

Lecture-2 (Genome re-arrangement)

1. What are the synteny block?

- Synteny block are the region of chromosomes that have conserved gene order across different species or within similar species. These block indicates evolutionary relationship and can help identify genetic similarity and rearrangement such as fission, fusion, duplication and translocation.

Genome Dot plot

k-morse algorithm : match sub-string of length k.

1. 1 morse algorithm \rightarrow match 1 character.
2. 2 morse algorithm \rightarrow match 2 character.
3. 3 morse algorithm \rightarrow match 3 character.

Sequence 1: A G C T A G C T A (horizontally)

Sequence 2: G C T A G C T T A (vertically)

Using 1 morse

$s_2 \backslash s_1$	A	G	C	T	A	G	C	T	A
G									
C									
T									
A									
G									
C									
T									
T									
A									

Procedure \rightarrow
Sequence এর element
অনুযায়ী একটা একটা
element match করব
যেটার সাথে match
হবে (.) দিচ্ছি দেখানো।

Using 2 morse-code:-

Sequence 1: A G C T A G C T A

Sequence 2: G C T A G C T T A

$s_1 \backslash s_2$	A	G	C	T	A	G	C	T	A
G									
C									
T									
A									
G									
C									
T									
T									
A									

using 3 morse-code:-

$s_1 \backslash s_2$	A	G	C	T	A	G	C	T	A
G									
C									
T									
A									
G									
C									
T									
T									
A									

Finding reverse complement :-

Sequence 1 : T T A T C T A C

Sequence 2 : A G A T A A A C

52 \ 51	T	T	A	T	C	T	A	C
A								
G								
A								
T								
A								
A								
A								
C								

AG → GA → CT

GA → AG → TC

AT → TA → AT

TA → AT → TA

AA → AA → TT

AA → AA → TT

AC → CA → GT

Secondary Diagonal.

Sorting by reversal :-

Step 0 : +2 -4 -3 +5 -8 -7 -6 +1

Step 1 : +2 +3 +4 +5 -8 -7 -6 +1

Step 2 : +2 +3 +4 +5 +6 +7 +8 +1

Step 3 : -8 -7 -6 -5 -4 -3 -2 +1

Step 4 : -8 -7 -6 -5 -4 -3 -2 -1

Step 5 : 1 2 3 4 5 6 7 8

Adjacency and breakpoint.

+0 x +2 x -4 ✓ -3 x +5 x -8 ✓ -7 ✓ -6 x +1 x +9

adjacency (P) + Breakpoint (P) = 3+6

=9.

T < A < D < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

T < A < A < A

$(+, -) \rightarrow$ অরু-অরু
 $(-, -) \rightarrow$ শেষ-অরু
 $(-, +) \rightarrow$ শেষ-শেষ
 $(+, +) \rightarrow$ অরু-শেষ

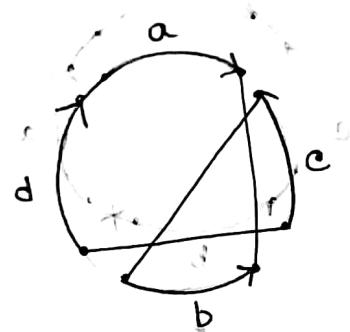
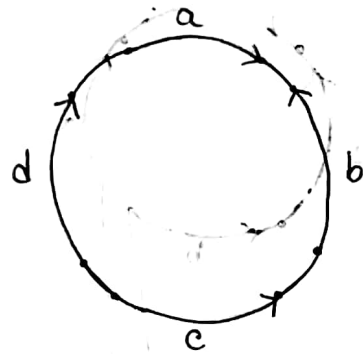
connection.

Genome Graph:-

Re-arrangement in tumor genom

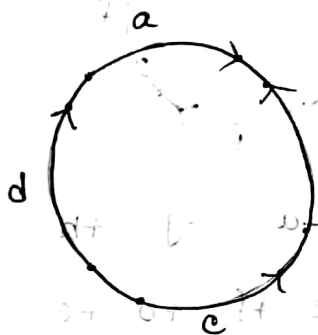
Genome Sequence: $+a -b -c +d$

Sequence: $+a -b -bc +d$

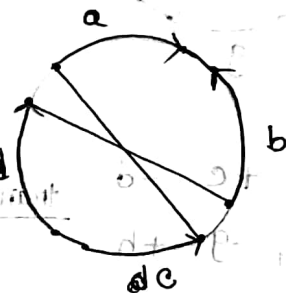


Genome e block operation:-

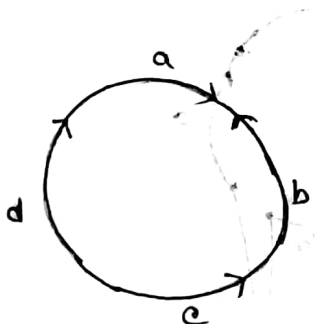
$+a -b -c +d \xrightarrow{P(3,4)} +a -b -d +c$



