Mathematics in Bio-Computing

M. Sakthi Balan

ECOM Research Lab
Education & Research
Infosys Techologies Limited
Bangalore

Contents

Contents

Infosys

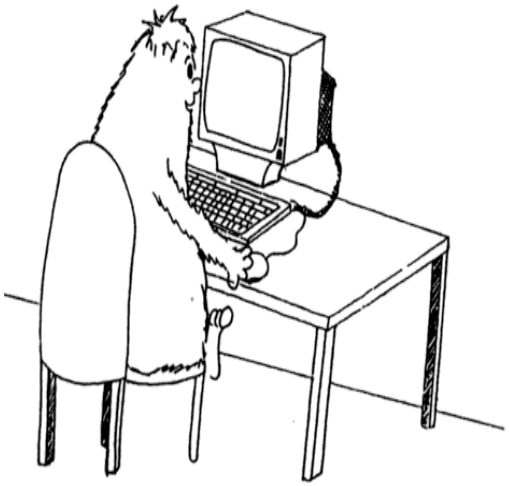
Contents

Infosys

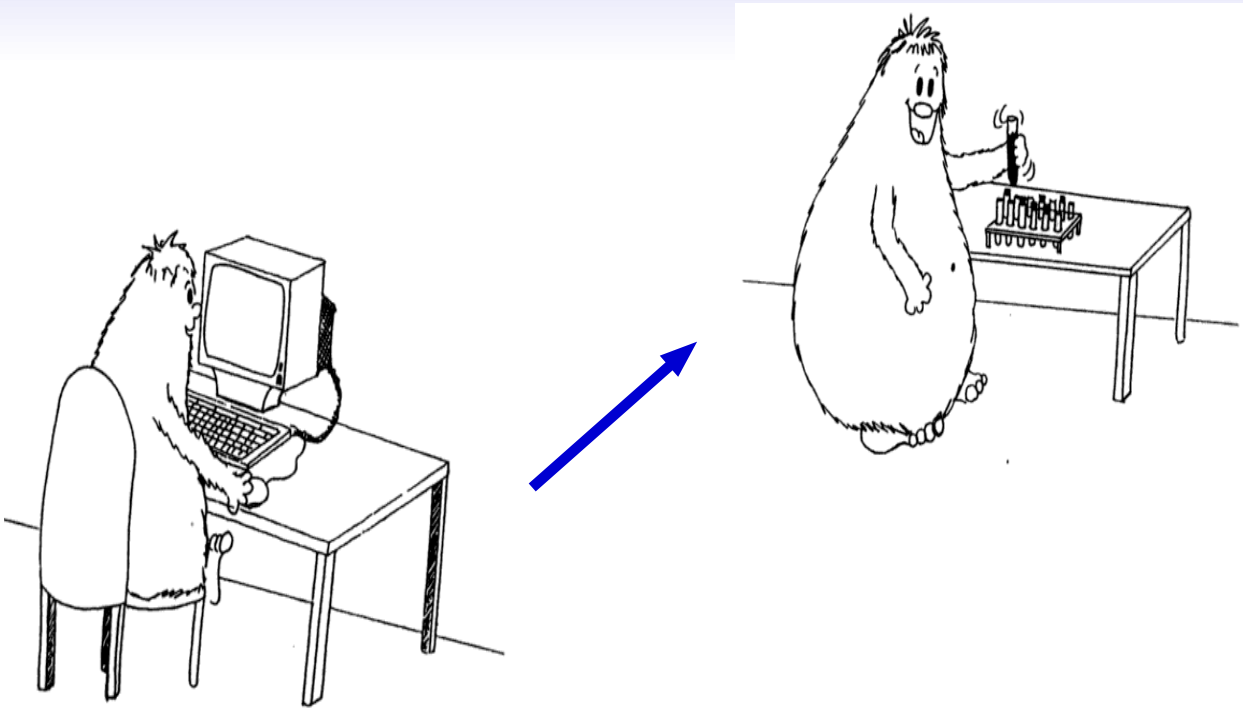
Contents

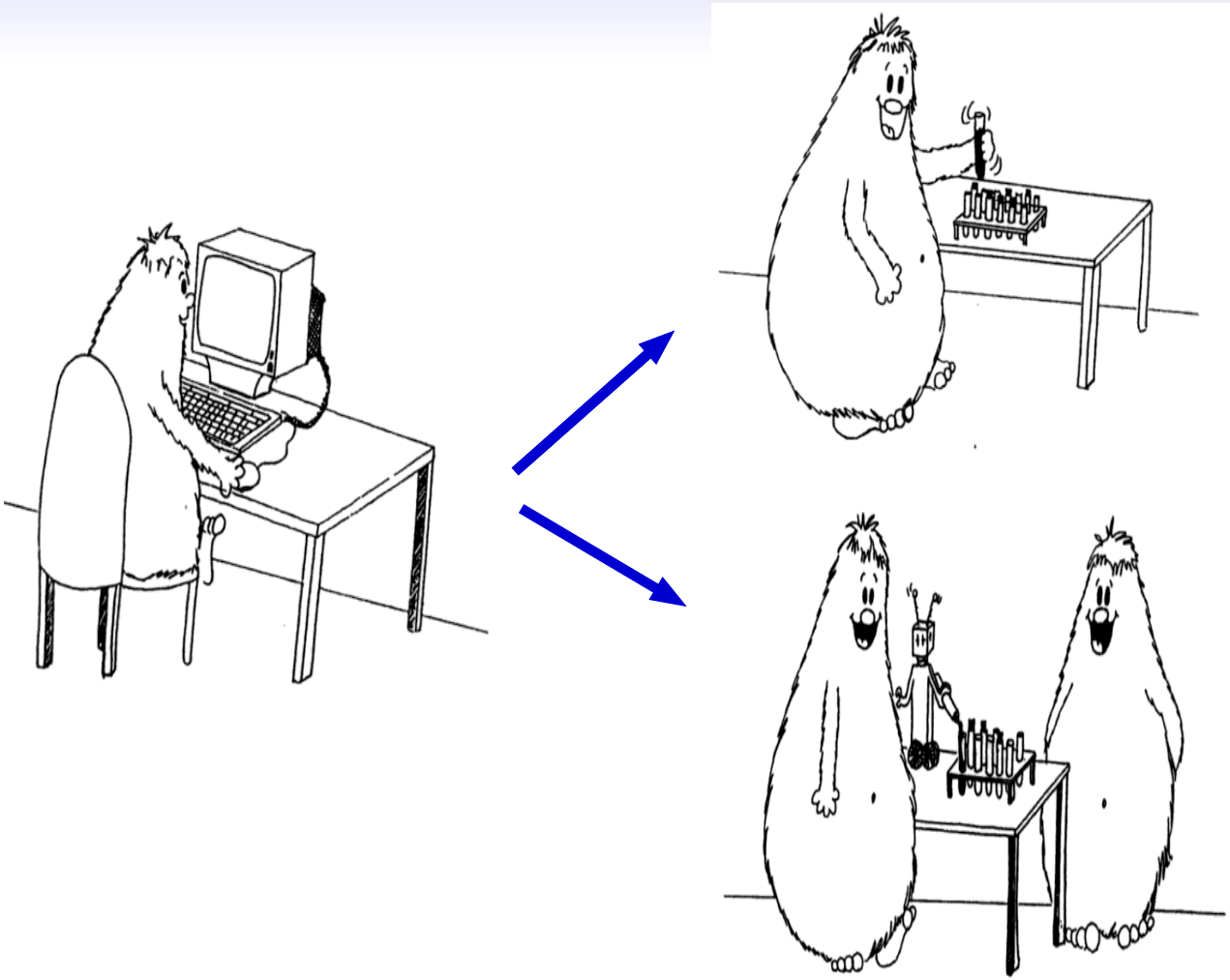
Infosys

Contents



Infosys®





Pictures from book: DNA Computing. New Computing Paradigms

Richard Feynman's visionary talk on sub-microscopic computers.

From *Computing with Cells and Atoms* by Cristian S. Calude and Gh. Păun

> *...It seems that progress in electronic hardware (and the corresponding software engineering) is not enough; for instance, the miniaturization is approaching the quantum boundary, where physical processes obey laws based on probabilities and