

DNA COMPUTING

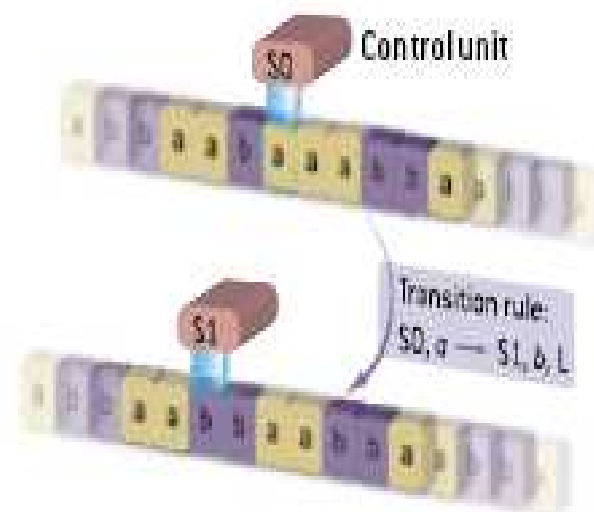
Turing Machine and Automata built from DNA
And Bio-molecular Computers

BY: Neha Sharma

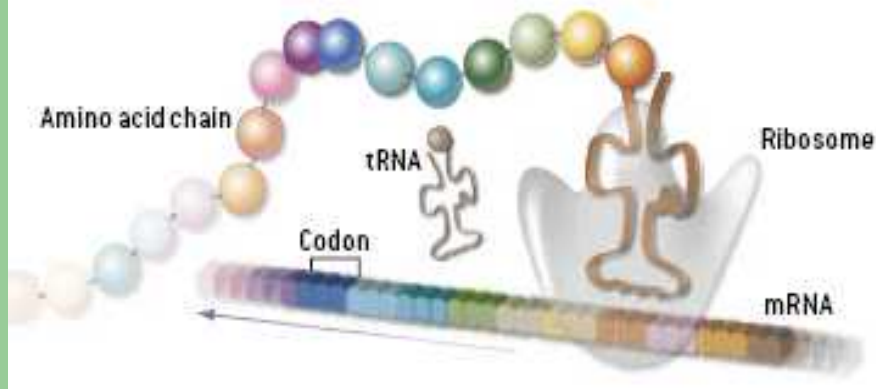


Turing Machine

- Information is stored on a tape using symbols such as 'a' & 'b'.
- Read/Write control unit processes tape using instructions provided by transition rules.
- Only one symbol position is processed at a time.



Biological Machine



mRNA is made up of nucleotide trios called codons.

- Information is encoded in gene transcripts known as mRNAs.
- Ribosomes reads the information from mRNA
- Only one codon of mRNA is processed at a time.
- The tRNA confirms the codon match then releases the amino acid to join the growing chain.

Similarity b/w Turing m/c & Bio-molecular m/c

- In both processes, information is stored in a string of symbols.
- Both operate by moving step by step along those strings.
- Symbols are modified or added according to the set of rules followed.

Basic DNA operations used to built a Turing m/c:

- Recognition of molecular building blocks.
- Cleavage & ligation of biopolymer molecules.
- Movement along a polymer
(either left or right)

Molecular Turing m/c model:

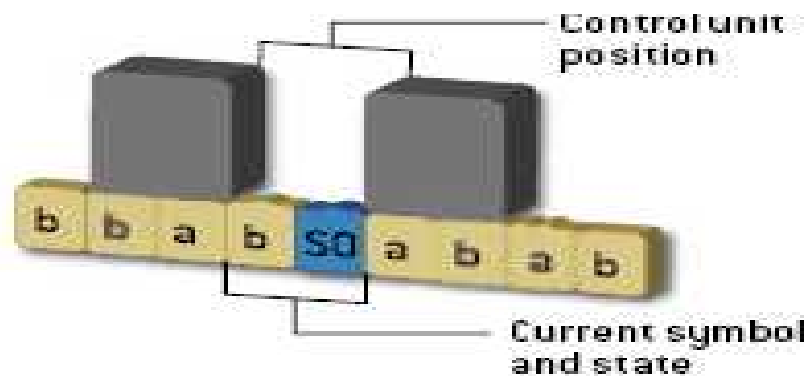
- Natural ability:
 1. To recognize symbols.
 2. To cleave the bond or to join the molecular subunits together.

Design of Molecular Turing m/c using an example:

- Machine with just (say) 2 states (S1, S2)
- Program consisting of 4 statements called set of rules.
- Double stranded DNA molecule as input string.
- 4 more short double stranded DNA molecule as transition rules or software.
- 2 natural DNA manipulating enzymes, FokI & ligase as hardware.

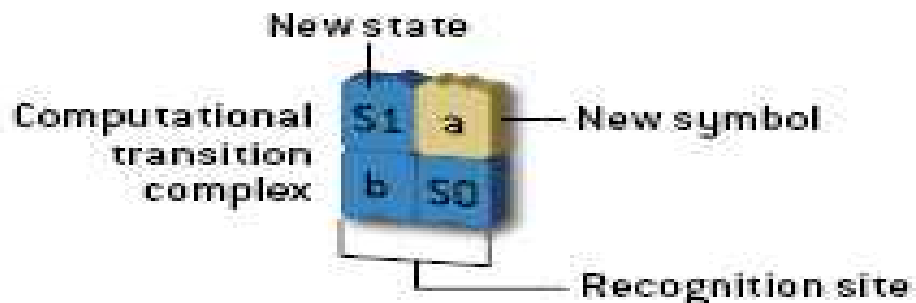
How it works?

- Operates on string of symbol molecules.
- In middle of control unit position, both symbol and machine's current state are defined.



