# Physical Mapping of DNA

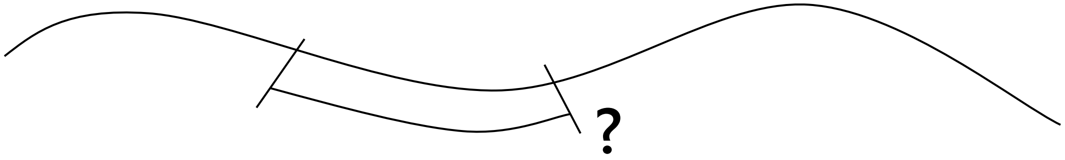
INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Dr. Jaspreet Kaur Dhanjal
Assistant Professor, Department for Computational Biology
Email ID: jaspreet@iiitd.ac.in

***September 19, 2025***

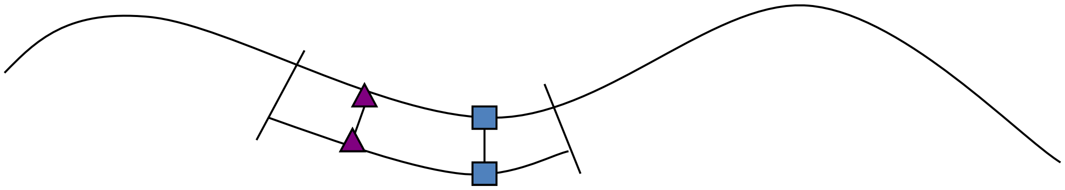
# Background

Given a sequence of DNA, how do we figure out where on some larger chromosome the sequence lies?



Look for markers that match in both the chromosome and the shorter sequence.

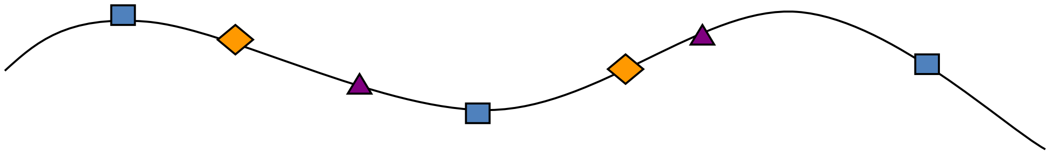
- Markers: Usually short, precisely defined sequences



Match!

# Creating the Physical Map

How do we create the original map?



Generate fingerprints (markers) with:

- Restriction site mapping
- Hybridization

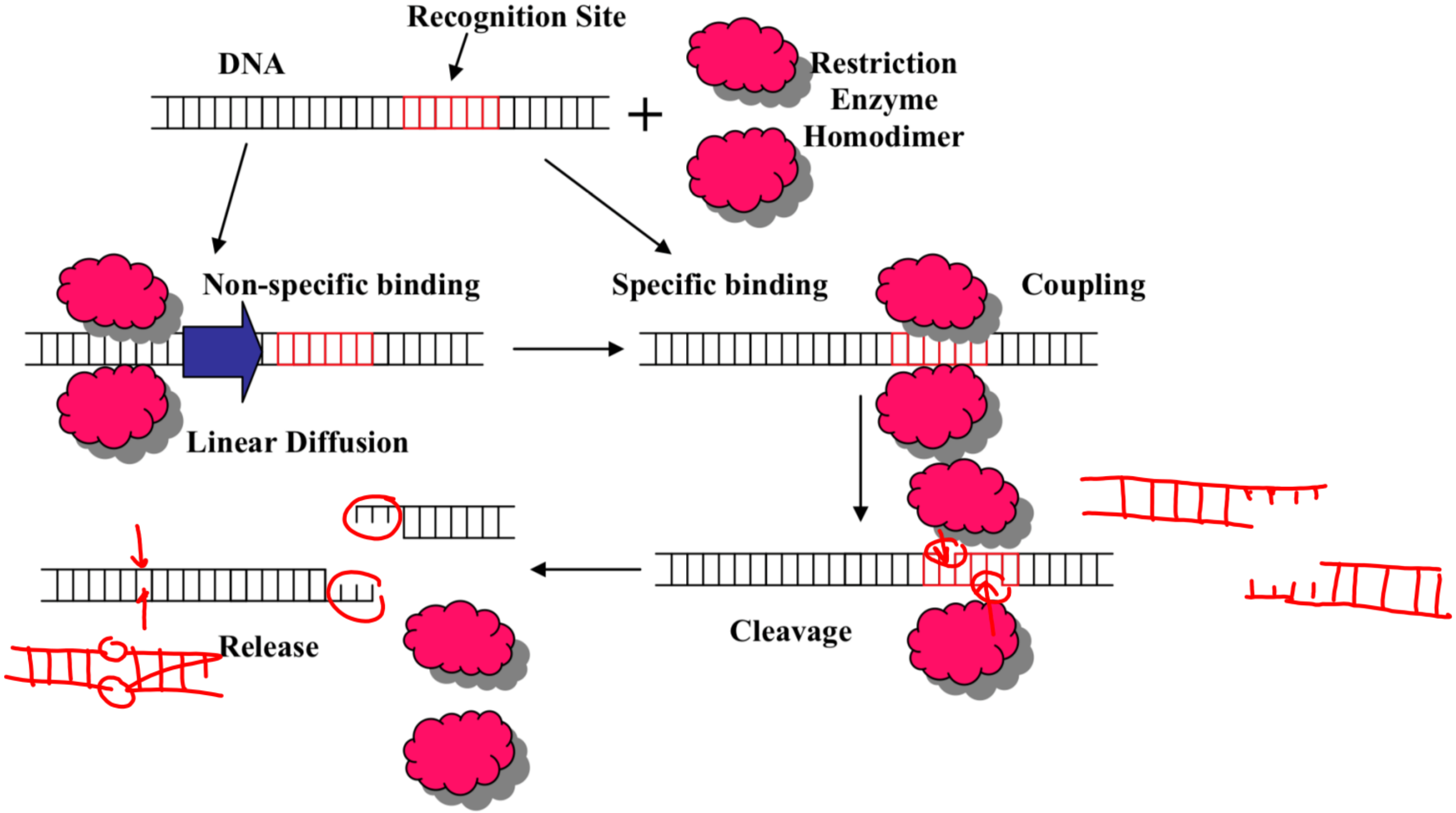
Can't we just expand the sequence assembly techniques we've already learned?
No!

Why not?

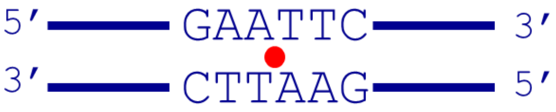
- A chromosome isn't just a few kilo bps long.
- Human chromosomes range in length from 51 million to 245 million base pairs.

# Restriction enzymes

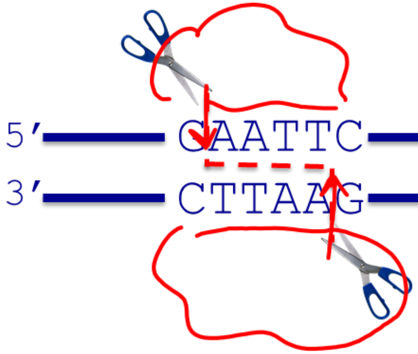
Molecular scissors that cut double stranded DNA molecules at specific points