non-determinism, something almost completely absent in the operation of classical computers. So, new breakthrough is needed...*

‘

Infosys®

Natural Computing

Bio

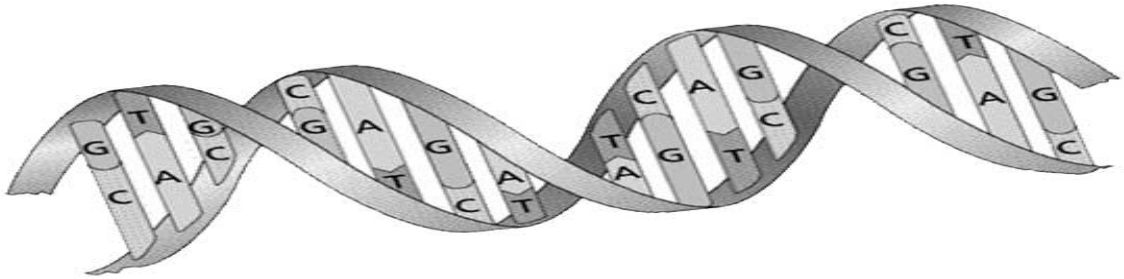
- DNA hybridization
- Immune reaction
- Membrane computing using cells

Others

- Quantum mechanical phenomena
- Reaction-diffusion process

Infosys

DNA Strands



- DNA consists of polymer chains – DNA strands.
- Chain consists of nucleotides that differ only in their bases.
- There are four bases: *A* (adenine), *G* (guanine), *C* (cytosine) and *T* (Thymine).
- Double-helical structure is formed through bonding of two strands. Watson-Crick complementarity – *A* always bonds with *T* and *G* with *C*.
- Orientation of DNA strands: 5′ to 3′(upper) and 3′ to 5′(lower).