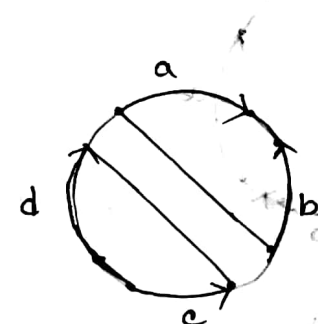
reversal



fission:- $+a -b -c +d \rightarrow (+a -b) (-c +d)$



fission
fusion

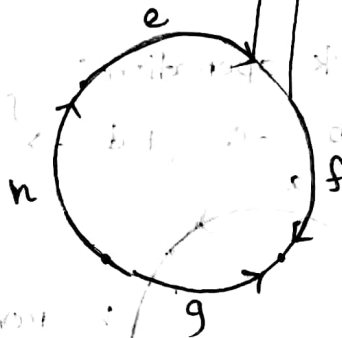
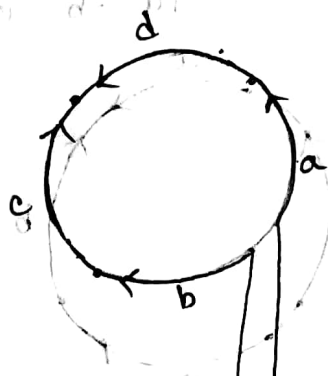
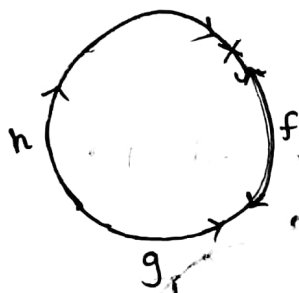
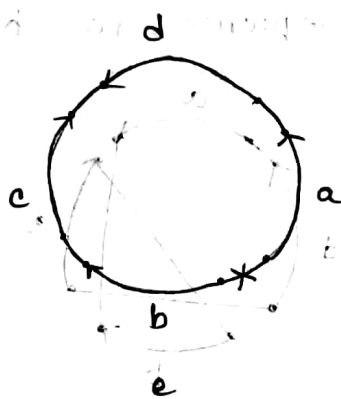


Translocation :-

$-a \rightarrow +b \quad +e \quad -d$
 $+e \rightarrow +f \quad -g \quad +h$

translocation →

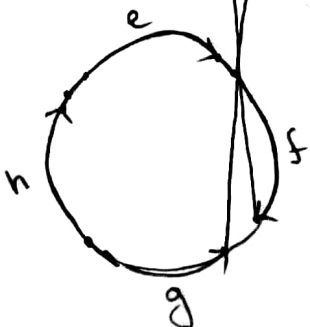
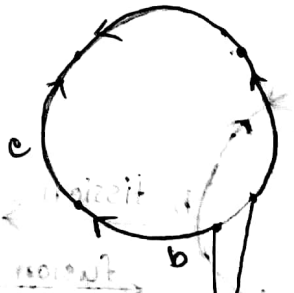
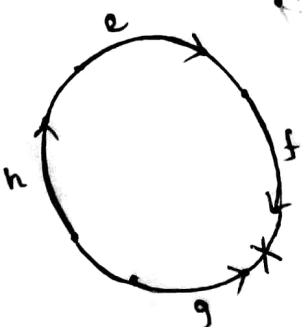
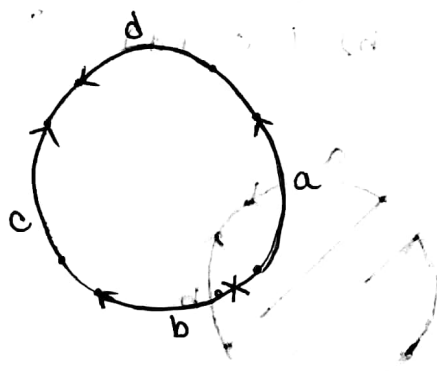
$-a \quad +f \quad -g \quad +h$
 $+e \quad +b \quad +c \quad +d$



