


## DNA 컴퓨팅



2000. 11. 18

한양대학교

이 상 구

Hanyang Univ., Korea

## DNA 계산의 기본 아이디어

- From Silicon to Carbon
- From Microchip to DNA 분자
  - 생체 분자가 갖는 정보처리 능력이 디지털 계산기에 있어서의 기본 스위칭 소자
- Quantum computing (양자 계산)
- DNA computing (분자 계산)
- 큰 기대를 할 수 있는 배경
  - DNA 분자가 갖는 큰 병렬성 (massive parallelism)
  - Watson-Crick 相補性 (Complementarity)

Hanyang Univ., Korea

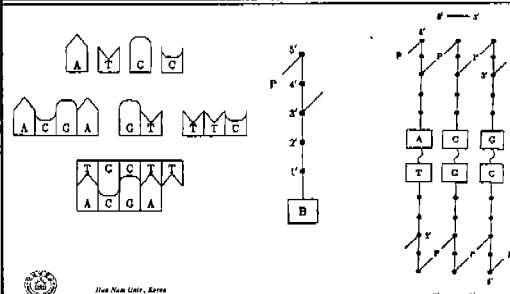
## DNA 염기

- DNA → DNA chain이라 불리는 polymer
  - Deoxyribonucleic acid(데옥시리보핵산)
  - Sugar(당), phosphate group(인산), nitrogenous base(질소 염기)
  - 이 chain은 nucleotides로 구성되어 있고, 각 nucleotide는 염기(bases)의 부분만 차이가 난다.
- 염기
 

{	A (Ademne)
	G (Guanine)
	C (Cytosine)
	T (Thymine)

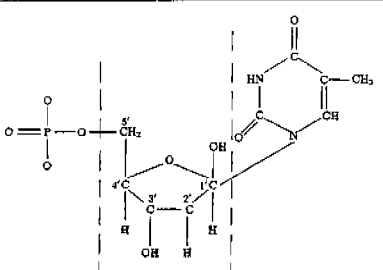
Hanyang Univ., Korea

## DNA 배열



Hanyang Univ., Korea

## Thymine 염기를 갖는 Nucleotide



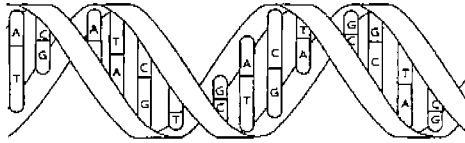
Hanyang Univ., Korea

## DNA의 기능

- DNA → Deoxyribonucleotides로 불리는 monomer deoxyribonucleic acid가 연결된 polymer 이다.
- DNA의 중요한 2가지의 기능
  - ① 단백질질을 생성하기 위한 부호화의 기능
  - ② 자손의 cell을 계승하기 위한 자기복제의 기능
- Nucleotide의 차이는 각각의 염기의 차이뿐이고 이 차이는 2개의 종류, 즉 purines와 pyrimidines으로 분류된다.
  - Purines는 A와 G가 있다.
  - Pyrimidines은 C와 T가 있다.

Hanyang Univ., Korea

## 이중 나선 구조



- DNA 분자는 2중의 기호열상의 구조
- 유전자 공학 - DNA를 조작하는 능력



Han Nam Univ., Korea

6

## DNA 컴퓨팅에서 사용되는 연산자

### • 1) 하이브리드 형성 (Hybridization)

- 각 염기들이 상보적인 염기들과 서로 결합을 하는 것을 의미함
  - 2개의 단일 가닥 사슬이 결합하여 1개의 이중 가닥 사슬을 형성



Han Nam Univ., Korea

7

## DNA 컴퓨팅에서 사용되는 연산자

### • 2) 결합법 (Ligation)

- 정착말단(sticky end)이 서로 결합하여 하나의 긴 사슬을 형성하는 것을 의미함



Han Nam Univ., Korea

8

## DNA 컴퓨팅에서 사용되는 연산자

### • 3) 서냉복원법 (Annealing)

- 생물학적인 의미는 조금씩 다르지만 DNA 컴퓨팅에서는 결합법 또는 하이브리드 형성법과 거의 동일한 의미로 사용됨



Han Nam Univ., Korea

9

## DNA 컴퓨팅에서 사용되는 연산자

### • 4) 중합효소 연쇄반응 (Polymerase Chain Reaction, PCR)

- 특정 염기배열(sequence)을 가지고 있는 DNA를  $n(n$ : 반복 횟수)배만큼 증가시키는 방법
  - 이중 가닥을 단일 가닥으로 분해(melting)한 후, 증폭시키고자 하는 염기배열의 상보 배열을 추가하여 이중 가닥으로 다시 만들



