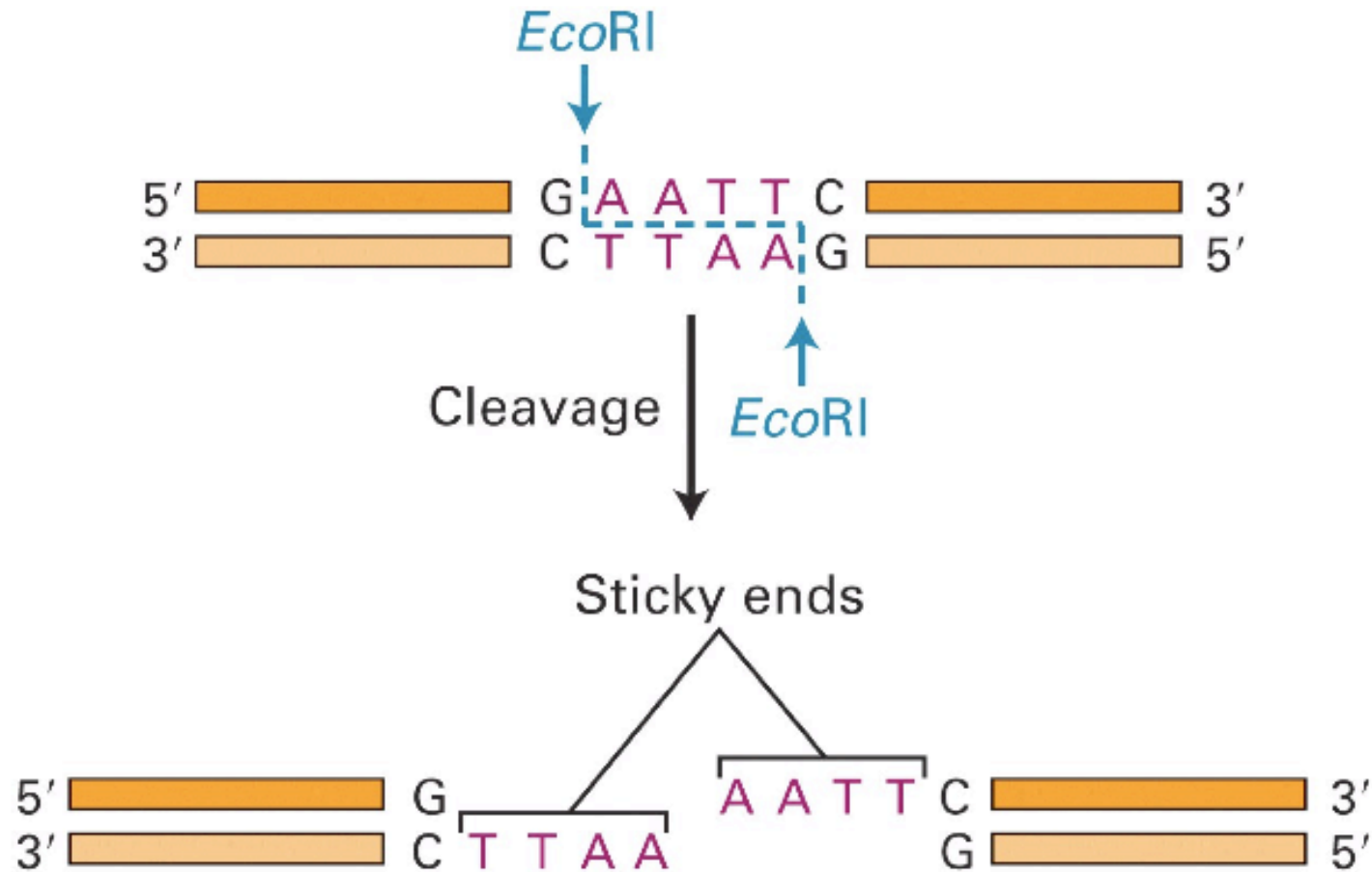

DNA Mapping and Brute Force Algorithms

Outline

- Restriction Enzymes
 - Gel Electrophoresis
 - Partial Digest Problem
 - Brute Force Algorithm for Partial Digest Problem
 - Branch and Bound Algorithm for Partial Digest Problem
 - Double Digest Problem
-

Molecular Scissors



Discovering Restriction Enzymes

- *HindII* - first restriction enzyme – was discovered accidentally in 1970 while studying how the bacterium *Haemophilus influenzae* takes up DNA from the virus
- Recognizes and cuts DNA at sequences:
 - GTGCAC
 - GTTAAC

Discovering Restriction Enzymes



Werner Arber **Daniel Nathans** **Hamilton Smith**

- Werner Arber** – discovered restriction enzymes
- Daniel Nathans** - pioneered the application of restriction for the construction of genetic maps
- Hamilton Smith** - showed that restriction enzyme cuts DNA in the middle of a specific sequence

My father has discovered a servant who serves as a pair of scissors. If a foreign king invades a bacterium, this servant can cut him in small fragments, but he does not do any harm to his own king. Clever people use the servant with the scissors to find out the secrets of the kings. For this reason my father received the Nobel Prize for the discovery of the servant with the scissors".