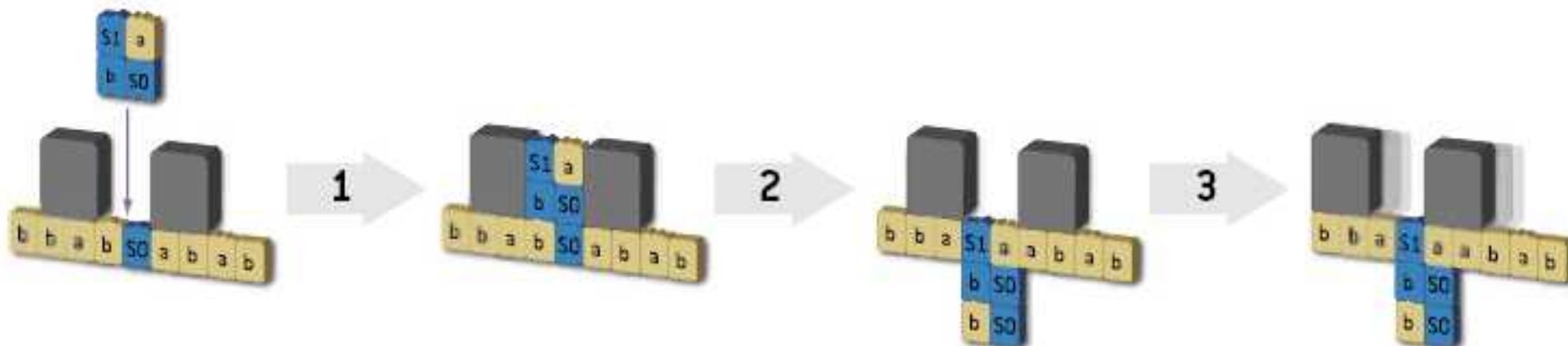
How it works continued...

- One computation transition is represented by:
 1. a molecule complex containing a new state & symbol.
 2. and a recognition site to detect the current state & symbol.



Continued...

A free-floating computational transition complex slides into the machine's control unit (1). The molecule complex binds to and then displaces the current symbol and state (2). The control unit can move one position to the left to accommodate another transition complex (3). The process repeats indefinitely with new states and symbols as long as transition rules apply.



Molecular Turing machine model

- A plastic model of molecular Turing machine:
- Yellow molecule blocks carry the symbols.
- Blue s/w molecules indicate States and define transition Rules.



Automata

- Devices that convert information from one form to another according to a definite procedure are known as automata.
- Automata is a special case of Turing machine operate by scanning a data tape.

Continue...

- It is a national computing machine that operates on finite sequences of symbols.
- The machine can be in one of a finite number internal states i.e. initial state and some as accepted states.
- Software consists of transition rules that specifies the next state.

Basic features and processes:

(with 2 internal states S_0 & S_1 and 2 i/p symbols a & b)

- 8 possible transition rules (T1-T8)
- Programs to select some of these transition rules.
- Deciding which internal states are accepting.

Continued...

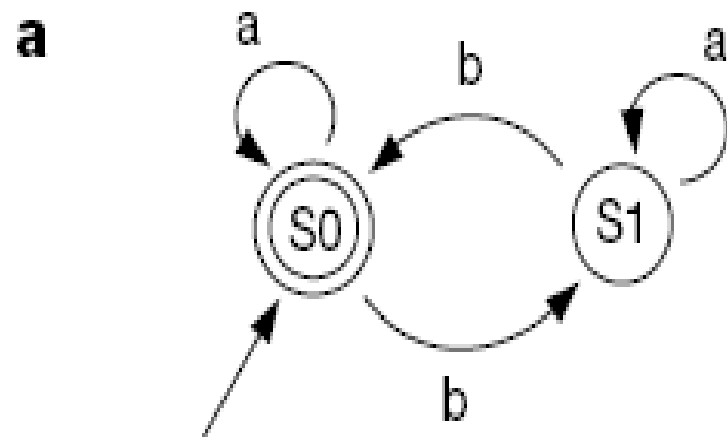
- These are the 8 possible transition rules with two input symbols 'a' & 'b' and with two internal states S0 & S1.

T1: S0	\xrightarrow{a}	S0
T2: S0	\xrightarrow{a}	S1
T3: S0	\xrightarrow{b}	S0
T4: S0	\xrightarrow{b}	S1
T5: S1	\xrightarrow{a}	S0
T6: S1	\xrightarrow{a}	S1
T7: S1	\xrightarrow{b}	S0
T8: S1	\xrightarrow{b}	S1

Continued...

- For a given program transition rules can be selected.
- State with the incoming arrow is the initial state.
- Double circle represents the accepting state.

Continued...



b

S0-abaaba ($S0 \xrightarrow{a} S0$)
S0-baaba ($S0 \xrightarrow{b} S1$)
S1-aaba ($S1 \xrightarrow{a} S1$)
S1-aba ($S1 \xrightarrow{a} S1$)
S1-ba ($S1 \xrightarrow{b} S0$)
S0-a ($S0 \xrightarrow{a} S0$)

S0 (final state)
The input is accepted

How to build a molecular automaton?

Raw material for building:

- DNA strand to serve both as input and software
- DNA enzymes as hardware.
- Nucleotides A,G,C,T encoded as symbols and machine's internal state.

Continued...

Hardware:

- Class IIS restriction nuclease FokI
- T4 DNA Ligase.
- ATP.

Software:

- 8 Short double stranded DNA molecules.
- Transition molecules encoded 8 possible transition rules.

Continued...