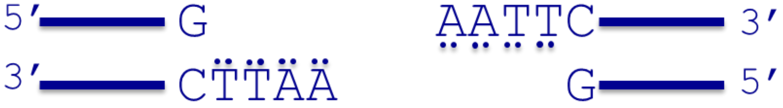
# Restriction enzymes



EcoRI recognition site is a palindrome with an axis of symmetry

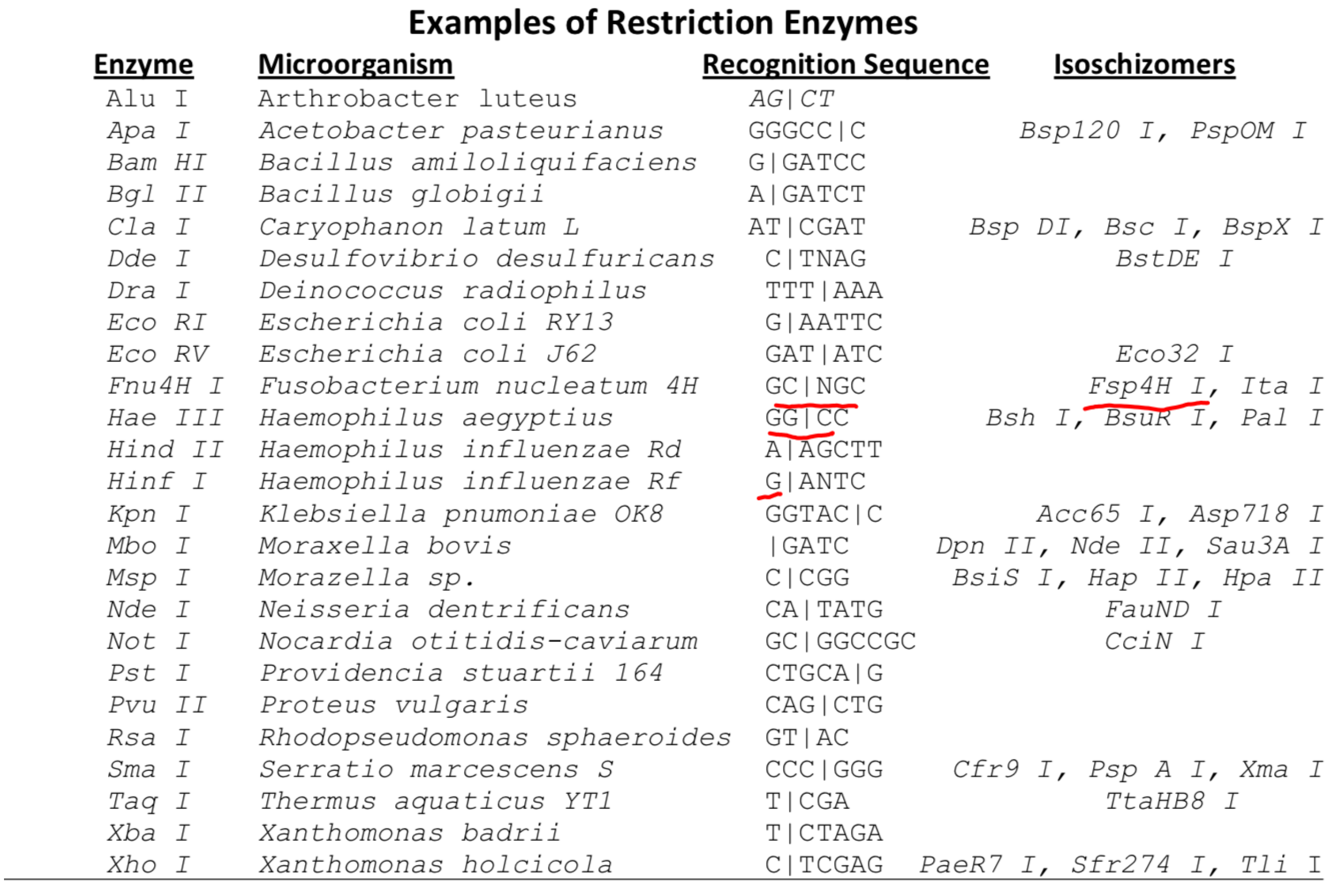


EcoRI dimer binds sequence and catalyzes double-strand cleavage



Products have “sticky ends” or overhanging bases.

# Restriction enzymes



### Examples of Restriction Enzymes

| Enzyme | Microorganism | Recognition Sequence | Isoschizomers |
|---|---|---|---|
| Alu I | Arthrobacter luteus | AG\|CT | |
| Apa I | Acetobacter pasteurianus | GGGCC\|C | Bsp120 I, PspOM I |
| Bam HI | Bacillus amiloliquifaciens | G\|GATCC | |
| Bgl II | Bacillus globigii | A\|GATCT | |
| Cla I | Caryophanon latum L | AT\|CGAT | Bsp DI, Bsc I, BspX I |
| Dde I | Desulfovibrio desulfuricans | C\|TNAG | BstDE I |
| Dra I | Deinococcus radiophilus | TTT\|AAA | |
| Eco RI | Escherichia coli RY13 | G\|AATTC | |
| Eco RV | Escherichia coli J62 | GAT\|ATC | Eco32 I |
| Fnu4H I | Fusobacterium nucleatum 4H | GC\|NGC | Fsp4H I, Ita I |
| Hae III | Haemophilus aegyptius | GG\|CC | Bsh I, BsuR I, Pal I |
| Hind II | Haemophilus influenzae Rd | A\|AGCTT | |
| Hinf I | Haemophilus influenzae Rf | G\|ANTC | |
| Kpn I | Klebsiella pnumoniae OK8 | GGTAC\|C | Acc65 I, Asp718 I |
| Mbo I | Moraxella bovis | \|GATC | Dpn II, Nde II, Sau3A I |
| Msp I | Morazella sp. | C\|CGG | BsiS I, Hap II, Hpa II |
| Nde I | Neisseria dentrificans | CA\|TATG | FauND I |
| Not I | Nocardia otitidis-caviarum | GC\|GGCCGC | CciN I |
| Pst I | Providencia stuartii 164 | CTGCA\|G | |
| Pvu II | Proteus vulgaris | CAG\|CTG | |
| Rsa I | Rhodopseudomonas sphaeroides | GT\|AC | |
| Sma I | Serratio marcescens S | CCC\|GGG | Cfr9 I, Psp A I, Xma I |
| Taq I | Thermus aquaticus YT1 | T\|CGA | TtaHB8 I |
| Xba I | Xanthomonas badrii | T\|CTAGA | |
| Xho I | Xanthomonas holcicola | C\|TCGAG | PaeR7 I, Sfr274 I, Tli I |