Operations on DNA

- The length of a DNA strand can be measured using gel electrophoresis.
- A known DNA strand can be fished in a solution containing many DNA strands using a filtering method.
- A double stranded DNA can be denatured into two single strands by heating.
- Two DNA strands can be hybridized into a single double stranded DNA molecule (DNA strands bind together respecting the Watson-Crick complementary property).

Infosys®

Operations on DNA

- A class of enzymes called polymerase can lengthen a partially double stranded to make it as a complete double stranded molecule.
- Enzymes called restriction endonucleases cut DNA strands at the specific site where a specific sequence of nucleotides are present.
- Enzymes ligases can paste two DNA strands with overhanging ends provided those ends are Watson-Crick complementary of each other. This process is called ligation.
- Specific set of DNA sequences can be multiplied using a polymerase chain reaction.
- The exact sequence in the DNA strand can be found out by polymerase action on the strand. This process is called sequencing.

Infosys®

DNA Strands

```
TATAGCCGCTCGATTACGGC
          GCTAATGCCG CGGCGCGTAT
```

Two sticky ends

DNA Strands

```
TATAGCCGCTCGATTACGGC
                GCTAATGCCG CGGCGCGTAT
```

Two sticky ends

```
TATAGCCGCTCGATTACGGC
                GCTAATGCCG
```

One sticky end

DNA Strands

```
TATAGCCGCTCGATTACGGC
                GCTAATGCCG CGGCGCGTAT
```

Two sticky ends

```
TATAGCCGCTCGATTACGGC
                GCTAATGCCG
```

One sticky end

```
TATAGCCGCTCGATTACGGC  GCCGCGCATATACGATGTAT
                GCTAATGCCG CGGCGCGTAT
```

Two sticky ends

DNA Strands

```
TATAGCCGCTCGATTACGGC
                GCTAATGCCG CGGCGCGTAT
```

Two sticky ends

```
TATAGCCGCTCGATTACGGC
                GCTAATGCCG
```

One sticky end

```
TATAGCCGCTCGATTACGGC  GCCGCGCATATACGATGTAT
                GCTAATGCCG CGGCGCGTAT
```

Two sticky ends

```
GCCGCGCATATACGATGTAT
```

Single strand

About complementarity

- Watson-Crick complementarity is given by nature.
- Note that on the condition that the bases are complementary in nature the two strands bind – hence in the perspective of computation we can view it as *in vitro* the hybridization takes place on some condition *D* being satisfied.
  - this gives the notion of computing.
  - this also resembles the transition function of a Turing machine.
  - this can be exploited at least for *in vitro* experiments.

Infosys

Peptides and Antibodies

- Peptides – short proteins – sequence over 20 basic amino acids.
- Interactions between peptides and antibodies – Immune reactions.
- Antibodies recognize specific sequence in peptides – epitopes.
- Affinity power of antibodies presents an option to remove and attach antibodies – resembles a rewriting system