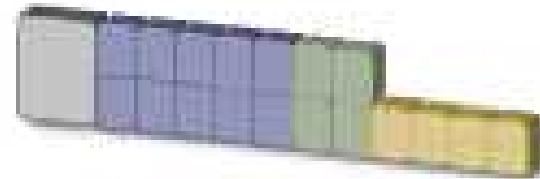
- Hardware: FokI recognizes the nucleotide sequence GGATG and cut a double stranded DNA at positions 9 and 13 downstream of that recognition site.



Continued...

Software consists of:

- Transition rules, encoded in 8 short double stranded DNA molecules containing the FokI recognition site (blue).
- followed by spacer nucleotides.
- Single stranded sticky end that will join to its complement on an input molecule.



Continued...

Transition molecules

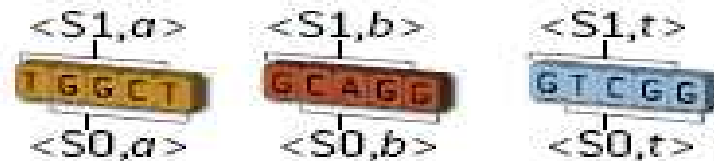
FokI recognition site



Continued...

Symbols and states:

- Combination of symbol and machine state is represented by four nucleotide sequence.
- This combination depends on how the nucleotide sequence is cleaved into four nucleotides.



Continued...

■ Symbols and states encoding

Symbol	a	b	terminator (t)
Encodings & <state, symbol> sticky ends	<p><S1, a> ↑ CTGGCT ↓ <S0, a></p>	<p><S1, b> ↑ CGCAGC ↓ <S0, b></p>	<p><S1, t> ↑ TGTCGC ↓ <S0, t></p>

Continued...

- System also contains two detection molecules.
- Each detection output molecule interacts selectively with different output molecule to form an output reporting molecule that indicates a final state (detected by gel electrophoresis).

■ Output-detection molecules

161 AGCG

S0-D

251 ACAG

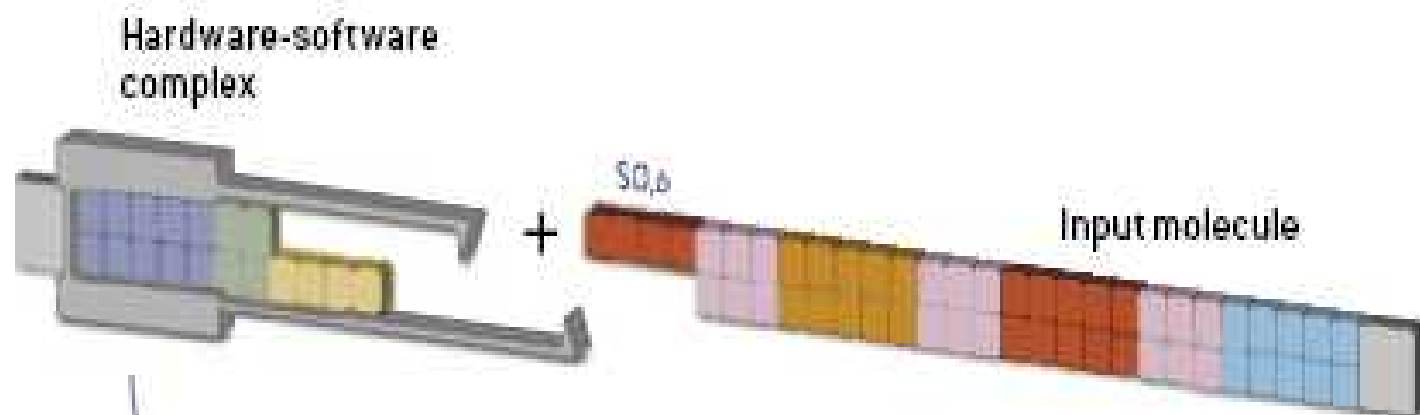
S1-D

Autonomous Computation:

- Initially it is placed on the leftmost input symbol in the initial state.
- In each transition the machine moves to the right changing its internal state.
- Computation terminates either when no rule applies or last symbol of input is being processed.
- An input is accepted if the computation ends in an accepting final state.

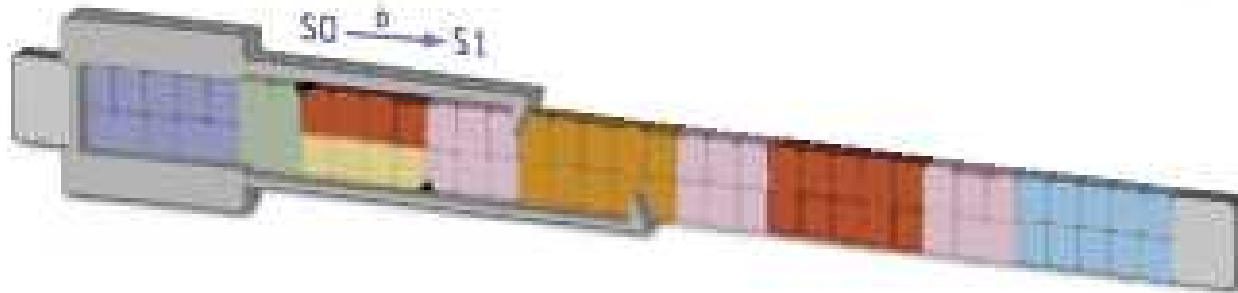
Continued...

- A hardware-software complex recognizes its complementary state/symbol combination on the input molecule.



Continued...

- The molecules join to form a hardware software input complex.



Continued...

This is how the molecule looks like.