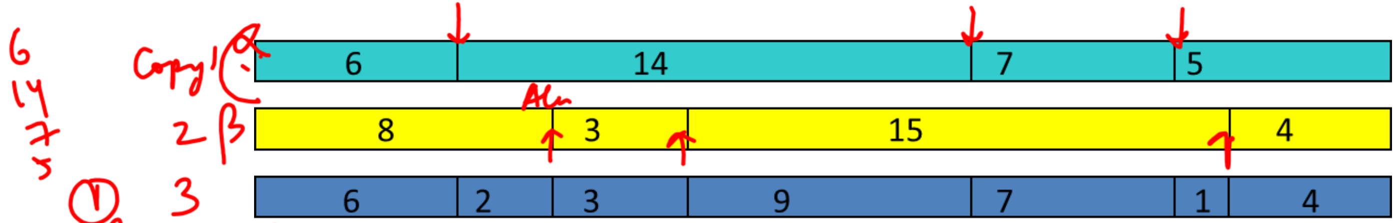
# Restriction Site Mapping

In this situation, the fingerprint is the length between restriction sites of given enzymes.

Make three copies of target DNA: strings A, B, C.

Apply one enzyme (α) to string A, another (β) to string B, and both (α and β) to string C.

Line up the fragments in A and B so they match C: This is the ***double digestion problem***.



α+β.

ATGAGCTTGGGCCGTGTCAGCTCCCCAGCTGTC

Alu 1 → AGCT

ATGAGCTTGGGCCGTGTCAGCTCCCCAGCTGTC

Alu I Apa I Alu I Alu I

Alu 1 : 5, 15, 8, 5

Apa 1: 12, 21

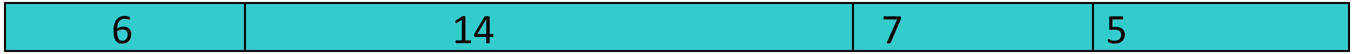
Both: 5, 7, 8, 8, 5

# Restriction Site Mapping

A variant is the ***partial digestion approach***:

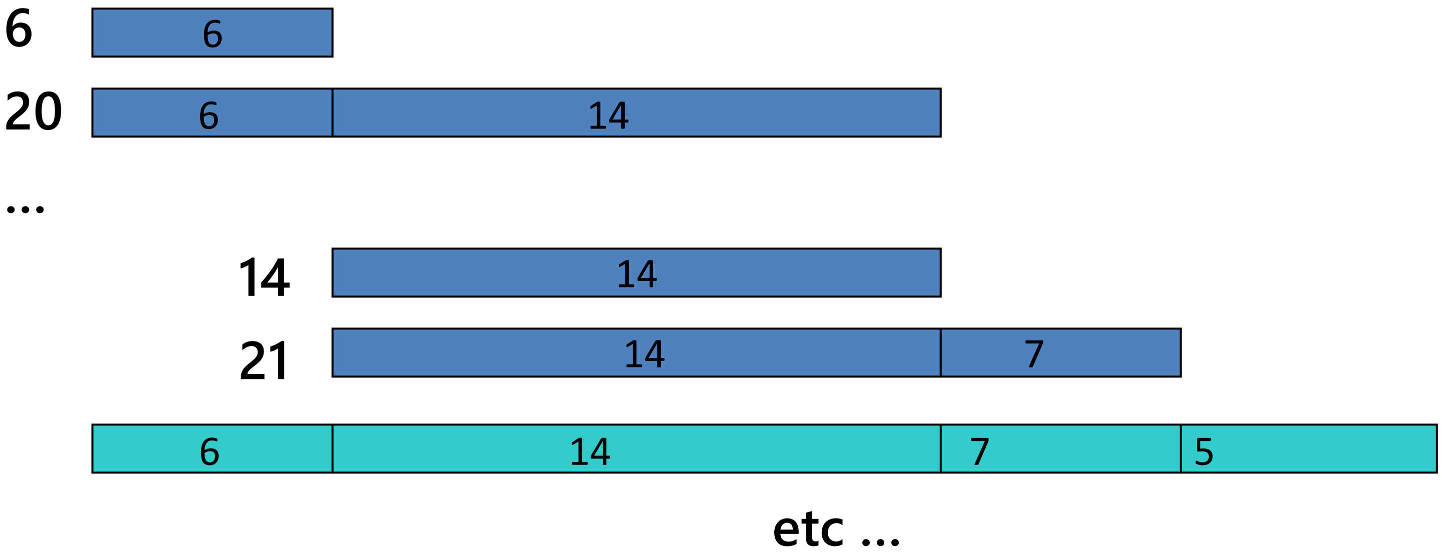
Use only one enzyme, but allow it to act for different time periods.