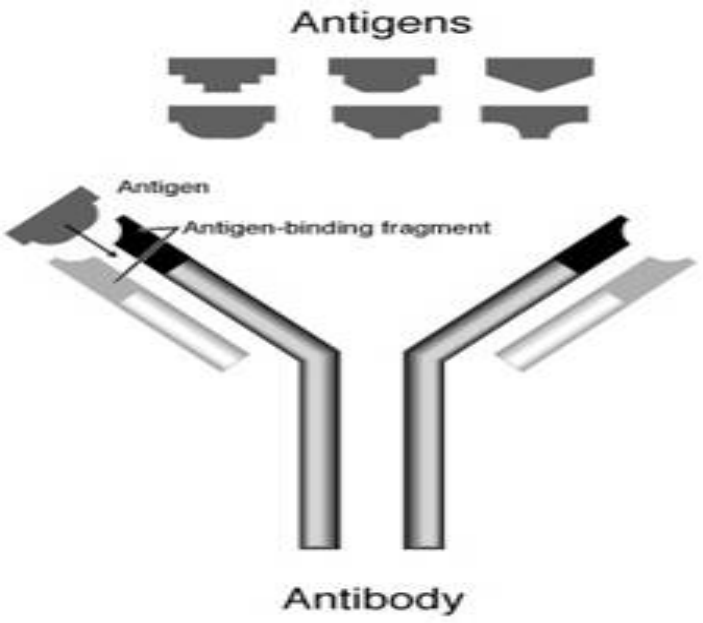
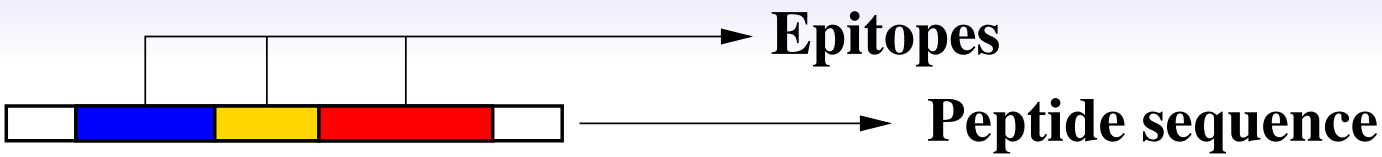
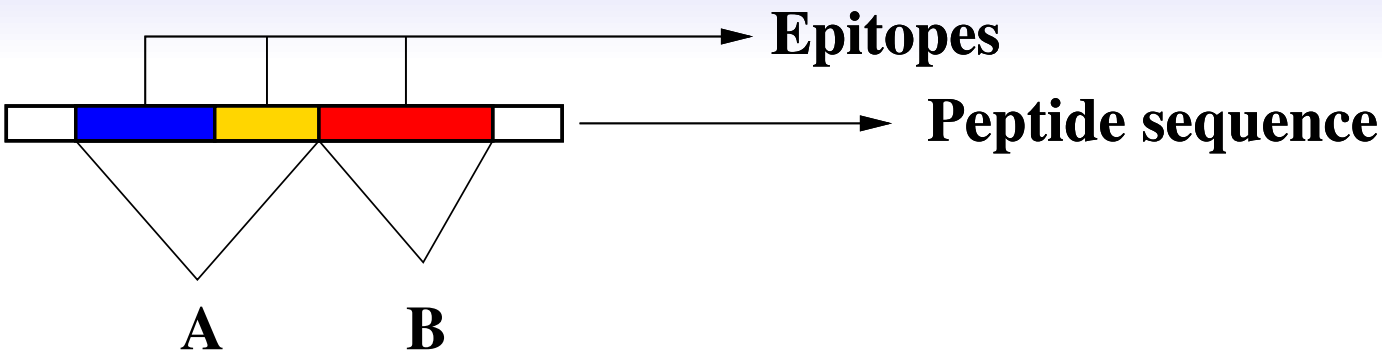
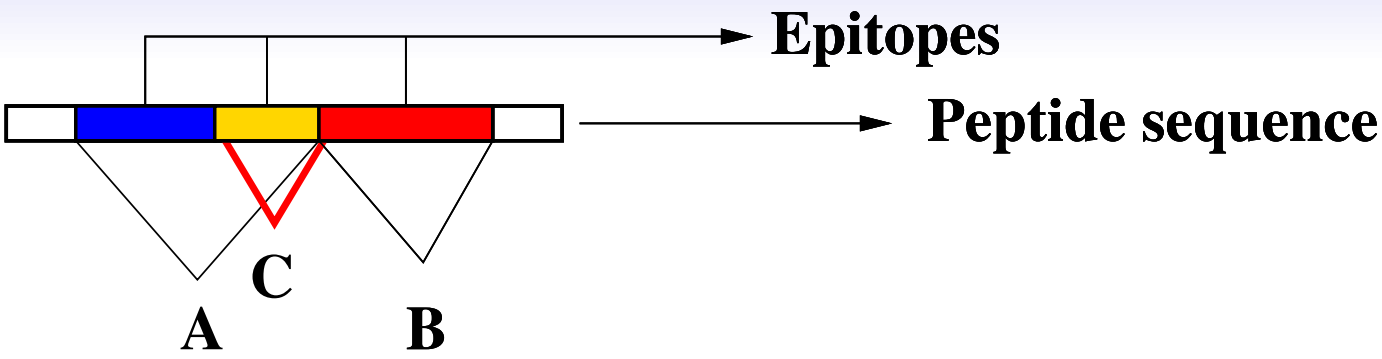


