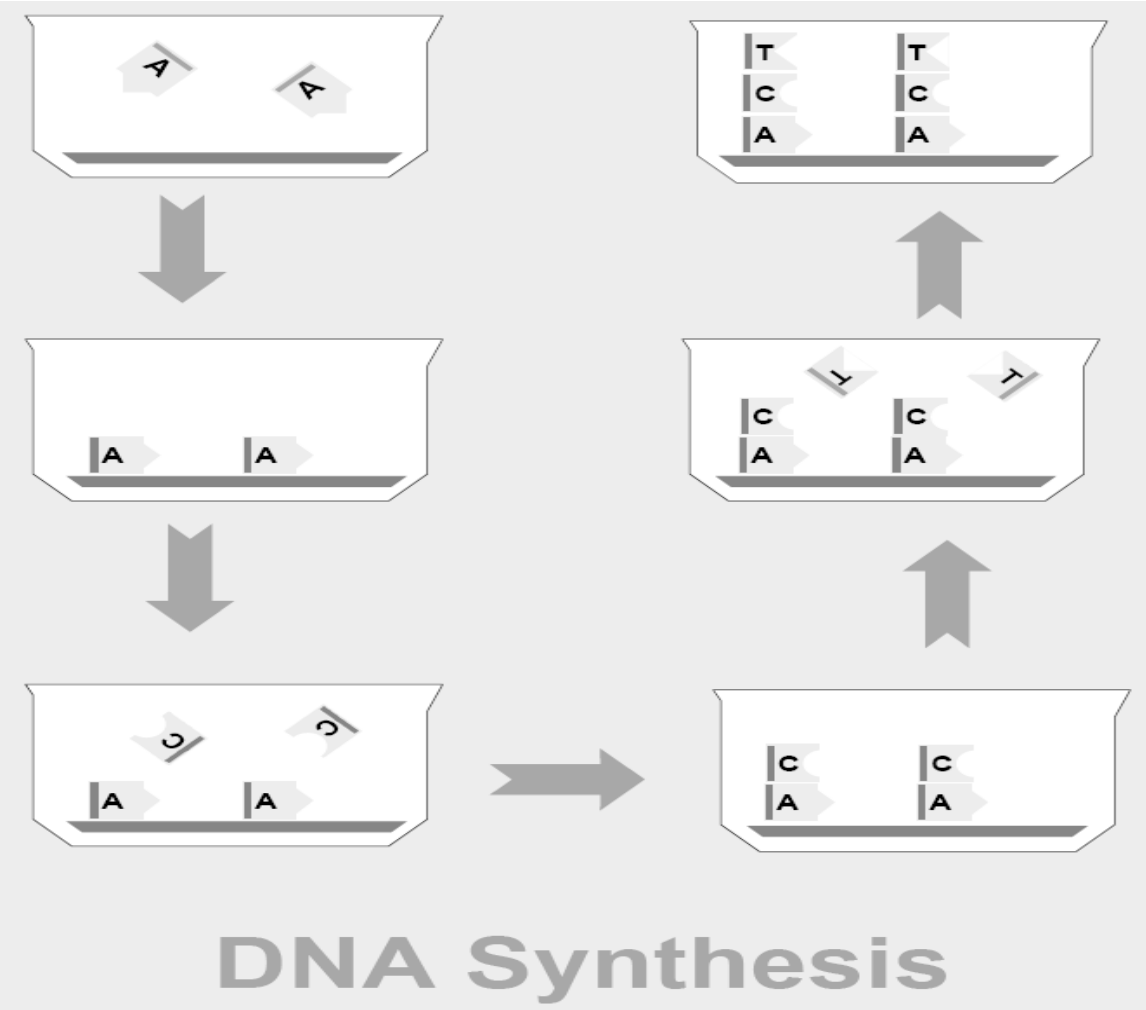
The structure of DNA double helix

Principles of DNA Computing

DNA is the major information storage molecule in living cells, and billions of years of evolution have tested and refined both this wonderful informational molecule and highly specific enzymes that can either duplicate the information in DNA molecules or transmit this information to other DNA molecules. Instead of using electrical impulses to represent bits of information, the DNA computer uses the chemical properties of these molecules by examining the patterns of combination or growth of the molecules or strings. DNA can do this through the manufacture of enzymes, which are biological catalysts that could be called the 'software', used to execute the desired calculation.