Han Nam Univ., Korea

10

## DNA 컴퓨팅에서 사용되는 연산자

### • 5) 겔 전기영동법 (Gel Electrophoresis)

- 특정 길이의 DNA를 추출하는 기법
  - DNA는 극성을 가지고 있고 염기 배열의 길이에 따라 서로 무게가 다름
  - DNA 배열의 무게에 따라 서로 분리가 되고 원하는 길이의 DNA 염기배열을 추출 가능



Han Nam Univ., Korea

11

## DNA 컴퓨팅에서 사용되는 연산자

- 6) 항체 친화력 반응(Antibody Affinity)
  - 전체 염기배열 중에서 특정 염기배열을 가지고 있는 DNA를 선택하는 방법
    - 항체 반응을 이용해 특정 염기 배열을 구문



Han Yong Univ. Korea

12

## DNA 컴퓨팅에서 사용되는 연산자

- 7) 효소 (Enzyme) 반응
  - 라이기즈(ligase)
    - 결합법에서 사용
    - 주로 사용되는 효소
  - 제한 효소(restriction enzyme)
    - 특정한 부분을 인식하여 이중 가닥 사슬을 포개서 (cleave) 단일 가닥 사슬로 만든 후, 다른 과정에 적용할 수 있도록 함



Han Yong Univ. Korea

13

## DNA 컴퓨팅 응용 분야

- NP-complete 문제의 해결
- Turing machine 구현
- Finite state machine 구현
- Boolean circuit 구현 (DNA Adder 등)
- DNA associative memory
- DES 암호 해독



Han Yong Univ. Korea

14

## Adleman의 실험

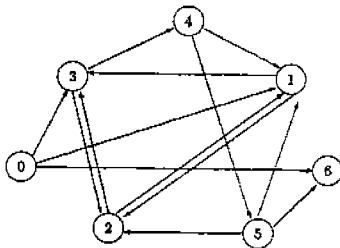
- DNA 계산의 궁극적인 영향은 아직 보이지 않음
- Feynman, "Miniaturization" 초미세한 컴퓨터를 만들 수 있는 가능성 제시
- Quantum computing, DNA computing



Han Yong Univ. Korea

15

## HPP(Hamiltonian path problem)



Han Yong Univ. Korea

16

## NP-complete problem, intractable

- Adleman - non-deterministic algorithm
  - Input: vertex의 수  $n$ 의 directed graph  $G$ 로 지정된  $V_{in}$ 과  $V_{out}$ 이 가능하다.
  - Step 1: 대량으로 자원을 이용하여  $G$ 에 대한 path를 random으로 생성한다.
  - Step 2:  $V_{in}$ 으로 시작하지 않거나,  $V_{out}$ 으로 끝나지 않는 path를 삭제한다.
  - Step 3:  $n$ 개의 vertex를 포함하지 않는 것을 삭제한다.
  - Step 4:  $n$ 개의 vertex  $v$ 에 대해  $v$ 를 포함하지 않는 path를 삭제한다.
  - Output: path가 남아 있으면 "Yes", 아니면 "No"




Han Yong Univ. Korea

17

## 자료 구조

- $S_i$  ( $0 \leq i \leq 6$ ), 각  $S_i$ 에 20-mer DNA 분자  
 $S_0 = \text{TATCGGATCGGTATATCCGA}$   
 $S_1 = \text{GCTATTCGAGCTTAAAGCTA}$   
 $S_2 = \text{GGCTAGGTACCAGCATGCTT}$
- Complementary mapping  $h$  (Watson-Crick morphism)  
 $h(A) = T, h(T) = A, h(C) = G, h(G) = C$   
 $h(\text{CATTAG}) = \text{GTAATC}$   
 $h(S_2) = \text{ATAGCCTAGCCATATAGGCT}$   
 $h(S_3) = \text{CGATAAGCTCGAATTTTCGAT}$