Figure: Antibody-antigen binding

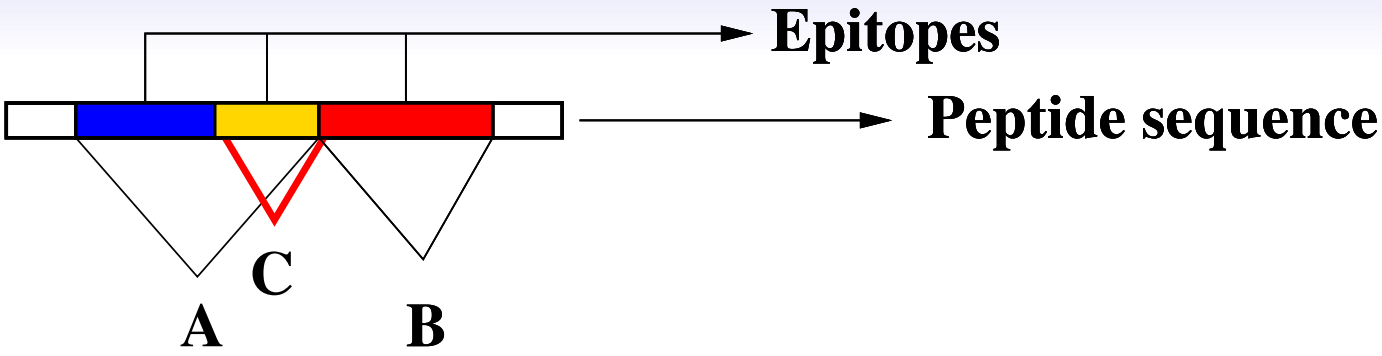

Epitopes
Peptide sequence



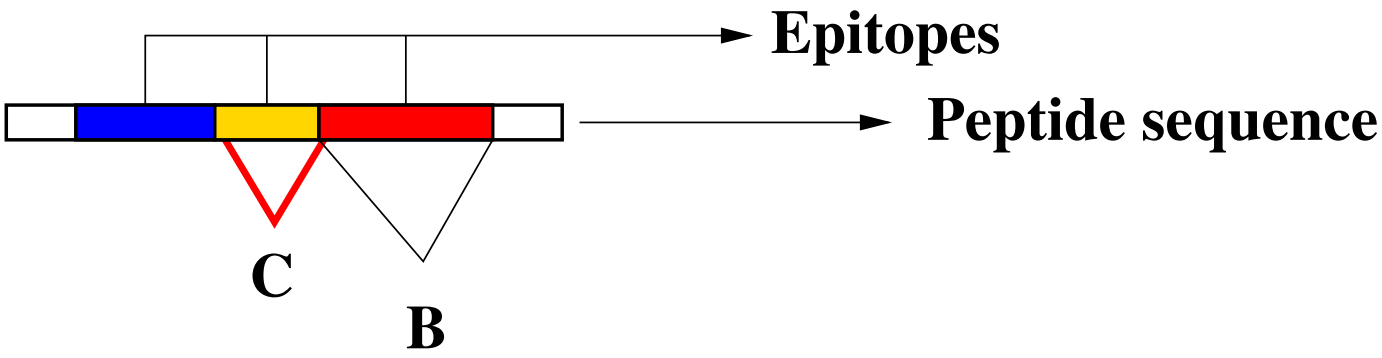
Peptide sequence with antibodies



Peptide sequence with antibodies



Peptide sequence with antibodies



Peptide sequence with antibodies

Peptides and Antibodies

- Epitopes for different antibodies or same antibody can overlap.
- There is a power called affinity associated with the binding of antibodies to its epitopes.
- One antibody can have more than one epitopes to bind with.
- There can be many antibodies that bind to a single or overlapping epitopes.

Infosys®

Bio-Computing

- Interactions between molecules as a computing model.
  - DNA hybridization,
  - DNA splicing,
  - Binding of antibodies to epitopes and so on.
- Massively parallel and non-deterministic.
  - Multiple copies of molecules – multiset.
- Has the potential to solve hard problems easily.
  - Brute force in a massively parallel way.
- Energy efficient.

Infosys

Contents

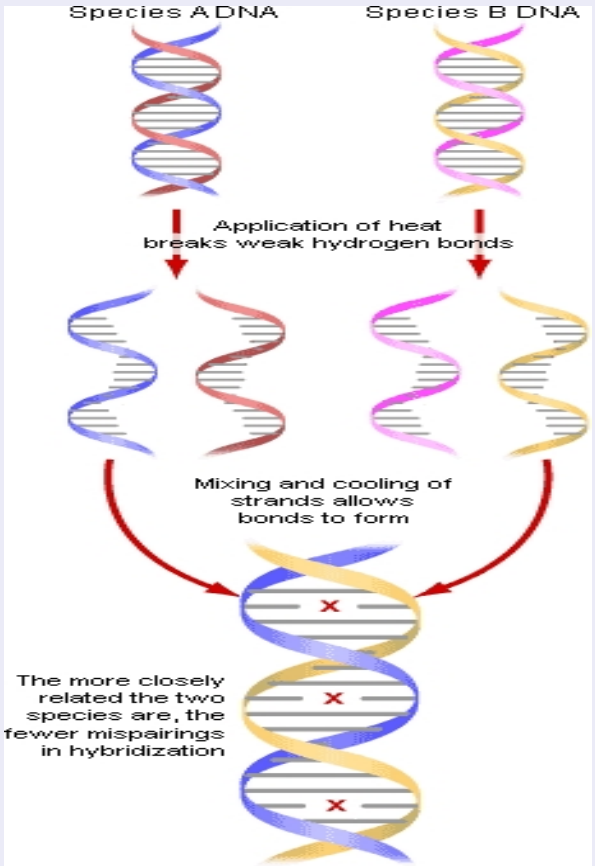
Infosys

DNA Computing

- Uses DNA strands and the interactions between strands as operations.
- Interactions are DNA hybridization, splicing and so on.
- Note that we will be having multiple copies of each strands so many things happen at the same time.
- Hence it is massively parallel and highly non-deterministic.