Daniel Nathans' daughter
(from Nobel lecture)

Recognition Sites of Restriction Enzymes

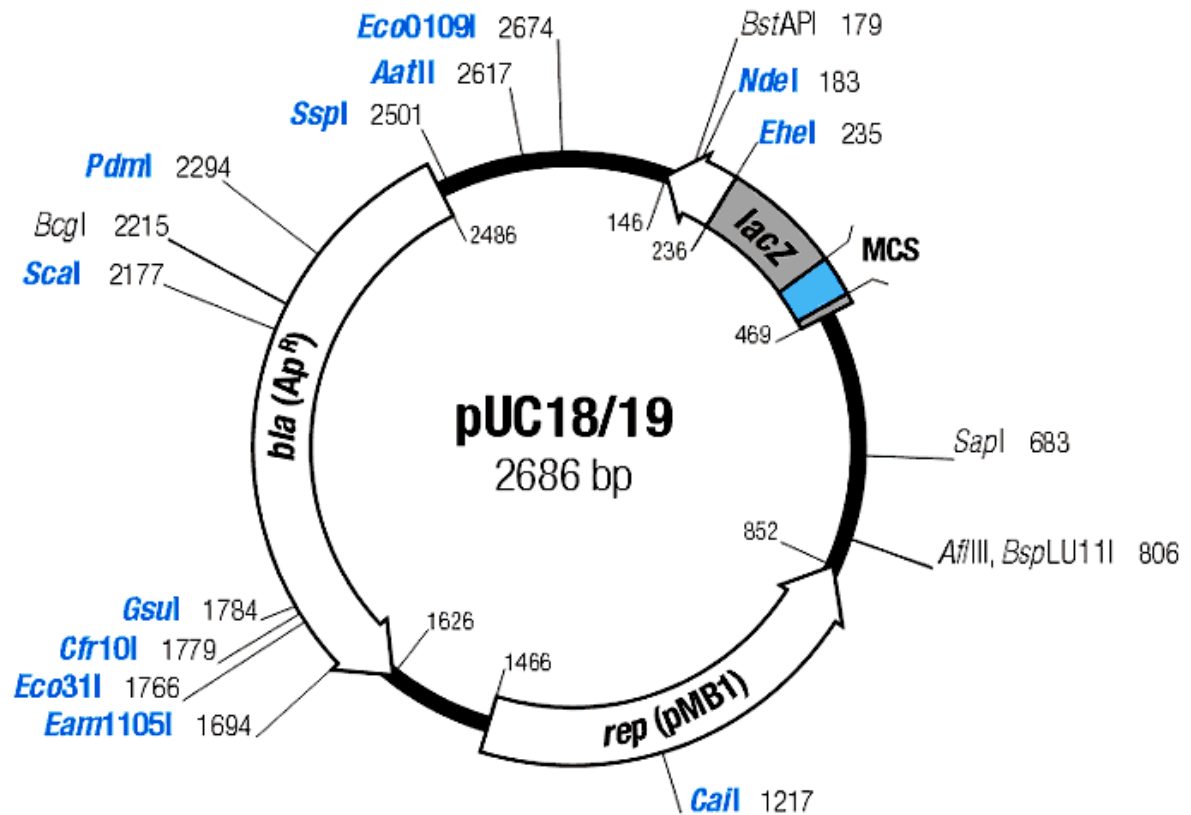
Enzyme	Source Microorganism	Recognition Site ^a	Ends Produced
BamI	<i>Bacillus amyloliquefaciens</i>	↓ -G-G-A-T-C-C- -C-C-T-A-G-G- ↑	Sticky
EcoRI	<i>Escherichia coli</i>	↓ -G A A T T C- -C T T A A G- ↑	Sticky
HindIII	<i>Haemophilus influenzae</i>	↓ -A-A-G-C-T-T- -T-T-C-G-A-A- ↑	Sticky
KpnI	<i>Klebsiella pneumoniae</i>	↓ -G-G-T-A-C-C- -C-C-A-T-G-G- ↑	Sticky

Uses of Restriction Enzymes

- Recombinant DNA technology
 - Cloning
 - cDNA/genomic library construction
 - DNA mapping
-