$-a \rightarrow +b \quad +c \quad -d$
 $+e \quad +f \quad -g \quad +h$

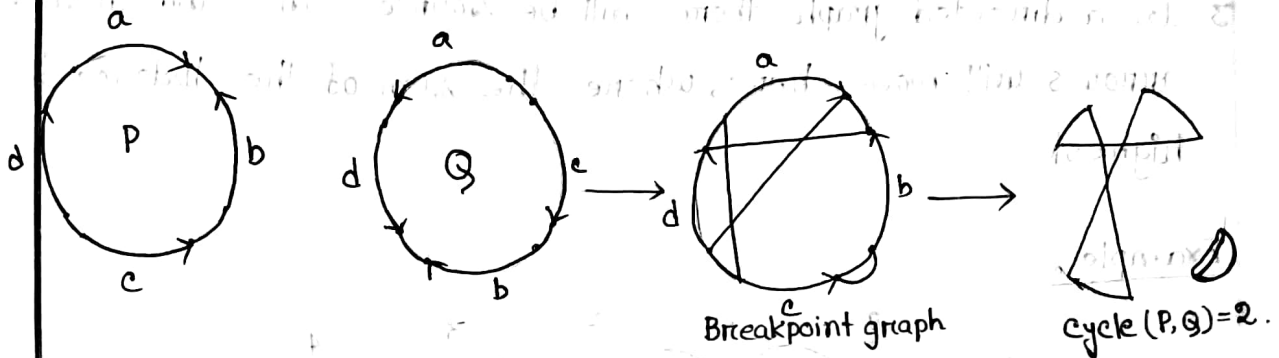
translocation →

$-a \quad -g \quad +h$
 $+e \quad +f \quad +b \quad +c \quad -d$

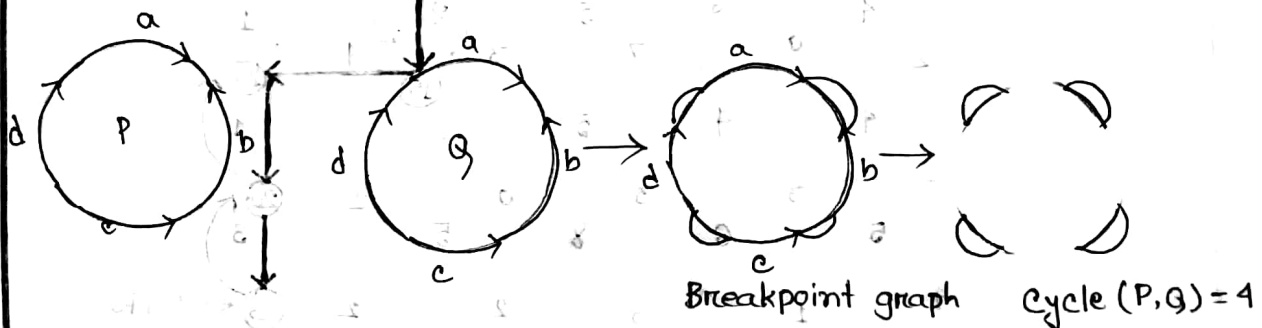


Break-point analysis graph :-

1. $P = +a -b -c +d$
 $Q = -a +c +b -d$



2. $P = +a -b -c +d$
 $Q = +a -b -c +d$



2 break distance :

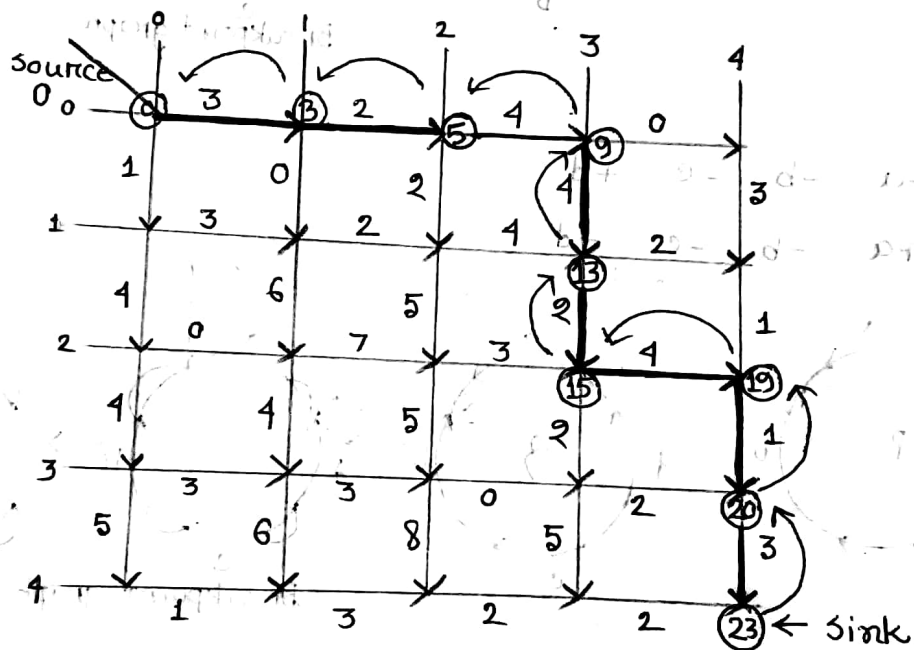
$$\begin{aligned} 2 \text{ break distance } (P, Q) &= \text{no of blocks} - \text{no of cycle } (P, Q) \\ &= 4 - 2 \\ &= 2. \end{aligned}$$