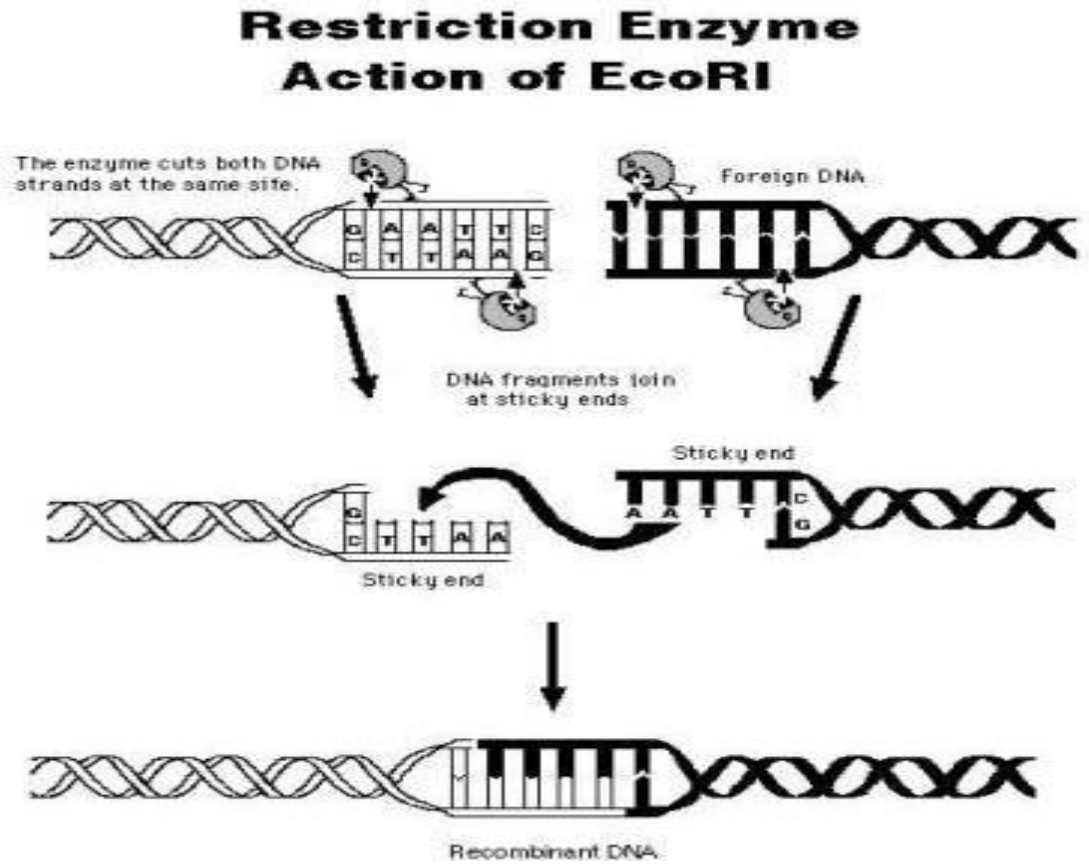
Infosys

DNA hybridization



- DNA splicing resembles a rewriting process.
- Cutting of two DNA strands at specific sites and making fragments of DNA.
- If the sticky ends of the fragments of DNA are complementary in nature then they recombine to form new DNA strands.
- Note that there will be several copies of DNA strands floating around – Parallel and non-deterministic.

Recombinant DNA



Tom Head's work

- Tom Head in 1987 gave a mathematical model for the recombinant behavior of DNAs by defining Splicing systems.
  - His work combined Formal language theory and Molecular biology.
  - He treated a DNA strand as a linear string or word over 4 alphabet $[A/T]$, $[C/G]$, $[G/C]$ and $[T/A]$.
  - His definition of splicing system resembles the recombinant behavior of DNA.
  - His study was aimed at the analysis of the generative capacity of these type of systems.

Infosys

Splicing system

- Splicing system is also called as H-system.
- The generative capacity of H-system is studied with respect to the Chomskian hierarchy of languages.
- A splicing operation over two words is defined as follows:

  Let the splicing rule be given by $(u_1 \# u_2 \$ u_3 \# u_4)$ where u_is are strings over a finite alphabet.

  The result of splicing x and y are z and w if and only if $x = x_1 u_1 u_2 x_2, y = y_1 u_3 u_4 y_2$ and $z = x_1 u_1 u_4 y_2, w = y_1 u_3 u_2 x_2$.
- The splicing system or H-system is a generative system that uses this splicing operation as a basic tool.

H scheme

For an H scheme $\sigma = (V, R)$ and a language $L \subseteq V^*$ we define

$$\sigma_1^0(L) = L,$$

$$\sigma_1^{i+1} = \sigma_1^i(L) \cup \sigma_1(\sigma_1^i(L)), i \geq 0,$$

and

$$\sigma_1^*(L) = \bigcup_{i \geq 0} \sigma_1^i(L).$$

H Scheme

For two families of languages FL_1 and FL_2 we define

$$H(FL_1, FL_2) = \{\sigma_1^*(L) \mid L \in F \in FL_1, \sigma = (V, R) \text{ with } R \in FL_2\}.$$

For an H scheme $\sigma = (V, R)$ where R is finite, we define the radius of σ as

$$radius(\sigma) = max(|x| \mid x = u_i, 1 \leq i \leq 4, \text{ for } u_1 \# u_2 \$ u_3 \# u_4 \in R\}.$$

If the radius of H scheme is bounded by a positive number p then it is denoted by $H(FL, [p])$.