18

## continued

$S_i = S_i S_i'$


- vertex  $i$  와 vertex  $j$  사이의 edge는  $h(S_i', S_j)$ 로 부호화  
 $e_{2 \rightarrow 1} = \text{CATATAGGCTCGATAAGCTC}$   
 $e_{1 \rightarrow 2} = \text{GAATTCGATATAGCCTAGC}$   
 $e_{1 \rightarrow 4} = \text{GAATTCGATCCGATCCATG}$
- ligation
- polymerase chain reaction(PCR)



19

## DNA의 연산


- merge : 주어진 시험관  $N_1$ 과  $N_2$ 에 대해  $N_1 \cup N_2$
- amplify : 주어진 시험관  $N$ 에 대해 2개의 copy를 만들
- detect :  $N$ 이 적어도 1개의 DNA 분자를 포함하면 "true", 아니면 "false"
- separate 또는 extract :  $N$ 과  $\{A, C, G, T\}$ 상의 기호열  $w$ 에 대해서 2개의 시험관  $+(N, w), -(N, w)$ 를 만들  
 $+(N, w)$  :  $w$ 를 부분기호열로서 포함하는  $N$ 의 모든 기호열(분자)  
 $-(N, w)$  :  $w$ 를 부분기호열로서 포함하지 않는  $N$ 의 모든 기호열(분자)



20

## continued


- length-separate : 주어진 시험관  $N$ 과 기호열  $w$ 에 대해서  $N$ 에 대한 길이  $n$  이하의 모든 배열분자를 포함하는 시험관  $(N, \leq n)$ 을 생성
- position-separate :  $N$ 에 관한 모든 배열분자에서  
 -  $w$ 로 시작하는 것을 모두 포함하는 시험관  $B(N, w)$ 를 생성  
 -  $w$ 로 끝나는 것을 모두 포함하는 시험관  $E(N, w)$ 를 생성



21

## example


- 주어진 시험관이 A와 G의 양쪽을 포함하는가?  
 (1)  $input(N)$   
 (2)  $N \leftarrow +(N, A)$   
 (3)  $N \leftarrow +(N, G)$   
 (4)  $detect(N)$
- 적어도 한 개의 A 또는 G를 포함하는 모든 배열분자  
 (1)  $input(N)$   
 (2)  $amplify(N)$ 를 시행하여 새로운 2개의 copy  $N_1$ 과  $N_2$ 를 생성  
 (3)  $N_A \leftarrow +(N_1, A)$   
 (4)  $N_G \leftarrow +(N_2, G)$   
 (5)  $N' \leftarrow -(N \cup A)$   
 (6)  $merge(N_A \cup N'_G)$



22

## Adleman의 실험

- $input(N)$
- $N \leftarrow B(N, s_0)$
- $N \leftarrow E(N, s_1)$
- $N \leftarrow (N, \leq 140)$
- for  $i = 1$  to 5 do begin  $N \leftarrow +(N, s_i)$  end
- $detect(N)$



23

## Linton의 Satisfiability problem

$$\alpha = (x_1 \vee \sim x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\sim x_1 \vee x_3) \wedge \sim x_3$$

• exhaustive search

example)  $\beta = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$

N에 관한 문자에서 i번째의 bit가 j(j=0,1)인 것으로 된 시험관을 S(N, i, j)라 하면

$$S(N, i, j) = +(N, a_j^i)$$

$$S^-(N, i, j) = -(N, a_j^i)$$



1998.10.14. 24

24

- |                                    |                    |
|------------------------------------|--------------------|
| (1) $input(N_0)$                   | (1) 00, 01, 10, 11 |
| (2) $N_1 \leftarrow B(N_0, 1, 1)$  | (2) 10, 11         |
| (3) $N_1' \leftarrow E(N_0, 1, 1)$ | (3) 00, 01         |
| (4) $N_2 \leftarrow (N_1', 2, 1)$  | (4) 01             |
| (5) $merge(N_1, N_2) = N_3$        | (5) 10, 11, 01     |
| (6) $N_4 \leftarrow B(N_3, 1, 0)$  | (6) 01             |
| (7) $N_4' \leftarrow E(N_3, 1, 0)$ | (7) 10, 11         |
| (8) $N_5 \leftarrow (N_4', 2, 0)$  | (8) 10             |
| (9) $merge(N_4, N_5) = N_6$        | (9) 01, 10         |
| (10) $detect(N_6)$                 |                    |



1998.10.14. 25

25