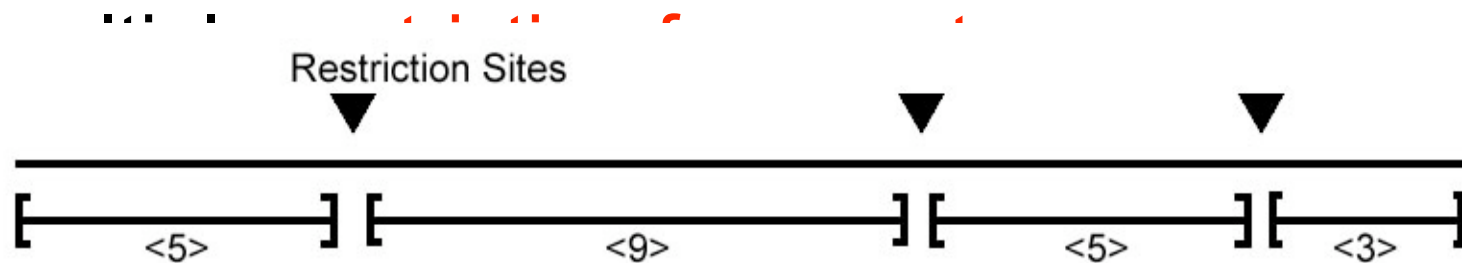
Restriction Maps

- A map showing positions of restriction sites in a DNA sequence
- If DNA sequence is known then construction of restriction map is a trivial exercise
- In early days of molecular biology DNA sequences were often unknown
- Biologists had to solve the problem of constructing restriction maps **without knowing DNA sequences**



Full Restriction Digest

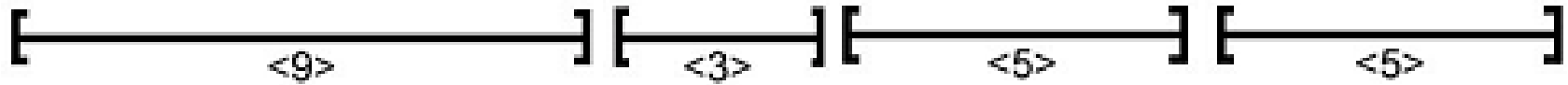
- Cutting DNA at each restriction site creates



- Is it possible to reconstruct the order of the fragments from the sizes of the fragments $\{3,5,5,9\}$?

Full Restriction Digest: Multiple Solutions

- Alternative ordering of restriction fragments:



VS



Measuring Length of Restriction Fragments

- Restriction enzymes break DNA into restriction fragments.
 - **Gel electrophoresis** is a process for separating DNA by size and measuring sizes of restriction fragments
 - Can separate DNA fragments that differ in length in only 1 nucleotide for fragments up to 500 nucleotides long
-