Other models

- Sticker systems
- Self-assembly systems
- Insertion-deletion systems
- Watson-Crick Automata

Infosys

Contents

Infosys

Peptide Computing

- Proposed by H. Hug and R. Schuler [Hug,Schuler 2001].
- Solve some difficult combinatorial problems.
  - Satisfiability problem.
  - Hamiltonian path problem [M.S. Balan et al 2002].
- Universally complete [M.S. Balan et al 2002].

Infosys

Formal Model for Peptide Computing

- Capabilities and limitations of this computing paradigm
- Understand how immune system computes

Peptide Computer

Quintuple: $\mathcal{P} = (X, E, A, \alpha, \beta)$

- X is a finite alphabet;
- $E \subseteq X^+$ is a language;
- A is a countable alphabet with $A \cap X^* = \emptyset$;
- $\alpha \subseteq E \times A$ is a relation;
- $\beta : E \times A \to \mathbb{R}_+$ is a mapping such that $\beta(e, a) > 0$ if and only if $(e, a) \in \alpha$.

Infosys

Peptide Computer

Quintuple: $\mathcal{P} = (X, E, A, \alpha, \beta)$

- X is a finite alphabet;
- $E \subseteq X^+$ is a language;
- A is a countable alphabet with $A \cap X^* = \emptyset$;
- $\alpha \subseteq E \times A$ is a relation;
- $\beta : E \times A \to \mathbb{R}_+$ is a mapping such that $\beta(e, a) > 0$ if and only if $(e, a) \in \alpha$.

Infosys

Peptide Computer

Quintuple: $\mathcal{P} = (X, E, A, \alpha, \beta)$

- X is a finite alphabet;
- $E \subseteq X^+$ is a language;
- A is a countable alphabet with $A \cap X^* = \emptyset$;
- $\alpha \subseteq E \times A$ is a relation;
- $\beta : E \times A \to \mathbb{R}_+$ is a mapping such that $\beta(e, a) > 0$ if and only if $(e, a) \in \alpha$.

Infosys

Peptide Computer

Quintuple: $\mathcal{P} = (X, E, A, \alpha, \beta)$

- X is a finite alphabet;
- $E \subseteq X^+$ is a language;
- A is a countable alphabet with $A \cap X^* = \emptyset$;
- $\alpha \subseteq E \times A$ is a relation;
- $\beta : E \times A \to \mathbb{R}_+$ is a mapping such that $\beta(e, a) > 0$ if and only if $(e, a) \in \alpha$.

Infosys

Peptide Computer

Quintuple: $\mathcal{P} = (X, E, A, \alpha, \beta)$

- X is a finite alphabet;
- $E \subseteq X^+$ is a language;
- A is a countable alphabet with $A \cap X^* = \emptyset$;
- $\alpha \subseteq E \times A$ is a relation;
- $\beta : E \times A \to \mathbb{R}_+$ is a mapping such that $\beta(e, a) > 0$ if and only if $(e, a) \in \alpha$.

Infosys

Peptide Computer

- A-attachment: partial mapping τ from decomposition of $w \in X^*$ with respect to E to A. $z = w_\tau$.
- If affinity of a is more in z we say it dominates.
- Reaction between words and symbols – if a dominates (i, j) in z then multiset $R(z, a)$ is formed and $\tau \to \tau'$.
- Reaction between words – if a in z' dominates some position in z.

Infosys

About reactions

- Reactions occur when instability occurs:
  - a dominates (i, j) in z.
  - a in z' dominates (i, j) in z.
- One basic reaction can trigger a sequence of reactions.

Infosys

About reactions

- Reactions occur when instability occurs:
  - a dominates (i, j) in z.
  - a in z' dominates (i, j) in z.
- One basic reaction can trigger a sequence of reactions.

Infosys

About reactions

- Reactions occur when instability occurs:
  - a dominates (i, j) in z.
  - a in z' dominates (i, j) in z.
- One basic reaction can trigger a sequence of reactions.

Infosys

About reactions

- Reactions occur when instability occurs:
  - a dominates (i, j) in z.
  - a in z' dominates (i, j) in z.
- One basic reaction can trigger a sequence of reactions.

Infosys

Definitions

- *Peptide configuration* is a finite multiset of words in $(X \cup \alpha)^+ \cup A$.
- Peptide configuration P is said to be *stable* if $R(P) = \{P\}$.
- *Peptide instruction* has the form $+P$ or $-P$ where P is a peptide configuration.
- *Peptide program* is the one which controls the instruction set and the halting function.
- *Peptide computation* is a sequence of transition of stable configurations from $c_0, c_1 \cdots c_i$ (with respect to the peptide program) where $\chi(c_i) = 1$ for the first time.
- A function f is peptide computable if we proper encoding and decoding together with a peptide program to carry out the computation.