A single strand of DNA is similar to a string consisting of a combination of four different symbols A G C T. Mathematically this means we have at our disposal a letter alphabet, Σ = {A GC T} to encode information which is more than enough considering that an electronic computer needs only two digits and for the same purpose. In a DNA computer, computation takes place in test tubes or on a glass slide coated in 24K gold. The input and output are both strands of DNA, whose genetic sequences encode certain information. A program on a DNA computer is executed as a series of biochemical operations, which have the effect of synthesizing, extracting, modifying and cloning the DNA strands.
As concerning the operations that can be performed on DNA strands the proposed models of DNA computation are based on various combinations of the following primitive bio-operations:

- ***Synthesizing*** a desired polynomial-length strand used in all models.

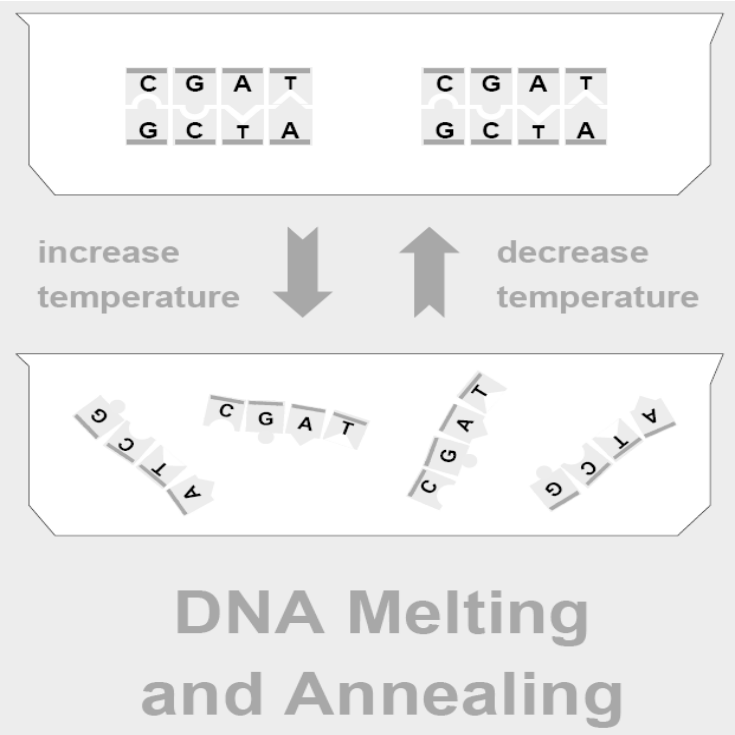


DNA Synthesis

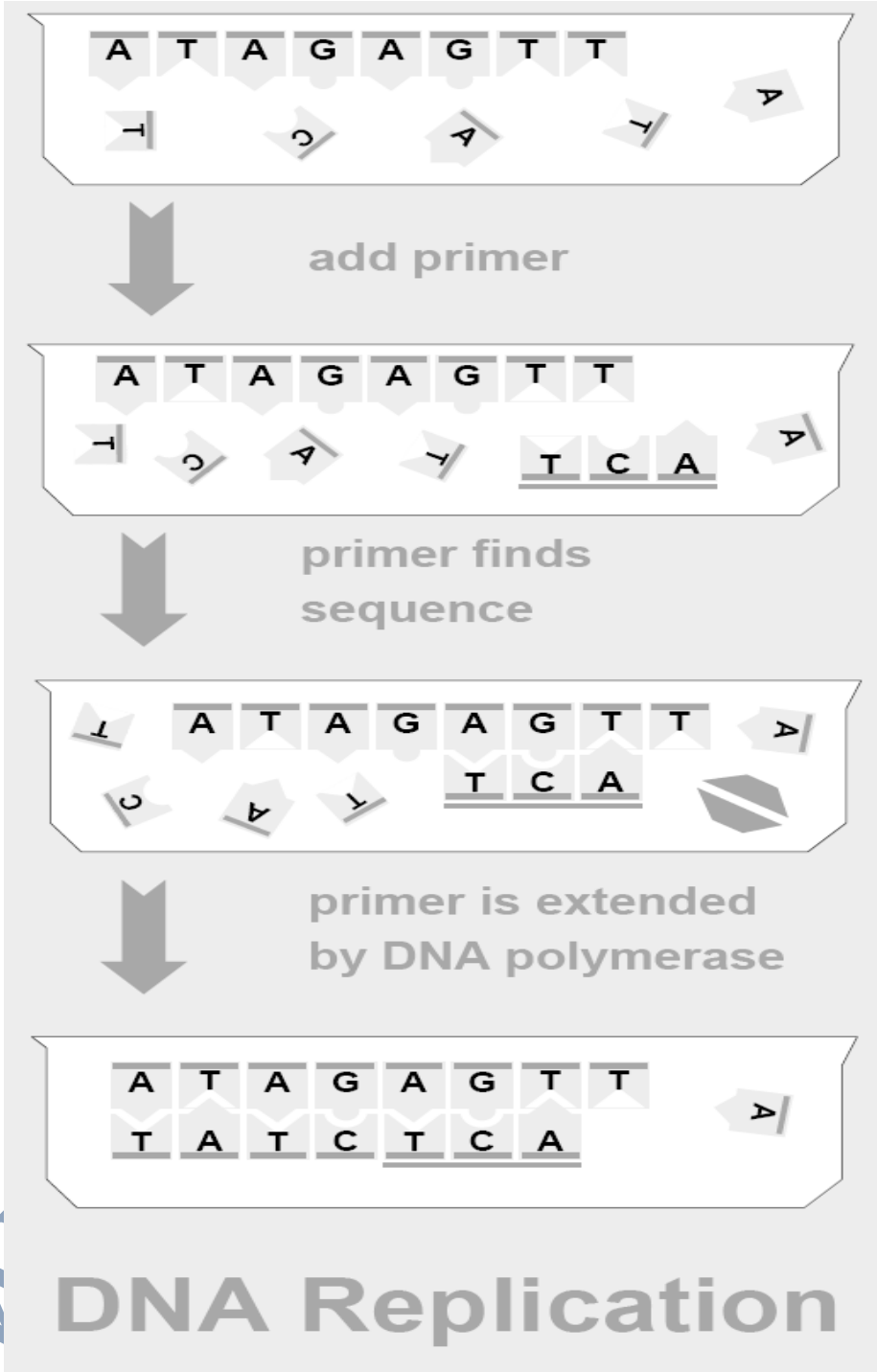
❖ ***Mixing*** **:** combine the contents of two test tubes into a third one to achieve union.

❖ ***Annealing*****:** bond together two single-stranded complementary DNA sequences by cooling the solution. Annealing in vitro is known as hybridization

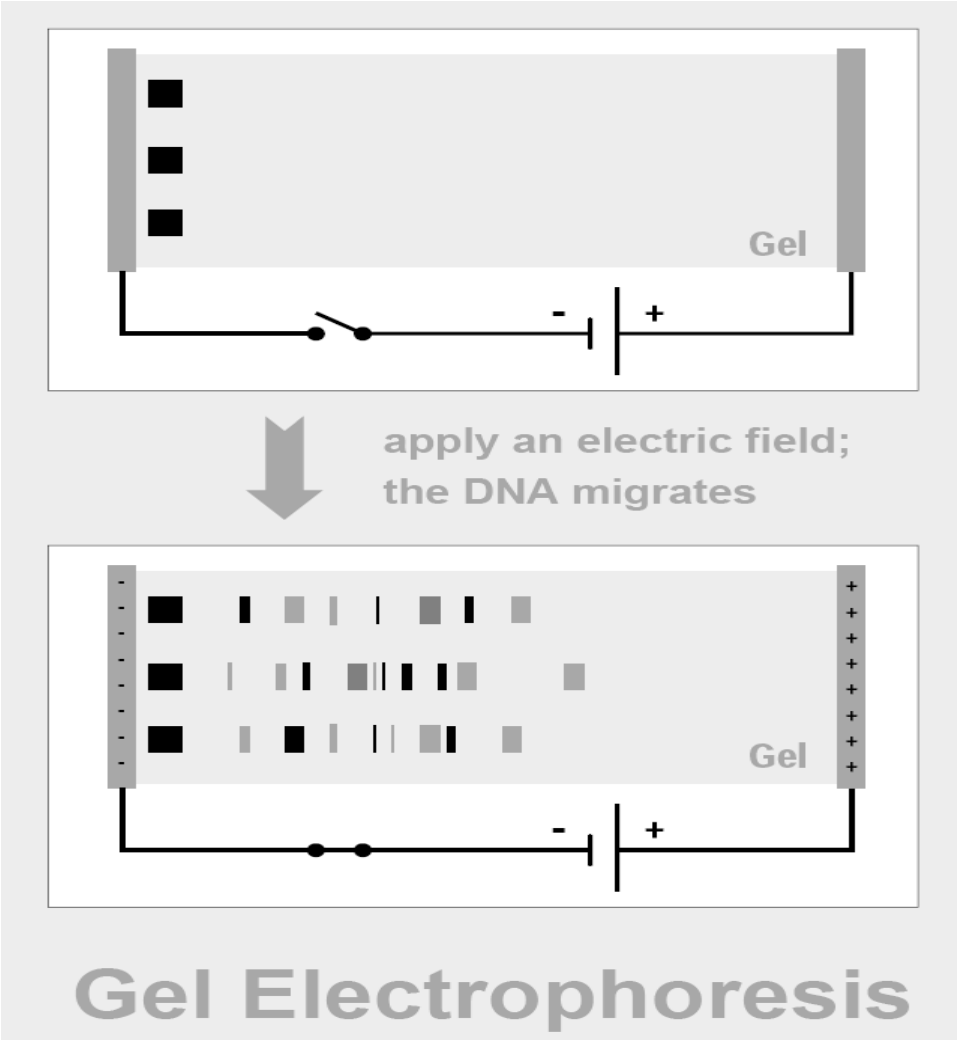
❖ ***Melting*****:** break apart a double-stranded DNA into its single-stranded complementary components by heating the solution. Melting in vitro is also known under the name of denaturation.