Lecture-3 (Dynamic programming)

Manhattan tourist problem

1. Seeking a path from source to sink.
2. arrows are free to move in down and right side.
3. In a directed graph, there will be source, sink and distance. arrow's will move there, where the sum of the distance is highest.

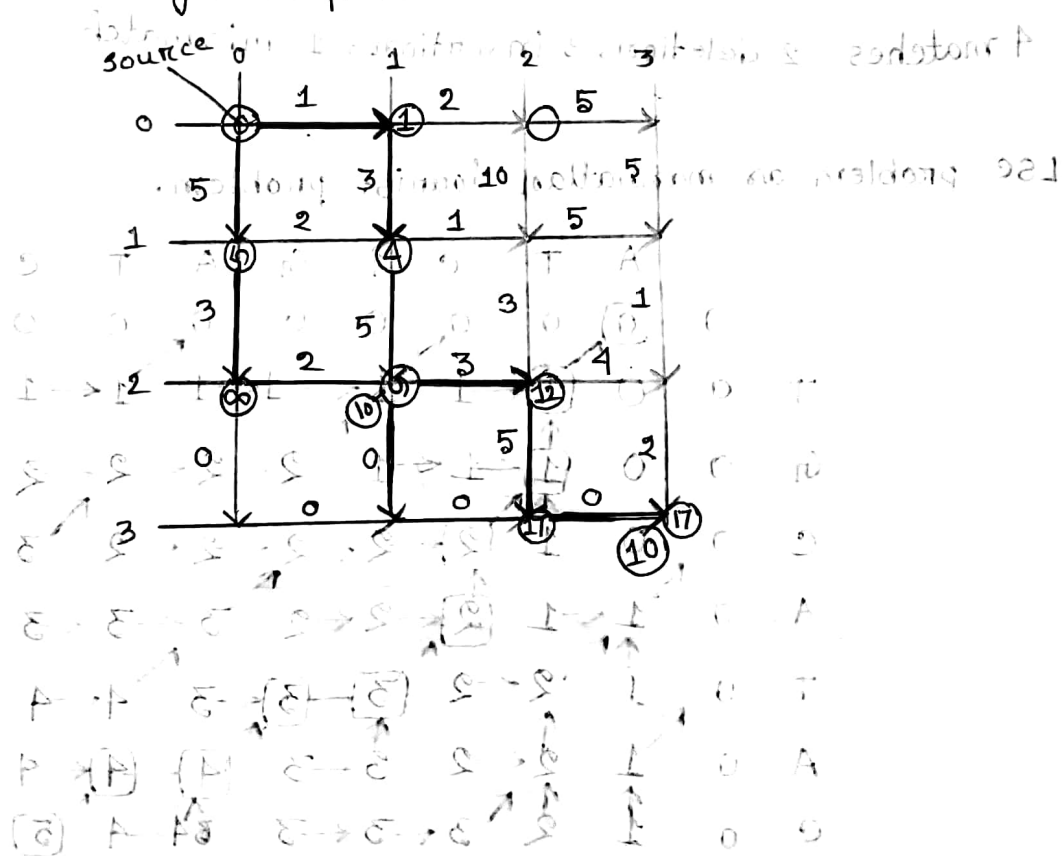
Example :



* Why greedy algorithm is not optimal?

- A greedy algorithm is not optimal because it makes locally optimal choices without considering the overall best path.

The greedy algorithm always chooses the highest-weighted street at each step. This approach can lead to dead-end or miss a better long term path.