Different restriction sites will be recognized.



Fragment sites: 6, 20, 27, 32;
14, 21, 26;
7, 12;
and 5

# Restriction Site Mapping



Fragment sites: 6, 20, 27, 32; 14, 21, 26; 7, 12; and 5

# Restriction site model

Back to the double digest problem:

Enzyme A: 1000, 2100, 1400, 500

Enzyme B: 1200, 2500, 1300

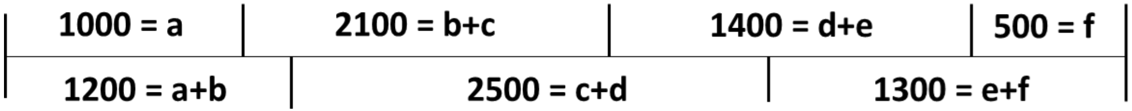
Enzyme A+B: 1000, 200, 1900, 600, 800, 500

Find permutations of A and B such that there is a one-to-one correspondence between all the subintervals and C.

| | | | | | | |
|---|---|---|---|---|---|---|
| A+B: | 1000 | 200 | 1900 | 600 | 800 | 500 |
| | a | b | c | d | e | f |

A: 1000= a, 2100= b+c, 1400= d+e, 500=f

B: 1200= a+b, 2500=c+d, 1300= e+f



1000 = a

2100 = b+c

1400 = d+e

500 = f

1200 = a+b

2500 = c+d

1300 = e+f

# Restriction site models

***Limitations***:

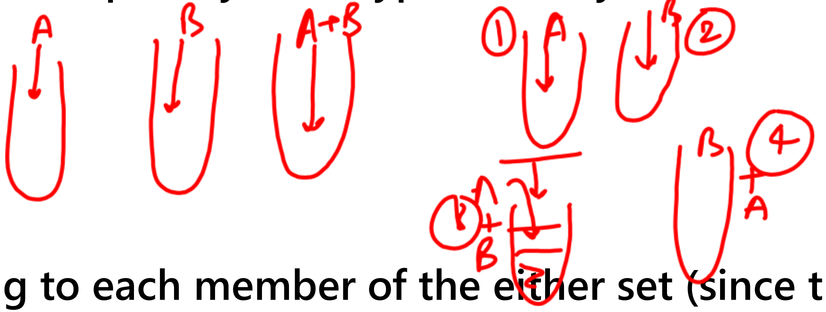
- This double digestion problem is NP-complete.
- Between 2 sites cut by A, there are three sites – b1, b2, b3 cut by enzyme B. It will be difficult to tell the order of the fragments [b1,b2] and [b2,b3].
- The number of solutions is k! for k = number of restriction sites.

# Enhanced Double Digestion Problem

- The Enhanced Double Digest (EDD) problem is NP-hard in the general case, but if the lengths of fragments in C (the string acted upon by both types of enzymes) are distinct, it can be solved in linear time!
- We have the multisets A and B.

  A = {6, 14, 7, 5}

  B = {8, 3, 15, 4}



- Take the actual fragments corresponding to each member of the either set (since the sets are only lengths). Apply the other enzyme to the fragment (i.e. apply enzyme β to fragments from A, and vice versa) to create subfragments.
- $AB_i$ is the multiset of subfragments created by applying enzyme β to fragments from A;
- $BA_j$ from applying enzyme α to fragments of B.