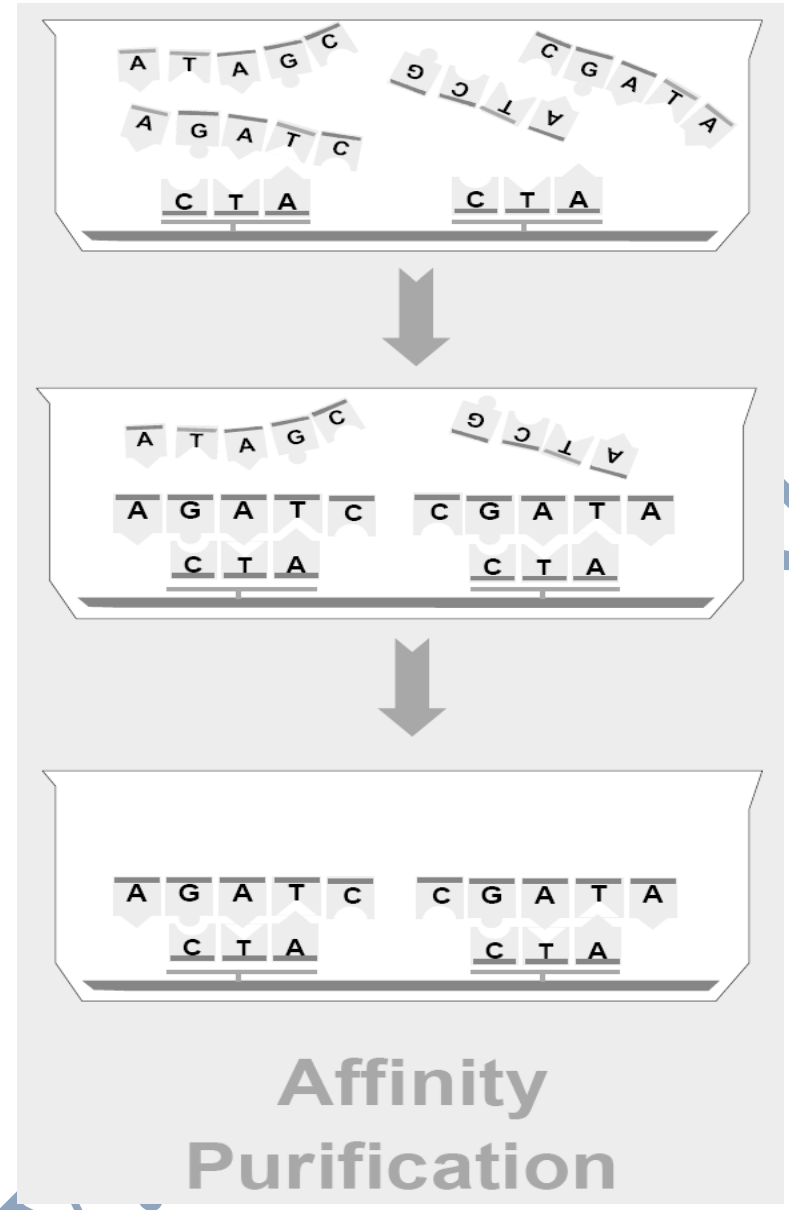
- ***Amplifying (copying)*:** make copies of DNA strands by using the Polymerase Chain Reaction PCR. The DNA polymerase enzymes perform several functions including replication of DNA. The replication reaction requires a guiding DNA single-strand called **template**, and a shorter oligonucleotide called a **primer**, that is annealed to it.