■ Example input molecule

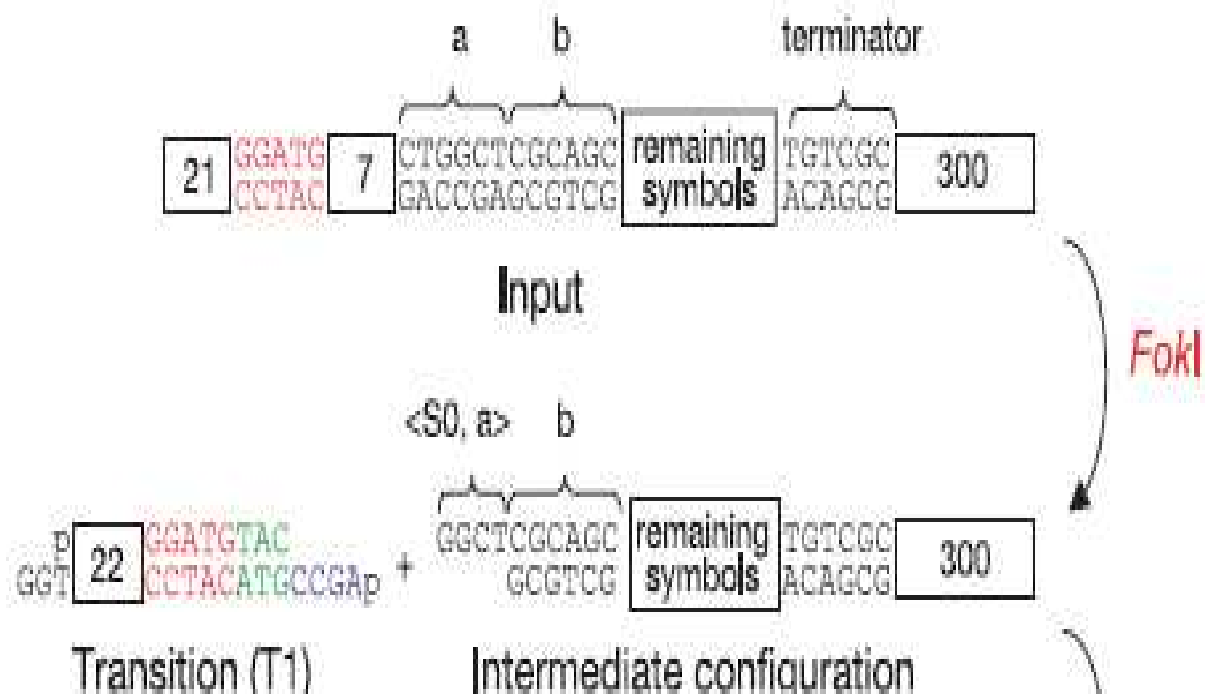


Continued...

- The FokI cleaves the input molecule to expose the next symbol.
- Input cleaved by FokI will expose 4 nucleotide sticky end, encoding initial state and 1st input symbol.
- Computation proceeds via cascade of transition cycles.
- In each cycle, sticky end of an applicable transition molecule ligates to input molecule, detecting the current state and symbol.

Continued...

Mechanism

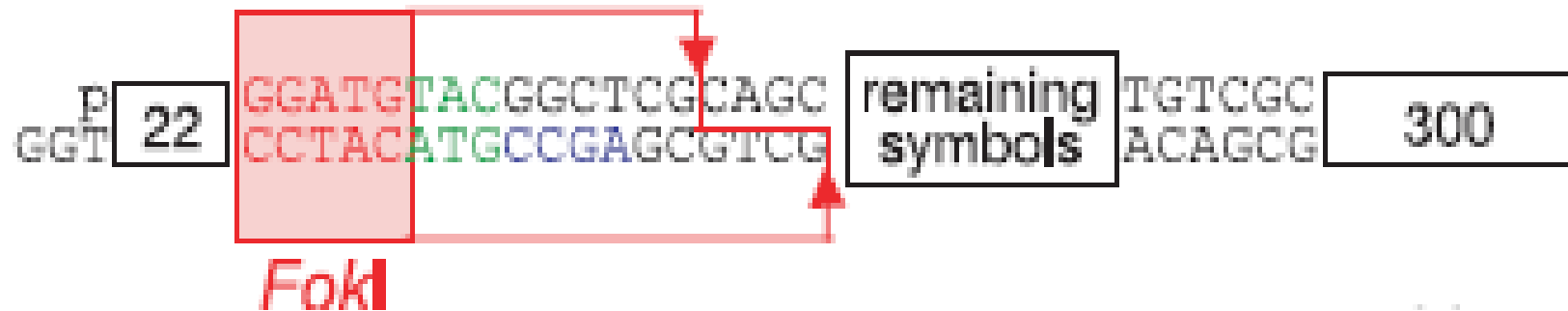


Continued...

- A new hardware software complex recognizes the next state and symbol on what remains on the input molecule.
- Reactions continued until no rule applies or the terminal symbol is revealed.

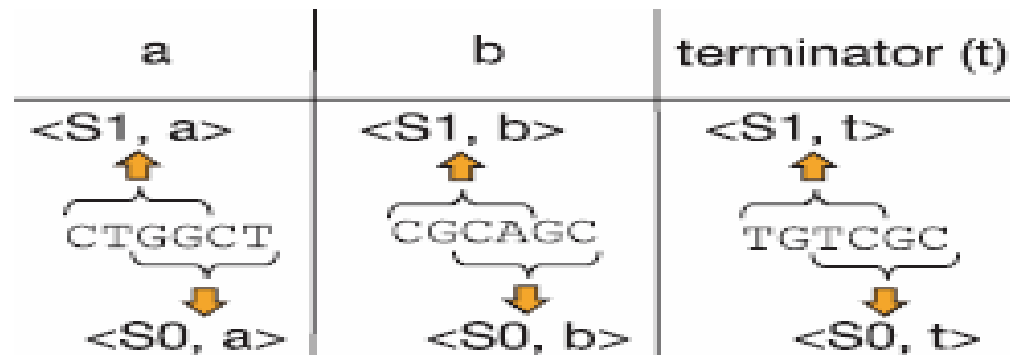
Continued...