# Enhanced Double Digestion Problem

Example:

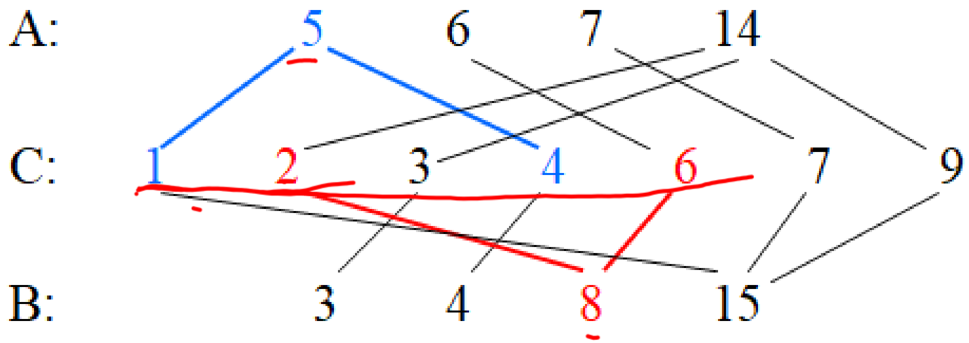
A={5, 6, 7, 14}

B={3, 4, 8, 15}

$AB_1$={1,4}, $AB_2$={6}, $AB_3$={7}, $AB_4$={2,3,9}

$BA_1$={3}, $BA_2$={4}, $BA_3$={2,6}, $BA_4$={1,7,9}

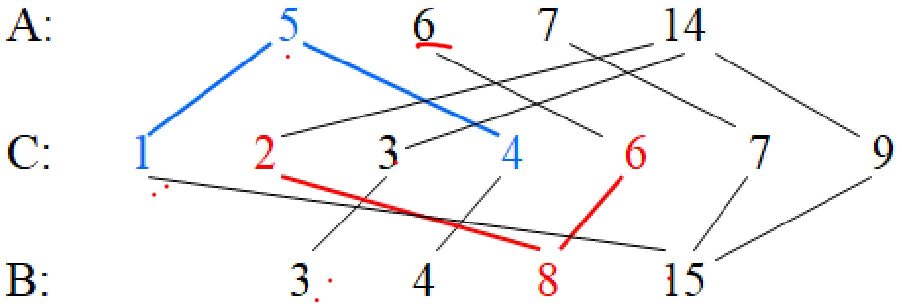
Given A, B, $AB_i$ and $BA_j$ for all i, j, construct an undirected graph that connects each element of A and B to its corresponding AB/BA. Note that all elements in C will be covered