- ***Separating*** the strands by length using a technique called gel electrophoresis that makes possible the separation of strands by length.

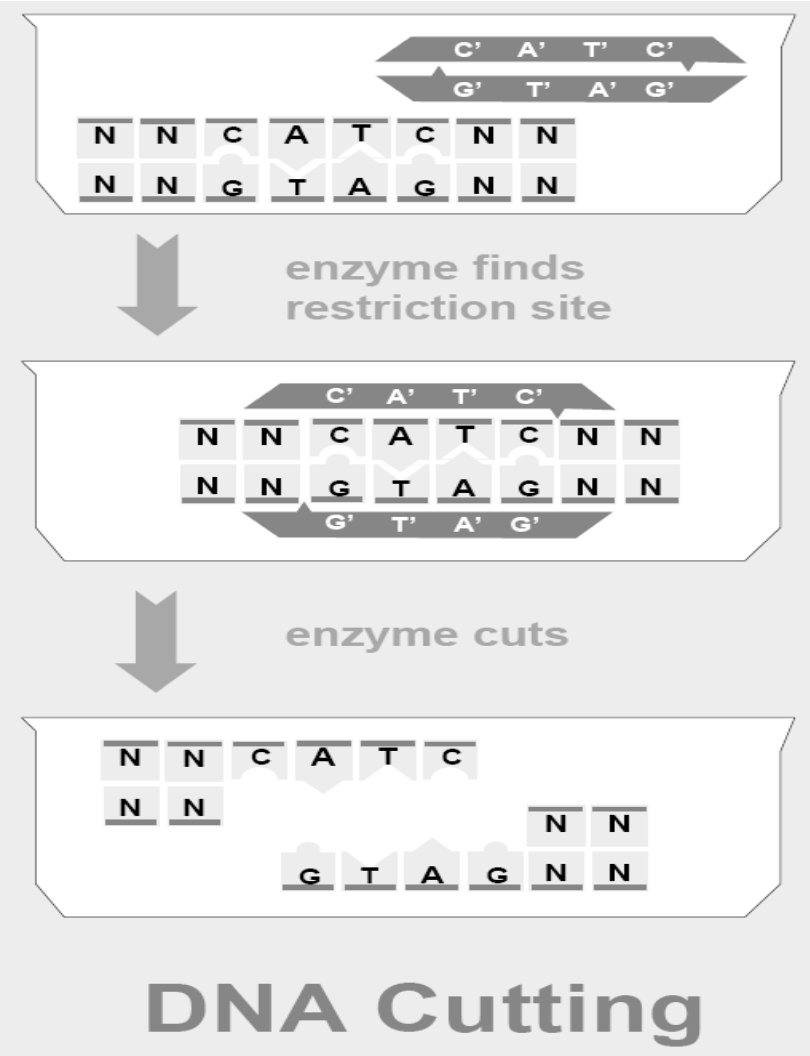


- ***Extracting*** those strands that contain a given pattern as a substring by using affinity purification.