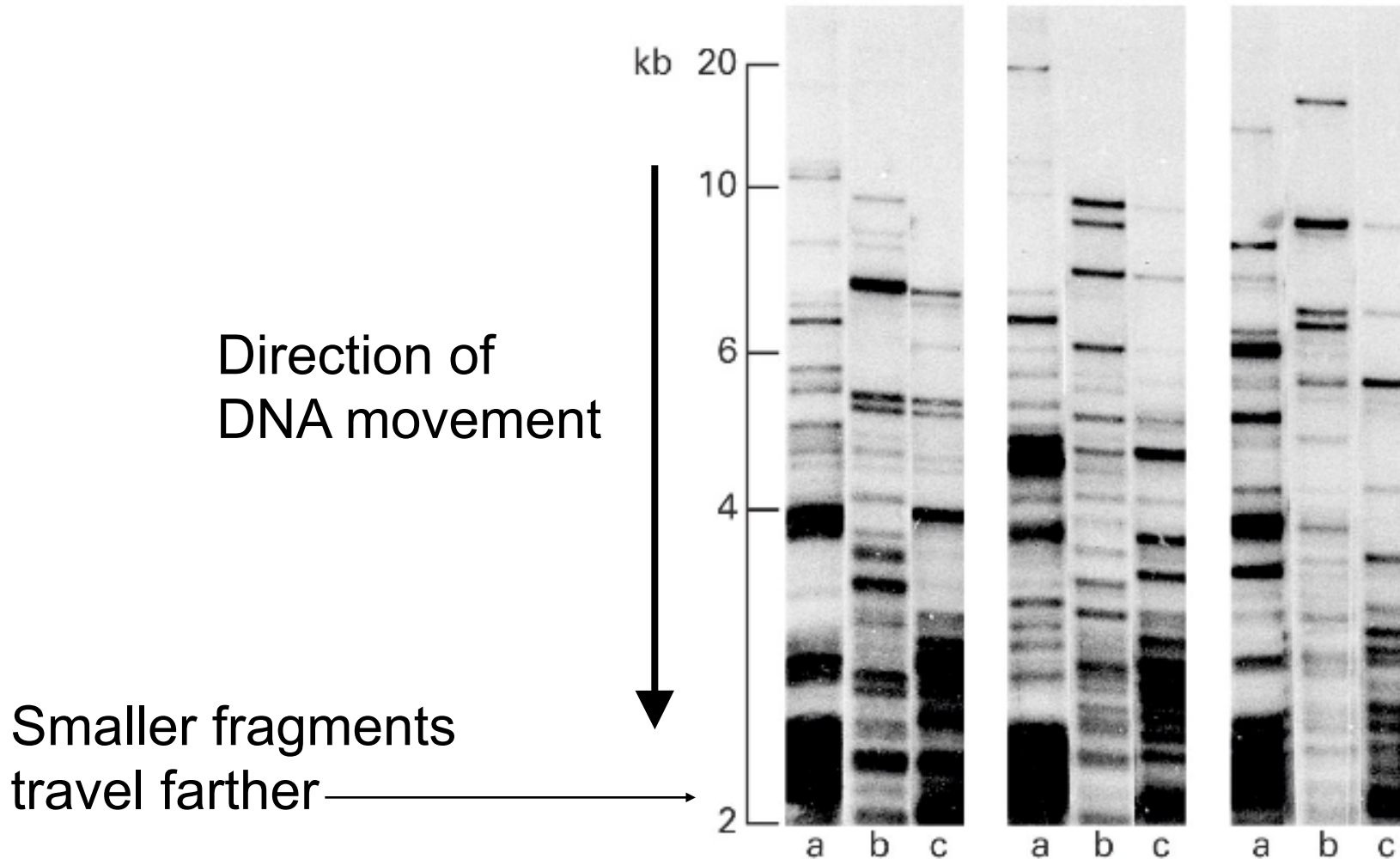
Gel Electrophoresis

- DNA fragments are injected into a gel positioned in an electric field
 - DNA are negatively charged near neutral pH
 - The ribose phosphate backbone of each nucleotide is acidic; DNA has an overall negative charge
 - DNA molecules move towards the positive electrode
-

Gel Electrophoresis (cont'd)

- DNA fragments of different lengths are separated according to size
 - Smaller molecules move through the gel matrix more readily than larger molecules
- The gel matrix restricts random diffusion so molecules of different lengths separate into different bands

Gel Electrophoresis: Example



Detecting DNA: Autoradiography

- One way to visualize separated DNA bands on a gel is **autoradiography**:
 - The DNA is radioactively labeled
 - The gel is laid against a sheet of photographic film in the dark, exposing the film at the positions where the DNA is present.
-

Detecting DNA: Fluorescence

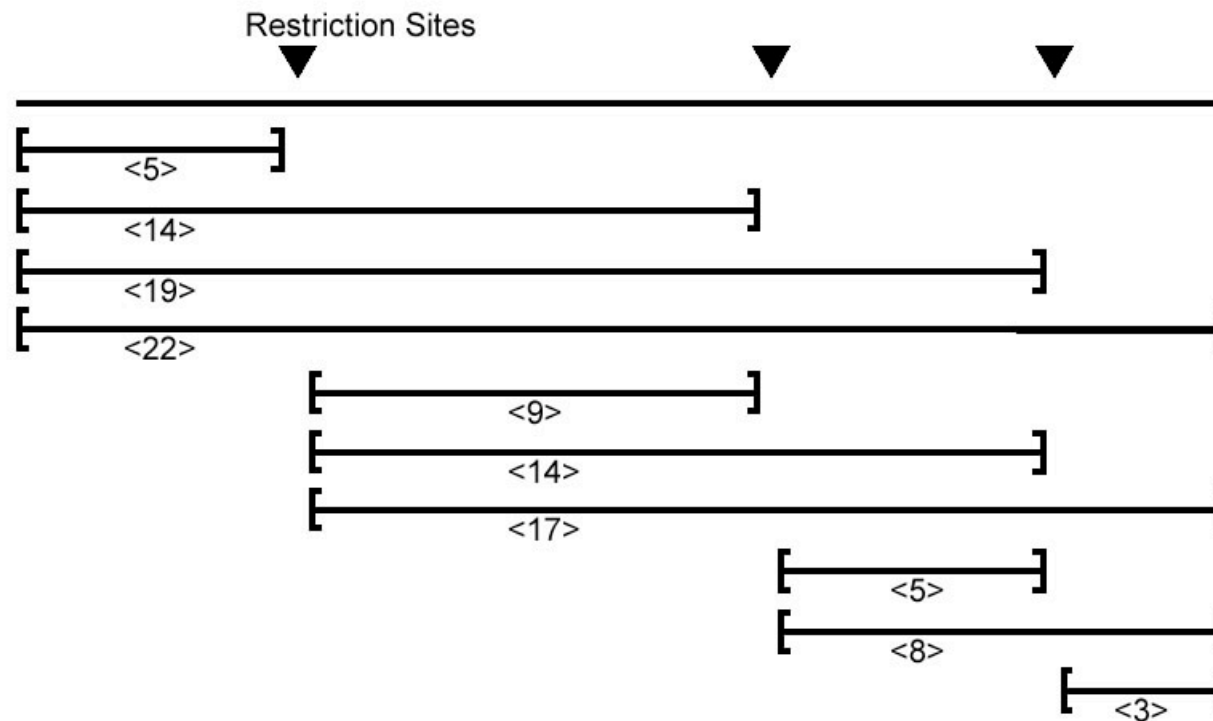
- Another way to visualize DNA bands in gel is **fluorescence**:
 - The gel is incubated with a solution containing the fluorescent dye ethidium
 - Ethidium binds to the DNA
 - The DNA lights up when the gel is exposed to ultraviolet light.
-