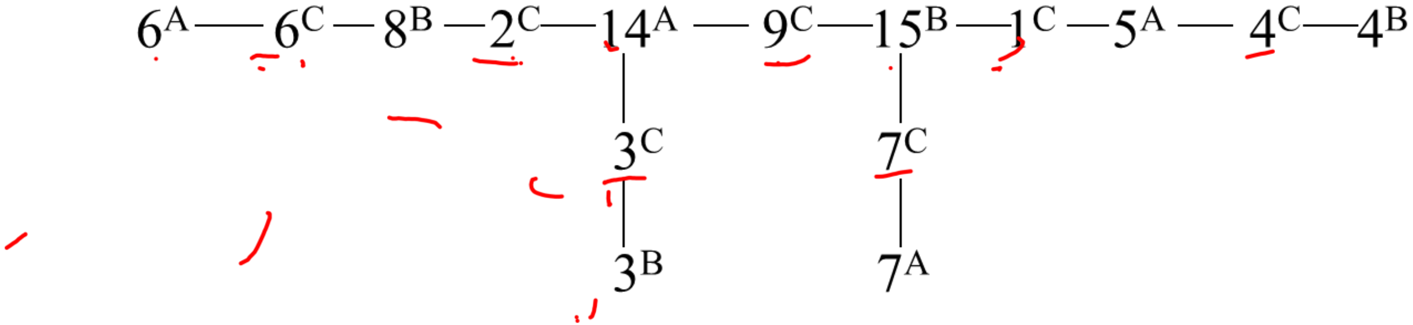


# Enhanced Double Digestion Problem

**Create a spanning tree:**

**Starting at random node, follow all paths from the node without repeating the edges.**

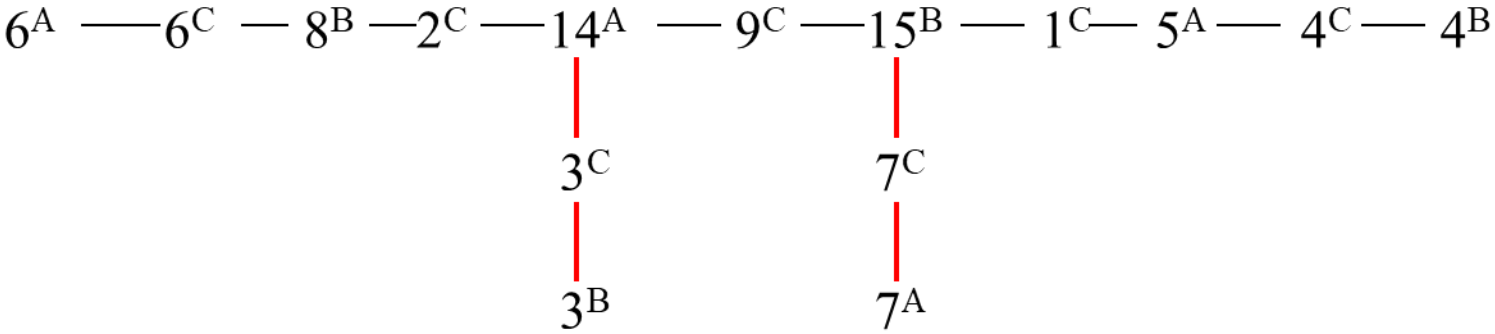




# Enhanced Double Digestion Problem

- The graph (G) will always be connected, and every node in A and B will only be adjacent to nodes from C. Each node from C connects to only one node each from A and B.

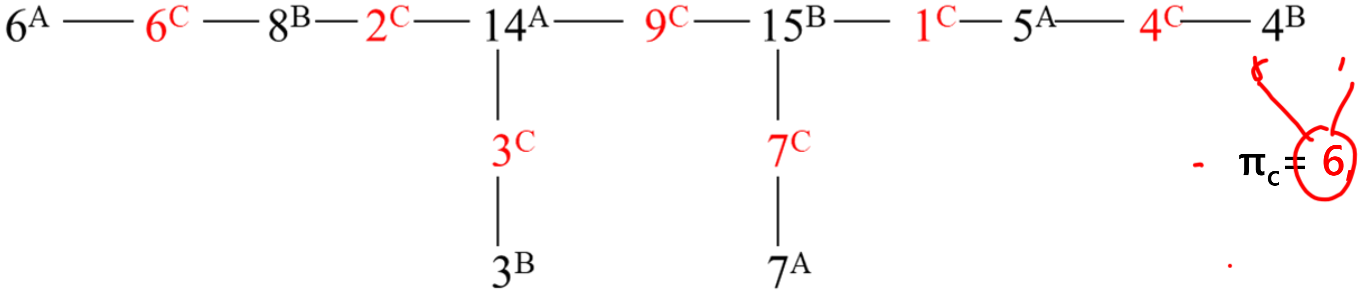
- If the problem can be solved: G will be a spanning tree, and any subtree that “hangs” on the longest path will be a 2-node length path (dangler).



# Enhanced Double Digestion Problem

- If the graph G is not a spanning tree, and not all subtrees hanged off the longest path are danglers, then there is no valid permutation.
- Perform Dangler-first search on the graph G…

- Traverse G starting at one end of a path S with the largest number of edges, reading only the nodes from C. Whenever reaching a node with degree greater than 2 (must have a dangler), read the nodes in C from the hanging danglers first, then continue to traverse S. This sequence is πc.