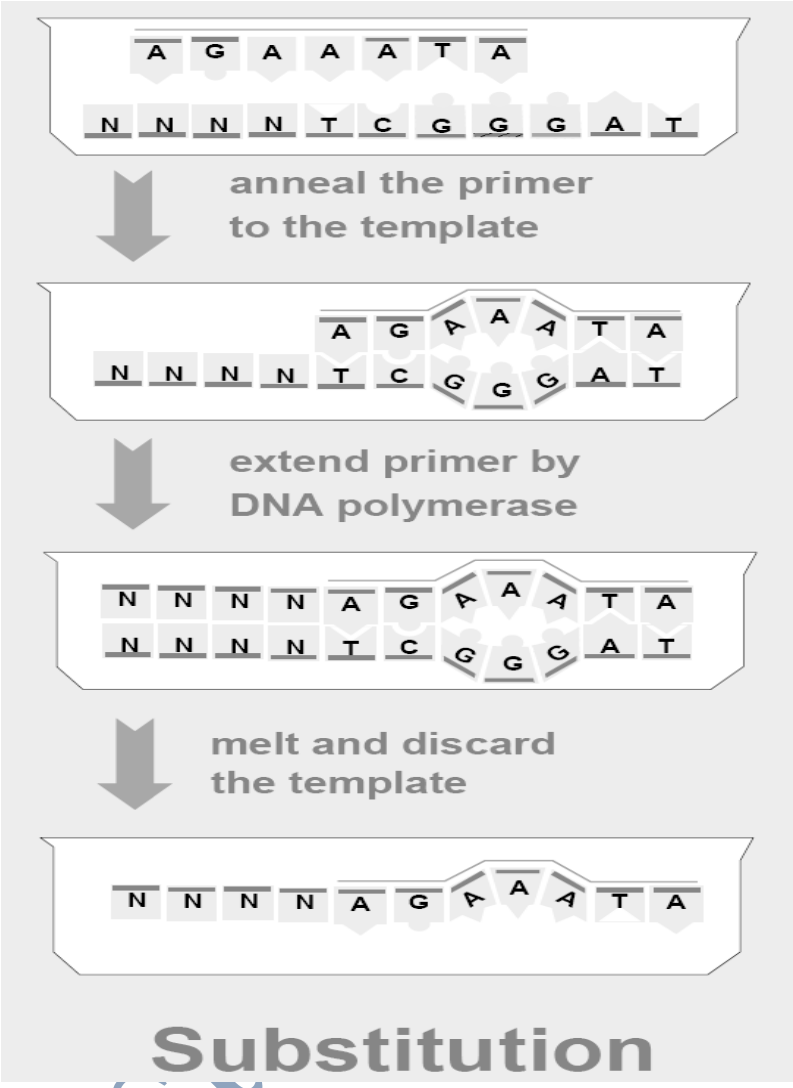
Affinity Purification

- ***Cutting*** DNA double-strands at specific sites by using commercially available restriction enzymes. One class of enzymes, called restriction endonucleases, will recognize a specific short sequence of DNA, known as a restriction site. Any double-stranded DNA that contains the restriction site within its sequence is cut by the enzyme at that location.



- ***Ligating*:** paste DNA strands with compatible sticky ends by using DNA ligases. Indeed, another enzyme called **DNA ligase**, will bond together, or ``ligate'', the end of a DNA strand to another strand.

- ***Substituting*:** substitute, insert or delete DNA sequences by using PCR site-specific oligonucleotide mutagenesis.



- ***Marking*** single strands by hybridization: complementary sequences are attached to the strands, making them double-stranded. The reverse operation is **unmarking** of the double-strands by denaturing, that is, by detaching the complementary strands. The marked sequences will be double-stranded while the unmarked ones will be single-stranded.

- ***Destroying*** the marked strands by using exonucleases, or by cutting all the marked strands with a restriction enzyme and removing all the intact strands by gel electrophoresis. (By using enzymes called **exonucleases**, either double-stranded or single-stranded DNA molecules may be selectively destroyed. The exonucleases chew up DNA molecules from the end inward, and exist with specificity to either single-stranded or double-stranded form.)

- ***Detecting*** and ***Reading***: given the contents of a tube, say ``yes'' if it contains at least one DNA strand, and ``no'' otherwise. PCR may be used to amplify the result and then a process called **sequencing** is used to actually read the solution.

In Short, DNA computers work by encoding the problem to be solved in the language of DNA: the base-four values A, T, C and G. Using this base four number system, the solution to any conceivable problem can be encoded along a DNA strand like in a Turing machine tape.

Every possible sequence can be chemically created in a test tube on trillions of different DNA strands, and the correct sequences can be filtered out using genetic engineering tools.