- The product is again cleaved by FokI inside the next symbol encoding & exposing a four nucleotide.

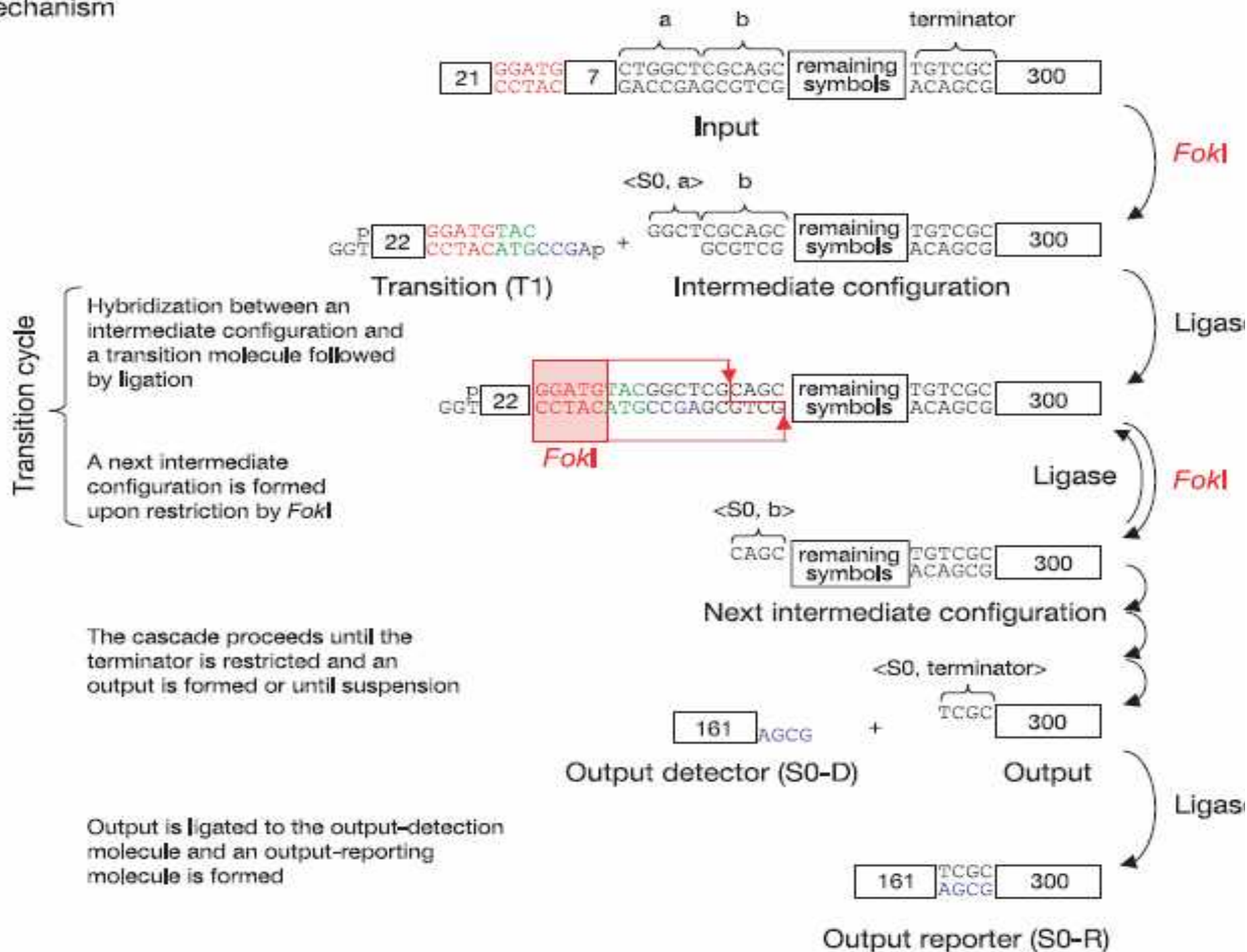


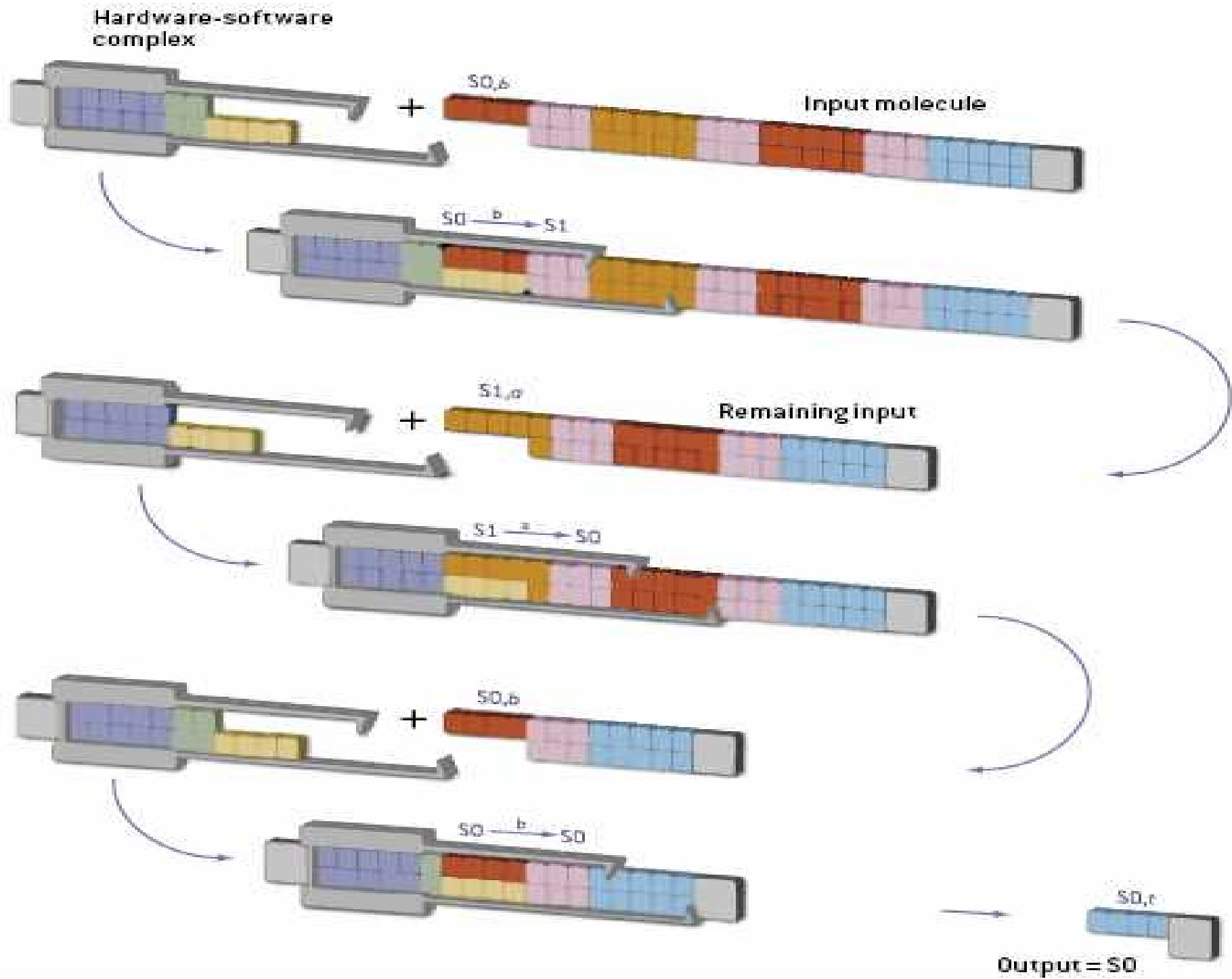
Continued...

- 6 base pair long symbols 'a' & 'b' are cleaved by FokI at only 2 different frames.
- Leftmost frame encoding S1 & rightmost frame encoding S0.



Mechanism





Continued...

- The computation proceeds until no transition molecule matches the exposed sticky end of the input or until the special terminator symbol is cleaved, forming an output molecule.
- The output molecule sticks to the output detector & the gel electrophoresis output is identified.