Partial Restriction Digest

- The sample of DNA is exposed to the restriction enzyme for only a limited amount of time to prevent it from being cut at all restriction sites
 - This experiment generates the set of all possible restriction fragments between every two (not necessarily consecutive) cuts
 - This set of fragment sizes is used to determine the positions of the restriction sites in the DNA sequence
-

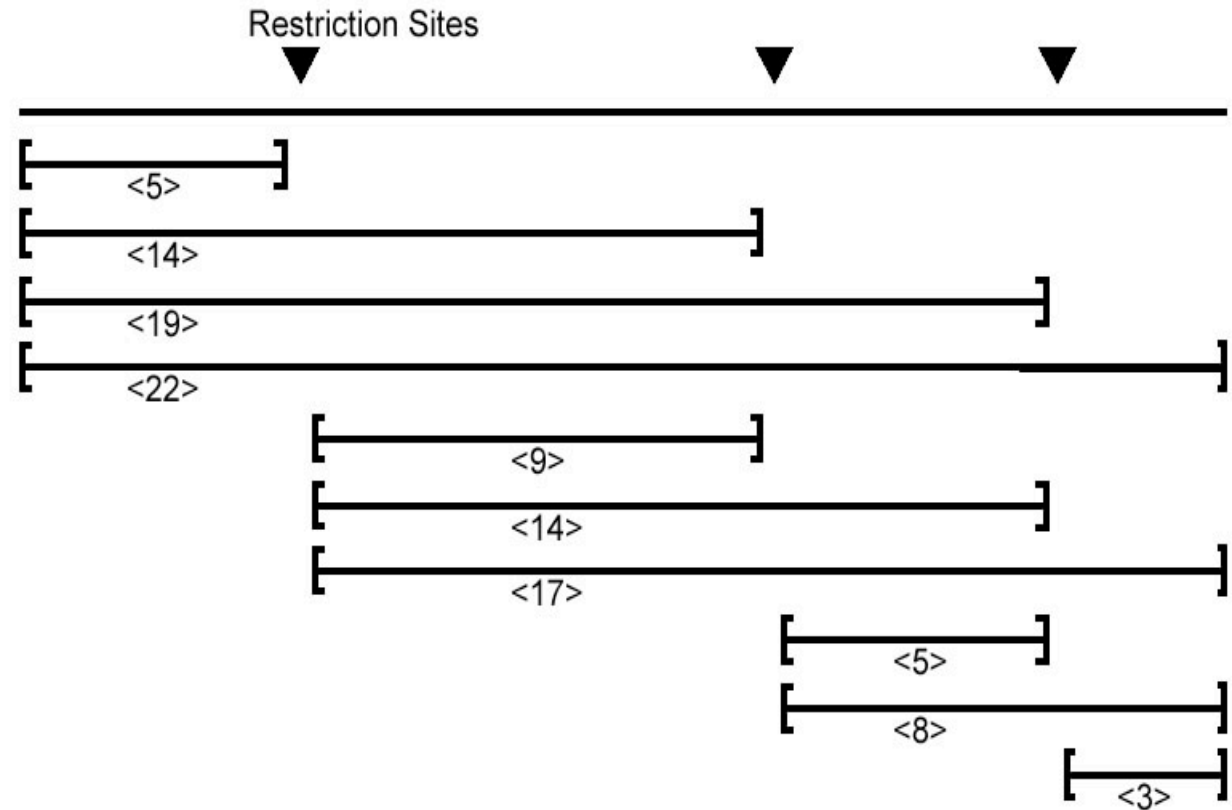
Partial Digest Example

- Partial Digest results in the following 10 restriction fragments:



Multiset of Restriction Fragments

- We assume that multiplicity of a fragment can be detected, i.e., the number of restriction fragments of the same length can be determined (e.g., by observing twice as much fluorescence intensity for a double fragment than for a single fragment)



Multiset: {3, 5, 5, 8, 9, 14, 14, 17, 19, 22}

Partial Digest Fundamentals

X : the set of n integers representing the location of all cuts in the restriction map, including the start and end

n : the total number of cuts

DX : the multiset of integers representing lengths of each of the $\binom{n}{2}$ fragments produced from a partial digest

One More Partial Digest Example

X	0	2	4	7	10
0		2	4	7	10
2			2	5	8
4				3	6
7					3
10					

Representation of $D\mathbf{X} = \{2, 2, 3, 3, 4, 5, 6, 7, 8, 10\}$ as a two dimensional table, with elements of

$$\mathbf{X} = \{0, 2, 4, 7, 10\}$$

along both the top and left side. The elements at (i, j) in the table is $x_j - x_i$ for $1 \leq i < j \leq n$.