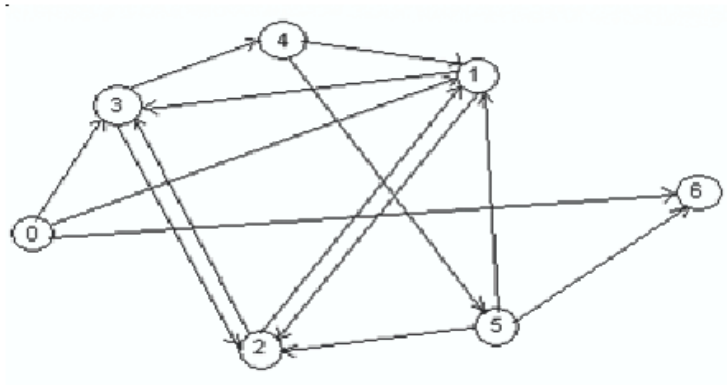
Example of DNA Computing : The Hamiltonian Path Problem

In 1994 Leonard M. Adleman showed how to solve the Hamilton Path Problem, using DNA computation.

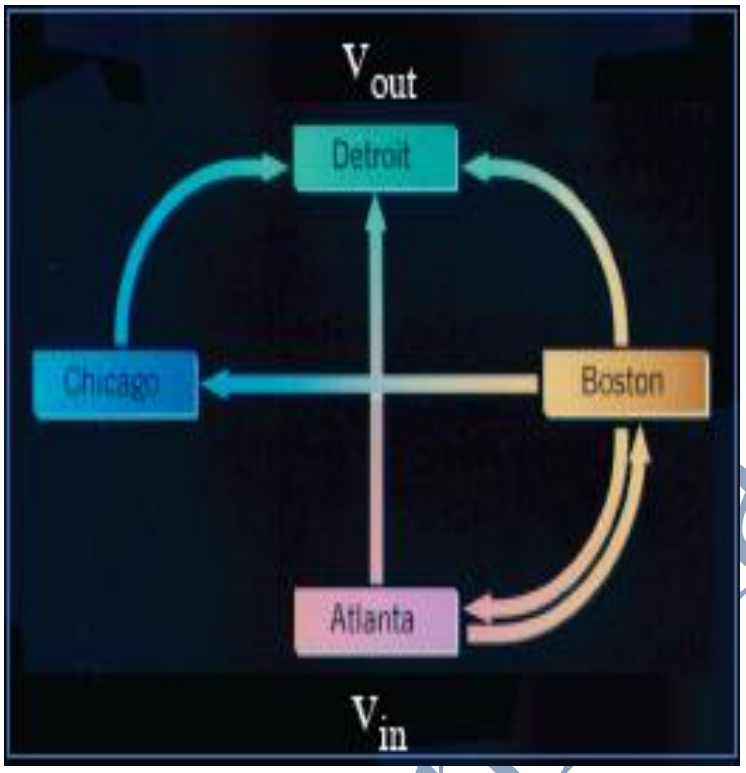
Hamiltonian Path Problem:

A directed graph G with designated nodes *vin* and *vout* is said to have a Hamiltonian path if and only if there exists a sequence of compatible one-way edges *e1, e2, ...en* that begins at *vin*, ends at *vout* and enters every other node exactly once. A simplified version of this problem, known as the traveling salesman problem, poses the following question: given an arbitrary collection of cities through which a salesman must travel, what is the shortest route linking those cities?

This problem is difficult for conventional computers to solve because it is a ”non-deterministic polynomial time problem”. These problems, when the instance size is large, are intractable with conventional computers, but can be solved using massively parallel computers like DNA computers. NP problems are intractable with deterministic (conventional/serial) computers, but can be solved using non-deterministic (massively parallel) computers. A DNA computer is a type of non-deterministic computer. The Hamiltonian Path problem was chosen by Adleman because it is known as ”NP-complete”.



Directed graph with node 0 as source (Vin) and node 6 as destination (Vout)



Simplified graph

Hamiltonian path : Atlanta–Boston–Chicago–Detroit

Adleman´s Algorithm

Input: A directed graph G with *n* vertices, and designated vertices vin and vout.
Step 1: Generate paths in G randomly in large quantities.
Step 2: Reject all paths that
- do not begin with vin and
- do not end in vout.

Step 3: Reject all paths that do not involve exactly *n* vertices.
Step 4: For each of the *n* vertices *v*:
- reject all paths that do not involve *v*.

Output: YES, if any path remains; NO, otherwise.

To implement step 1, each node of the graph was encoded as a random 20-base strand of DNA. Then, for each edge of the graph, a different 20-base oligonucleotide was generated that contains the second half of the source code plus the first half of the target node.