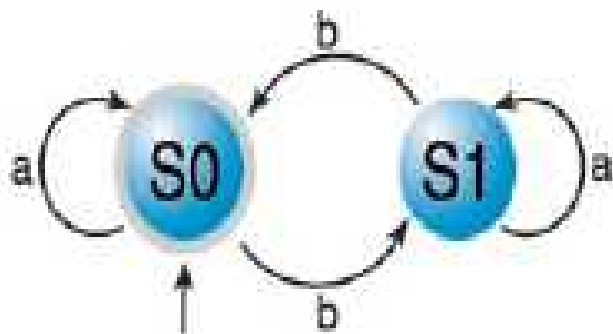
Stochastic Finite Automata

- In this, the output of the computation is distributed over the final states rather than a single final state.
- Unlike deterministic automata that selects at most one transition rule for each state symbol, stochastic automata uses all transition rules, ascribing each transition with a pre-defined probability.

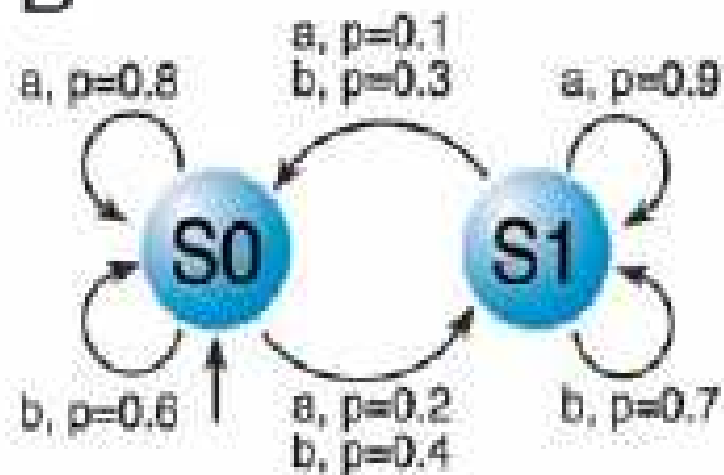
Continued...

- The output of a stochastic computation is the probability to obtain each expected final state.
- Stochastic automata are useful for the analysis of sequence or process that are not deterministic.