Partial Digest Problem: Formulation

Goal: Given all pairwise distances between points on a line, reconstruct the positions of those points

- Input: The multiset of pairwise distances L , containing $n(n-1)/2$ integers
- Output: A set X , of n integers, such that $DX = L$

Partial Digest: Multiple Solutions

- It is not always possible to uniquely reconstruct a set X based only on DX .

- For example, the set

$$X = \{0, 2, 5\}$$

and

$$(X + 10) = \{10, 12, 15\}$$

both produce $DX = \{2, 3, 5\}$ as their partial digest set.

- The sets $\{0, 1, 2, 5, 7, 9, 12\}$ and $\{0, 1, 5, 7, 8, 10, 12\}$ present a less trivial example of non-uniqueness. They both digest into:

$$\{1, 1, 2, 2, 2, 3, 3, 4, 4, 5, 5, 5, 6, 7, 7, 7, 8, 9, 10, 11, 12\}$$

Homometric Sets

	0	1	2	5	7	9	12
0		1	2	5	7	9	12
1			1	4	6	8	11
2				3	5	7	10
5					2	4	7
7						2	5
9							3
12							

	0	1	5	7	8	10	12
0		1	5	7	8	10	12
1			4	6	7	9	11
5				2	3	5	7
7					1	3	5
8						2	4
10							2
12							

Brute Force Algorithms

- Also known as exhaustive search algorithms; examine every possible variant to find a solution
 - Efficient in rare cases; usually impractical
-

Partial Digest: Brute Force

1. Find the restriction fragment of maximum length M .
 M is the length of the DNA sequence.
2. For every possible set

$$\mathbf{X} = \{0, x_2, \dots, x_{n-1}, M\}$$

compute the corresponding $D\mathbf{X}$

5. If $D\mathbf{X}$ is equal to the experimental partial digest L , then \mathbf{X} is the correct restriction map

BruteForcePDP

1. BruteForcePDP(L, n):
 2. $M \leftarrow$ maximum element in L
 3. for every set of $n - 2$ integers $0 < x_2 < \dots < x_{n-1} < M$
 4. $X \leftarrow \{0, x_2, \dots, x_{n-1}, M\}$
 5. Form DX from X
 6. if $DX = L$
 7. return X
 8. output “no solution”
-

Efficiency of BruteForcePDP

- BruteForcePDP takes $O(M^{n-2})$ time since it must examine all possible sets of positions.
- One way to improve the algorithm is to limit the values of x_j to only those values which occur in L .

AnotherBruteForcePDP

1. AnotherBruteForcePDP(L, n)
 2. $M \leftarrow$ maximum element in L
 3. for every set of $n - 2$ integers $0 < x_2 < \dots < x_{n-1} < M$
 4. $X \leftarrow \{ 0, x_2, \dots, x_{n-1}, M \}$
 5. Form DX from X
 6. if $DX = L$
 7. return X
 8. output “no solution”
-

AnotherBruteForcePDP

1. AnotherBruteForcePDP(L, n)
2. $M \leftarrow$ maximum element in L
3. for every set of $n - 2$ integers $0 < x_2 < \dots < x_{n-1} < M$ from L
4. $X \leftarrow \{ 0, x_2, \dots, x_{n-1}, M \}$
5. Form DX from X
6. if $DX = L$
7. return X
8. output “no solution”

Efficiency of AnotherBruteForcePDP

- It's more efficient, but still slow
- If $L = \{2, 998, 1000\}$ ($n = 3$, $M = 1000$), BruteForcePDP will be extremely slow, but AnotherBruteForcePDP will be quite fast
- Fewer sets are examined, but runtime is still exponential: $O(n^{2n-4})$

Branch and Bound Algorithm for PDP

1. Begin with $X = \{0\}$
2. Remove the largest element in L and place it in X
3. See if the element *fits* on the right or left side of the restriction map
4. When it fits, find the other lengths it creates and remove those from L
5. Go back to step 1 until L is empty

Branch and Bound Algorithm for PDP

1. Begin with $X = \{0\}$
2. Remove the largest element in L and place it in X
3. See if the element *fits* on the right or left side of the restriction map
4. When it fits, find the other lengths it creates and remove those from L
5. Go back to step 1 until L is empty

WRONG ALGORITHM

Defining $D(y, X)$

- Before describing PartialDigest, first define

$$D(\mathbf{y}, \mathbf{X})$$

as the multiset of all distances between point \mathbf{y} and all other points in the set \mathbf{X}

$$D(\mathbf{y}, \mathbf{X}) = \{|\mathbf{y} - x_1|, |\mathbf{y} - x_2|, \dots, |\mathbf{y} - x_n|\}$$

for $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$

PartialDigest Algorithm

PartialDigest(L):

$width \leftarrow$ Maximum element in L

DELETE($width, L$)

$X \leftarrow \{0, width\}$

PLACE(L, X)

PartialDigest Algorithm (cont'd)