City	*DNA Name*	*Complement*
Atlanta	ACTTGCAG	TGAACGTC
Boston	TCGGACTG	AGCCTGAC
Chicago	GGCTATGT	CCGATACA
Detroit	CCGAGCAA	GGCTCGTT

City	*DNA Flight Number*
Atlanta - Boston	GCAGTCGG
Atlanta - Detroit	GCAGCCGA
Boston - Chicago	ACTGGGCT
Boston - Detroit	ACTGCCGA
Boston - Atlanta	ACTGACTT
Chicago - Detroit	ATGTCCGA

To implement step 2, the product of step 1 was amplified by PCR using oligonucleotide primers representing *vin* and *vout and ligase enzyme.* This amplified and thus retained only those molecules encoding paths that begin with *vin* and end with *vout.* ~1014 computations are carried out in a single second.

For implementing step 3, agarose gel electrophoresis allowed separation and recovery of DNA strands of the correct length. The desired path, if it exists, would pass through all seven nodes, each of which was assigned a length of 20 bases. Thus PCR products encoding the desired path would have to be 140 bp.

Step 4 was accomplished by successive use of affinity purification for each node other than the start and end nodes.

- The solution strand has to be filtered from the test tube:

GCAG TCGG ACTG GGCT ATGT CCGA
Atlanta → Boston → Chicago → Detroit

Thus we see in a graph with n vertices, there are a possible (n-1)! permutations of the vertices between beginning and ending vertex.
To explore each permutation, a traditional computer must perform O(n!) operations to explore all possible cycles. However, the DNA

computing model only requires the representative oligos. Once placed in solution, those oligos will anneal in parallel, providing all possible paths in the graph at roughly the same time. That is equivalent to O(1) operations, or constant time. In addition, no more space than what was originally provided is needed to contain the constructed paths.

Present & Future DNA Computer

A year ago, researchers from the Weizmann Institute of Science in Rehovot, Israel, unveiled a programmable molecular computing machine composed of enzymes and DNA molecules instead of silicon microchips. "This re-designed device uses its DNA input as its source of fuel," said Ehud Shapiro, who led the Israeli research team. This computer can perform 330 trillion operations per second, more than 100,000 times the speed of the fastest PC.

While a desktop PC is designed to perform one calculation very fast, DNA strands produce billions of potential answers simultaneously. This makes the DNA computer suitable for solving "fuzzy logic" problems that have many possible solutions rather than the either/or logic of binary computers. In the future, some speculate, there may be hybrid machines that use traditional silicon for normal processing tasks but have DNA co-processors that can take over specific tasks they would be more suitable for.

Advantages

- Perform millions of operations simultaneously.
- Generate a complete set of potential solutions and conduct large parallel searches.
- Efficiently handle massive amounts of working memory.
- They are inexpensive to build, being made of common biological materials.
- The clear advantage is that we have a distinct memory block that encodes bits.
- Using one template strand as a memory block also allows us to use its compliment. as another memory block, thus effectively doubling our capacity to store information.

Disadvantages

- Generating solution sets, even for some relatively simple problems, may require impractically large amounts of memory (lots and lots of DNA strands are required)
- Many empirical uncertainties, including those involving: actual error rates, the generation of optimal encoding techniques, and the ability to perform necessary bio-operations conveniently in vitro (for every correct answer there are millions of incorrect paths generated that are worthless).
- DNA computers could not (at this point) replace traditional computers. They are not programmable and the average dunce can not sit down at a familiar keyboard and get to work.

References

- Paun, G., Rozenberg, G., and Salomaa, A., *DNA Computing*, Springer,1998.
- DNA COMPUTING-GRAPH ALGORITHMS [lec-12.pdf] G. P. Raja Sekhar, Dept. of Mathematics, IIT, Kharagpur
- Leonard M. Adleman, *Computing with DNA*, Scientific American, August 1998.
- From Microsoft to Biosoft Computing with DNA, Lila Kari, Department of Computer Science University of Western Ontario
- L.Adleman. On constructing a molecular computer. 1st DIMACS workshop on DNA based computers, Princeton, 1995. In DIMACS series, vol.27 (1996)
- L.Adleman, P.Rothemund, S.Roweis, E.Winfree. On applying molecular computation to the Data Encryption Standard. 2nd DIMACS workshop on DNA based computers, Princeton, 1996