$\nearrow \rightarrow \uparrow \rightarrow \nearrow \uparrow \nearrow \uparrow \nearrow$

9 - A - T A 9 0 F - = v

$$\mathcal{D} \cap A \cap T = \mathcal{D} \cap T \cap A = W$$

Alignment : 2 row representation

V = ATCTGATG

W = TGCATAC

A	T	-	C	-	T	G	A	T	G
-	T	G	C	A	T	-	A	-	C

4 matches, 2 deletion, 3 insertion, 1 mismatch.

LSC problem as manhattan tourist problem.

		A	T	C	T	G	A	T	C
	0	0	0	0	0	0	0	0	0
T	0	0	1	1	1	1	1	1	1
G	0	0	1	1	1	2	2	2	2
C	0	0	1	2	2	2	2	2	3
A	0	1	1	2	2	2	3	3	3
T	0	1	2	2	3	3	3	4	4
A	0	1	2	2	3	3	4	4	4
C	0	1	2	3	3	3	4	4	5

← ↑ ↑ ↑ ↑ ↓ ← ↑ ← ↑

V = - T G C A T - A - C

W = A T - C - T G A T C

Lecture-4 (Sequence alignment)

Scoring Alignment:

$$W = C^T A G C$$

V = C A G T C

$$-match = +2$$

`mismatch = -1`

$$\text{indel} = -3$$

$$S_{i,j} = \max \left\{ \begin{array}{ll} S_{i-1,j-1} + 2 & \text{if match} \\ S_{i-1,j-1} - \mu & \text{if mismatch} \\ S_{i-1,j} & - \sigma \\ S_{i,j-1} & - \sigma \end{array} \right\} \text{indel}$$

Global alignment

W

		C	T	A	G	C
-	0	-3	-6	-9	-12	-15
C	-3	-2	-1	-4	-7	-10
A	-6	-4	1	-7	-5	-8
G	-9	-7	-2	0	-3	0
T	-12	-10	-5	-3	0	2
C	-15	-10	-5	-3	-2	-1

alignment algorithm score = 2

$$W = C^T A G^{-1} C$$
$$V = C - A G_T C$$

$$\begin{aligned} \text{score} &= +2 - 3 + 2 + 2 - 3 + 2 \\ &= +2. \end{aligned}$$

Local Alignment

$$s_{i,j} = \max \begin{cases} s_{i-1,j-1} + 1 & \text{if } v_i = w_j \\ s_{i-1,j-1} - \mu & \text{if } v_i \neq w_j \\ s_{i-1,j} - \sigma & \\ s_{i,j-1} - \sigma & \\ 0 & \end{cases}$$

- match হলে +1
- mismatch হলে -1
- gap হিসাব Left, right
- negative মান আসলে 0 দ্বারা replace
- highest number, arrow থেকে w, v হিসাব।

Example : 1 $V = A T C G$
 $w = T C C$

match = +1

mismatch, $\mu = -1$

gap, $\sigma = -2$

-	-	A	T	C	G
-	0	0	0	0	0
T	0	-1 -2	-2 1	-1 0	-2 0
C	0	-1 -2	-2 0	-1 2	-2 0
G	0	-1 -2	-2 0	-2 1	-1 0

$V = A T C G$

$w = - T C C$

↑↑↑↑→↑

→ - 0 - A T C G

→ T C C A - - -

→ - - - - -

Example :2

Sequence 1: A A A C

Sequence 2: A A G

match = +1

mismatch = -1

Gap = -2

	-	A	A	A	C
-	0	0	0	0	0
A	0	1	1	1	0
A	0	1	2	2	0
G	0	0	0	1	1

Sequence 1: A A A C

Sequence 2: - A A G

Gap Penalties

1. Constant : Consider consecutive gap as one

T A C C T A G

T - - - T A -

$$\text{match} = +1, \text{gap/indel} = -1$$

$$\text{score} = +1 - 1 + 1 + 1 - 1 = +1$$

2. Linear : Consider all gap individually.

T A C C T A G

T - - - T A -

$$\text{match} = +1, \text{indel} = -1$$

$$\text{Score} = +1 - 1 - 1 - 1 + 1 + 1 - 1 = -1.$$

3. Affine :

T A C C T A G

T - - - T A -

$$\text{match} = +1, \text{indel} = -1, \text{gap opening} = -2, \text{gap extension} = -1$$

$$\text{Score} = +1 - 2 - 1 - 1 + 1 + 1 - 2$$

$$= -3$$

$$\text{gap penalty} = \rho + \sigma(k-1)$$

• gap opening
বসে একটি gap
এ যেটা প্রথম
সেটা হিসাব
করবে।

• gap extension
বসে একটি gap
থাকলে প্রথম
gap বাদ দিবে।