1. PLACE(L, X)
2. if L is empty
3. output X
4. return
5. $y \leftarrow$ maximum element in L
6. Delete(y, L)
7. if $D(y, X) \notin L$
8. Add y to X and remove lengths $D(y, X)$ from L
9. PLACE(L, X)
10. Remove y from X and add lengths $D(y, X)$ to L
11. if $D(\text{width}-y, X) \notin L$
12. Add $\text{width}-y$ to X and remove lengths $D(\text{width}-y, X)$ from L
13. PLACE(L, X)
14. Remove $\text{width}-y$ from X and add lengths $D(\text{width}-y, X)$ to L
15. return

An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$
$$X = \{ 0 \}$$

An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0 \}$$

Remove 10 from L and insert it into X . We know this must be the length of the DNA sequence because it is the largest fragment.

An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 10 \}$$



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 10 \}$$

Take 8 from L and make $y = 2$ or 8. But since the two cases are symmetric, we can assume $y = 2$.



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 10 \}$$

We find that the distances from $y=2$ to other elements in X are $D(y, X) = \{8, 2\}$, so we remove $\{8, 2\}$ from L and add 2 to X .



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 10 \}$$



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 10 \}$$

Take 7 from L and make $y = 7$ or $y = 10 - 7 = 3$. We will explore $y = 7$ first, so $D(y, X) = \{7, 5, 3\}$.



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 10 \}$$

For $y = 7$ first, $D(y, X) = \{7, 5, 3\}$. Therefore we remove $\{7, 5, 3\}$ from L and add 7 to X .



$$D(y, X) = \{7, 5, 3\} = \{1/27 - 0^{1/2}, 1/27 - 2^{1/2}, 1/27 - 10^{1/2}\}$$

An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 7, 10 \}$$

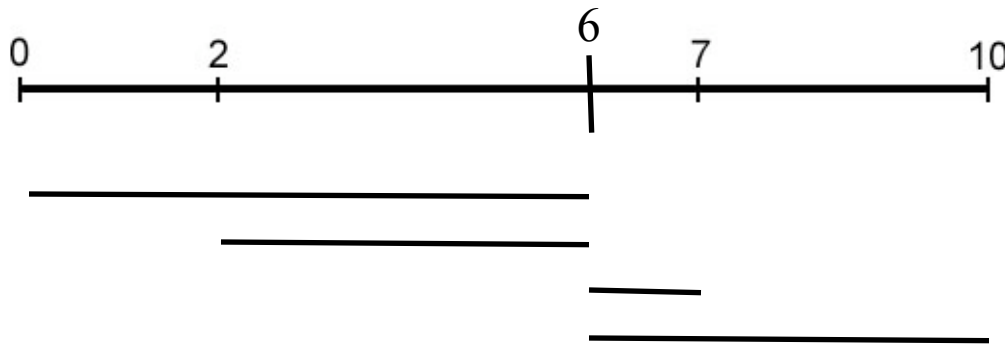


An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 7, 10 \}$$

Take 6 from L and make $y = 6$. Unfortunately $D(y, X) = \{6, 4, 1, 4\}$, which is not a subset of L . Therefore we won't explore this branch.



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 7, 10 \}$$

This time make $y = 4$. $D(y, X) = \{4, 2, 3, 6\}$, which is a subset of L so we will explore this branch. We remove $\{4, 2, 3, 6\}$ from L and add 4 to X .



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 4, 7, 10 \}$$



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 4, 7, 10 \}$$

L is now empty, so we have a solution, which is X .



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 7, 10 \}$$

To find other solutions, we backtrack.



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 10 \}$$

More backtrack.



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 2, 10 \}$$

This time we will explore $y = 3$. $D(y, X) = \{3, 1, 7\}$, which is not a subset of L , so we won't explore this branch.



An Example

$$L = \{ 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 \}$$

$$X = \{ 0, 10 \}$$

We backtracked back to the root. Therefore we have found all the solutions.