Infosys

Definitions

- *Peptide configuration* is a finite multiset of words in $(X \cup \alpha)^+ \cup A$.
- Peptide configuration P is said to be *stable* if $R(P) = \{P\}$.
- *Peptide instruction* has the form $+P$ or $-P$ where P is a peptide configuration.
- *Peptide program* is the one which controls the instruction set and the halting function.
- *Peptide computation* is a sequence of transition of stable configurations from $c_0, c_1 \cdots c_i$ (with respect to the peptide program) where $\chi(c_i) = 1$ for the first time.
- A function f is peptide computable if we proper encoding and decoding together with a peptide program to carry out the computation.

Infosys

Definitions

- *Peptide configuration* is a finite multiset of words in $(X \cup \alpha)^+ \cup A$.
- Peptide configuration P is said to be *stable* if $R(P) = \{P\}$.
- *Peptide instruction* has the form $+P$ or $-P$ where P is a peptide configuration.
- *Peptide program* is the one which controls the instruction set and the halting function.
- *Peptide computation* is a sequence of transition of stable configurations from $c_0, c_1 \cdots c_i$ (with respect to the peptide program) where $\chi(c_i) = 1$ for the first time.
- A function f is peptide computable if we proper encoding and decoding together with a peptide program to carry out the computation.

Infosys®

Definitions

- *Peptide configuration* is a finite multiset of words in $(X \cup \alpha)^+ \cup A$.
- Peptide configuration P is said to be *stable* if $R(P) = \{P\}$.
- *Peptide instruction* has the form $+P$ or $-P$ where P is a peptide configuration.
- *Peptide program* is the one which controls the instruction set and the halting function.
- *Peptide computation* is a sequence of transition of stable configurations from $c_0, c_1 \cdots c_i$ (with respect to the peptide program) where $\chi(c_i) = 1$ for the first time.
- A function f is peptide computable if we proper encoding and decoding together with a peptide program to carry out the computation.

Infosys®

Definitions

- *Peptide configuration* is a finite multiset of words in $(X \cup \alpha)^+ \cup A$.
- Peptide configuration P is said to be *stable* if $R(P) = \{P\}$.
- *Peptide instruction* has the form $+P$ or $-P$ where P is a peptide configuration.
- *Peptide program* is the one which controls the instruction set and the halting function.
- *Peptide computation* is a sequence of transition of stable configurations from $c_0, c_1 \cdots c_i$ (with respect to the peptide program) where $\chi(c_i) = 1$ for the first time.
- A function f is peptide computable if we proper encoding and decoding together with a peptide program to carry out the computation.

Infosys

Definitions

- *Peptide configuration* is a finite multiset of words in $(X \cup \alpha)^+ \cup A$.
- Peptide configuration P is said to be *stable* if $R(P) = \{P\}$.
- *Peptide instruction* has the form $+P$ or $-P$ where P is a peptide configuration.
- *Peptide program* is the one which controls the instruction set and the halting function.
- *Peptide computation* is a sequence of transition of stable configurations from $c_0, c_1 \cdots c_i$ (with respect to the peptide program) where $\chi(c_i) = 1$ for the first time.
- A function f is peptide computable if we proper encoding and decoding together with a peptide program to carry out the computation.

Infosys

Other Models

Automaton Models inspired by Peptide Computing

- Binding-Blocking Automata
- String Binding-Blocking Automata
- Rewriting Binding-Blocking Automata

Remarks

- Unwanted DNA hybridization:
  - Theory of codes is being used to design DNA strands.
- Cross-reactivity in peptide-antibody interactions:
  - Theory of codes can be used.
- Three dimensional structure of proteins.
- Automation issues.

Infosys®

References

- L. Adleman: Molecular Computation of Solutions To Combinatorial Problem, Science, 266: 1021-1024, (Nov. 11) 1994.
- H. Hug, R. Schuler: Strategies for the development of a peptide computer. Bioinformatics 17 (2001), 364–368.
- J. Hartmanis. On the weight of computations. Bulletin of the European Association for Theoretical Computer Science, 55:136–138, 1995.
- T. Head, X. Chen, M. Yamamura, and S. Gal. Aqueous computing: A survey with an invitation to participate. Journal of computer science and technology, 17(6):672–681, 2002.

Infosys®

References

- M.S. Balan, H. Jürgensen, On the Universality of Peptide Computing, Natural Computing, 7(1): 71-94 (2008).
- M.S. Balan, H. Jürgensen, Peptide Computing: Universality and Theoretical Model, Unconventional Computation, LNCS 4135, pp. 57–71, 2006.
- M.S. Balan, K. Krithivasan and Y. Sivasubramanyam, Peptide Computing: Universality and Complexity, In N. Jonoska and N. Seeman, editors, Proceedings of Seventh International Conference on DNA based Computers, LNCS 2340 pp. 290-299, 2002.
- G. Păun, G. Rozenberg, A. Salomaa: DNA Computing. New Computing Paradigms. Springer-Verlag, 1998.
- Y. Ishida: Immunity-Based Systems. Springer-Verlag, Berlin, 2004.

Infosys®