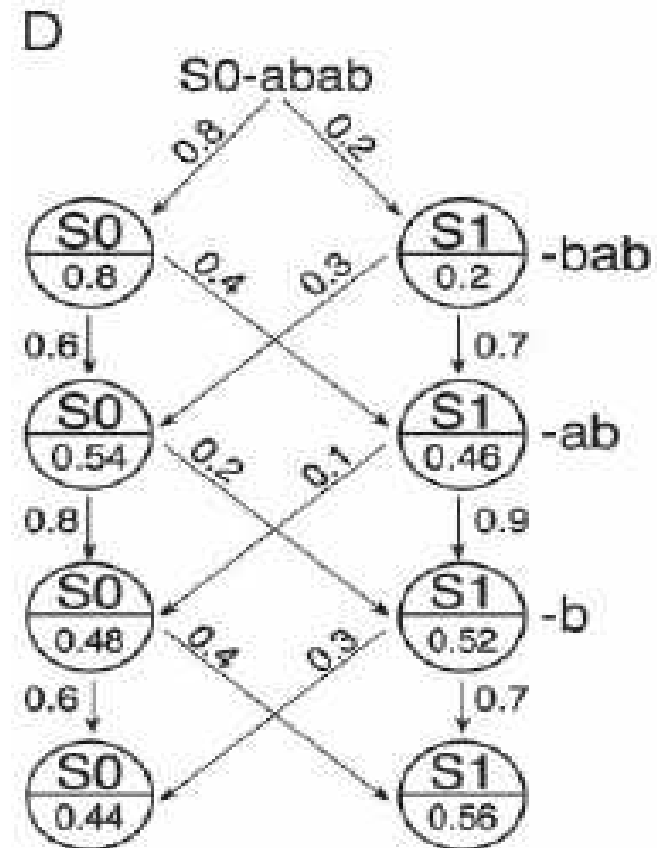
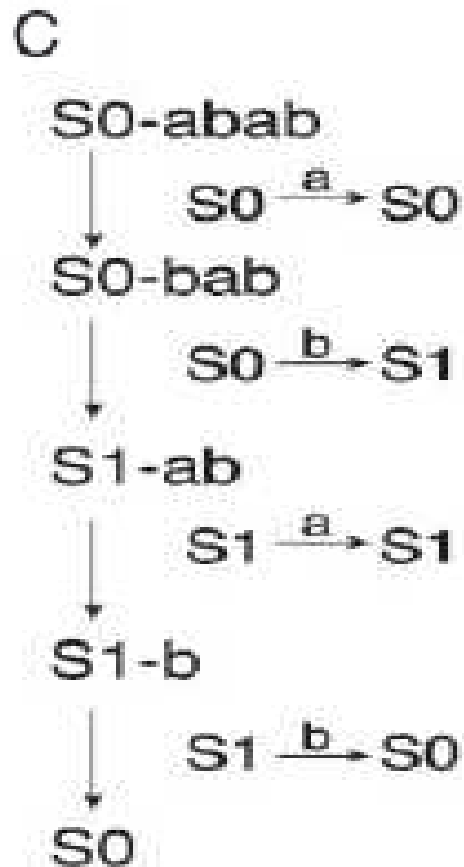
A Deterministic finite automaton.



B Stochastic finite automaton.



Computation Path of Deterministic and Stochastic automata



Bio-molecular Computers

- These are the programmable, autonomous computing machines in which the: input, output, software and hardware are made of biological machines.
- Such computers could produce a system for logical control of biological processes.

Modules of bio-molecular Computers:

- Computational module (a stochastic molecular automata)
- Input module: by which specific mRNA levels regulate software molecule concentrations & hence automaton transition probabilities.
- Output module: capable of controlled release of a short single stranded DNA molecule.

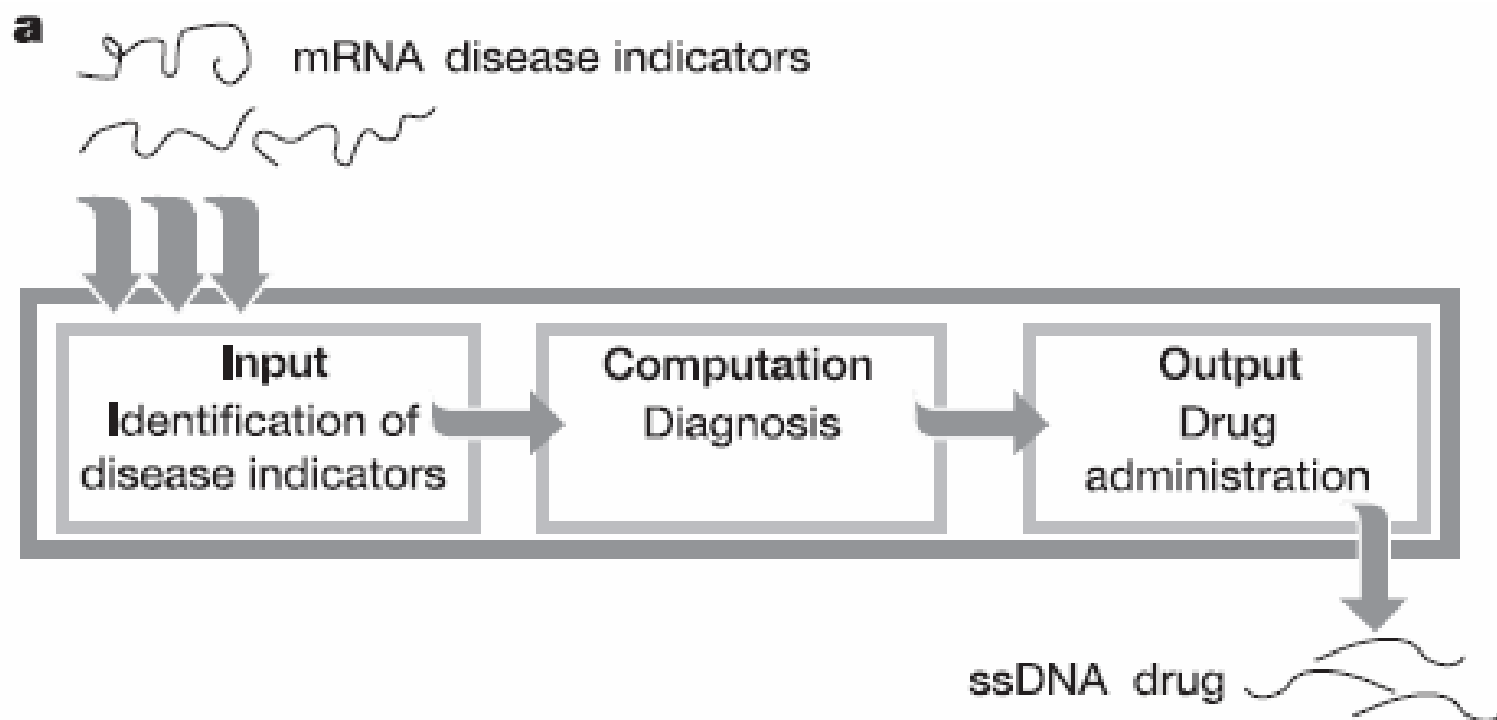
Bio-molecular Computer as a Doctor

Within a cell:

- It could sense signals from environment indicating disease.
- Can process disease using programmed medical knowledge.
- Output in the form of a therapeutic drug (molecule).

How do they work in medical diagnosis

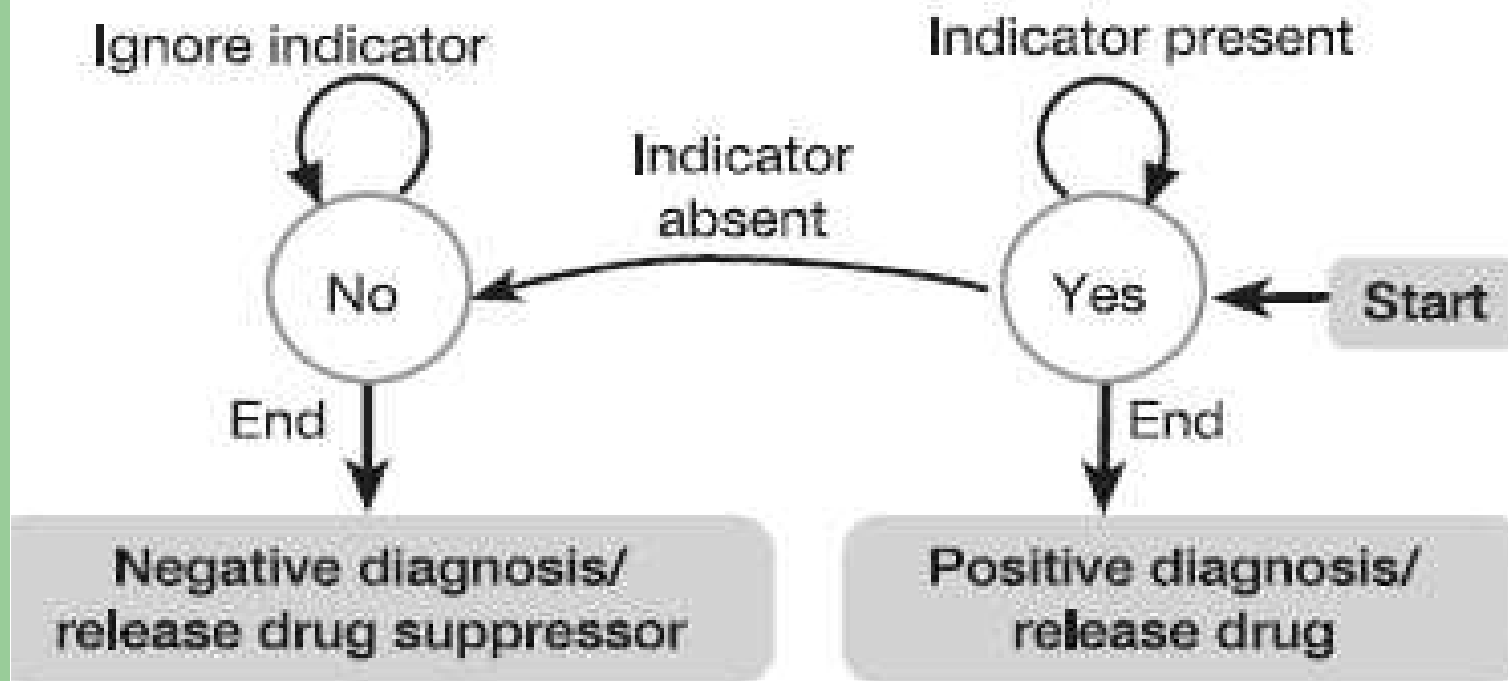
- Perform identification of mRNA molecules at specific levels say for cancer.
- Produce biologically active molecules i.e. a therapy which in this case is a single stranded DNA with known anti cancer activity.



Computation module of Bio-molecular Computer.

- Computation module is a molecular automaton that processes rules
- Automaton has two states:
 - positive means yes
 - negative means no
- Computation starts in positive state and if it ends in that state we call the result positive diagnosis otherwise negative diagnosis.

Continued...



Continued...

- Because of probabilistic nature of computation and imprecise behaviour of bio-molecular elements, some computations may end inevitably with a positive diagnosis even if the disease symptoms were absent and vice versa.
- This problem can be solved by using system of checks and balances.

How the problem can be solved?

Output :

- After positive diagnosis, a single stranded DNA drug is released.
- When disease indicators are absent, automata releases a drug suppressor.
- With thousands of both type of diagnostic molecules computing simultaneously, majority will make the correct diagnosis.

Continued...

- For this instead of using a single automaton, 2 types of automata are used:
 1. One that releases drug on positive diagnosis.
 2. And another that releases drug suppressor molecule on negative diagnosis.

**Diagnostic molecule
in final yes state**



Positive diagnosis



**Antisense
drug release**



**High level of
active drug**

Antisense



**Low level of
suppressed drug**

**Antisense
suppressor**



Thank You

Neha Sharma