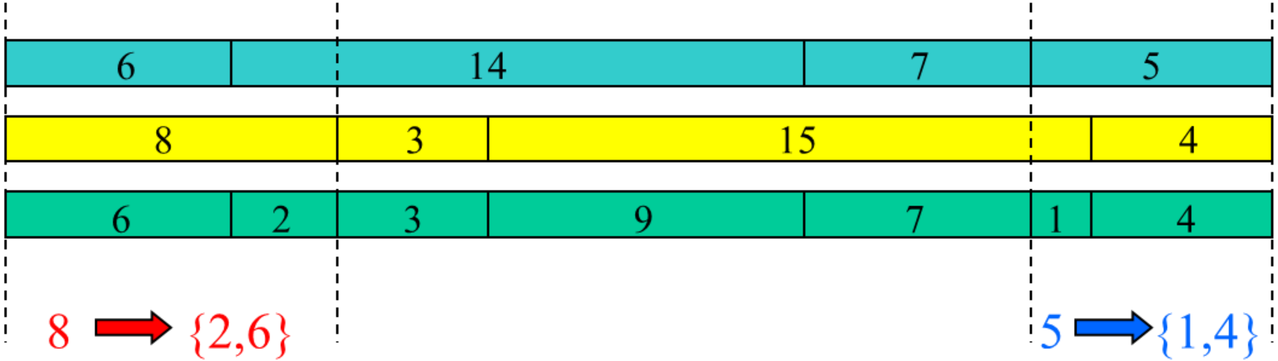


- $\pi_c = 6, 2, 3, 9, 7, 1, 4$

# Enhanced Double Digestion Problem

The elements in each $AB_i$ form a consecutive subsequence in $\pi_c$. Likewise, the elements in each $BA_j$ also form a consecutive subsequence in $\pi_c$.

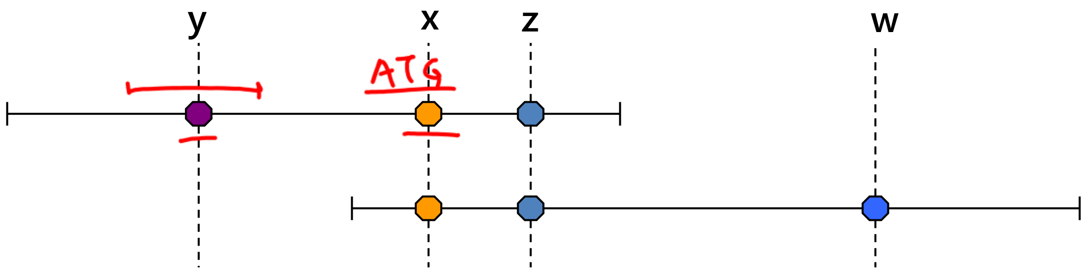
This permutation is a valid permutation… meaning: we have the answer!



# Hybridization Mapping

- Check whether specific small sequences (called probes) bind (hybridize) to fragments (clones)
- The fingerprint is the subset of probes that successfully hybridize to the clone.
- If some portion of one clone's fingerprint matches another, they are likely to be from overlapping regions of the target.

- Probes x, y, z, bound to clone A; x, w and z bound to clone B… overlap in x and z.



# Hybridization mapping model

- Consecutive Ones Property Model (C1P)
- This can be solved in linear time!
- Assumptions:
  - The probes are unique.
  - There are no errors.
  - All of the correspondences of clones and probes have been found.

# Hybridization mapping model

- Build a matrix (n x m), n = number of clones, m = number of probes.
- Entry i,j is a binary code for whether probe j hybridized to clone i.

Probes →

Clones →

| | | | | |
|---|---|---|---|---|
| F1 | 1 | 1 | 0 | 1 |
| F2 | 0 | 1 | 0 | 1 |
| F3 | 1 | 0 | 1 | 1 |

4 markers

Here, probe 1 hybridized to clone 1, probe 2 hybridized to clone 1, probe 1 hybridized to clone 3, probe 4 hybridized to clone 3.

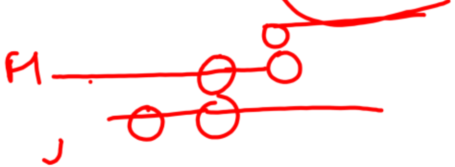
# Hybridization mapping model

- Find a permutation of the columns (probes) such that all the 1s in each row (clone) are consecutive.

| | Probes | | | |
|---|---|---|---|---|
| Clones | 1 | 0 | 0 | 1 |
| | 0 | 1 | 0 | 1 |
| | 1 | 0 | 1 | 0 |

→

| | Probes | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| Clones | 0 | 1 | 1 | 0 |
| | 0 | 0 | 1 | 1 |
| | 1 | 1 | 0 | 0 |



# Hybridization mapping model

- This algorithm can be run in linear time!
- Unfortunately, the assumption that there are no errors isn’t useful because biology isn’t a mathematical model. Probes may not bind; DNA may be replicated incorrectly.