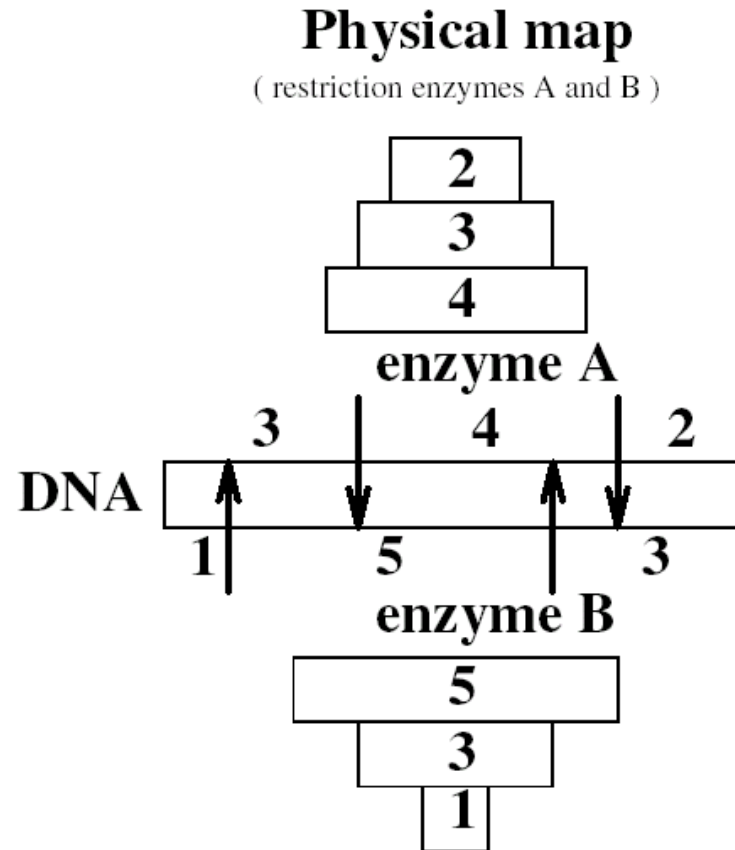
Analyzing PartialDigest Algorithm

- Still exponential in worst case, but is very fast on average
- Informally, let $T(n)$ be time PartialDigest takes to place n cuts
 - No branching case: $T(n) < T(n-1) + O(n)$
 - Quadratic
 - Branching case: $T(n) < 2T(n-1) + O(n)$
 - Exponential

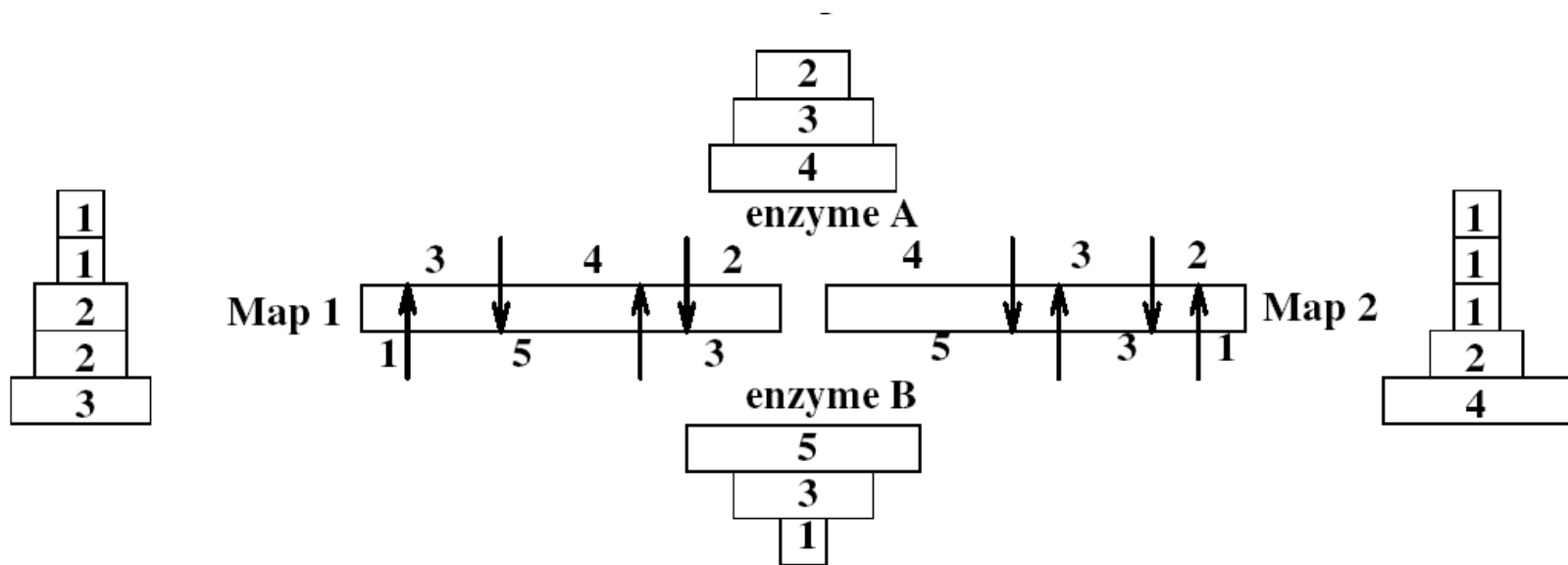
Double Digest Mapping

- Double Digest is yet another experimentally method to construct restriction maps
 - Use two restriction enzymes; three **full** digests:
 - One with only first enzyme
 - One with only second enzyme
 - One with both enzymes
- Computationally, Double Digest problem is more complex than Partial Digest problem

Double Digest: Example



Double Digest: Example



Without the information about X (i.e. $A+B$), it is impossible to solve the double digest problem as this diagram illustrates

Double Digest Problem

Input: dA – fragment lengths from the digest with enzyme A .

dB – fragment lengths from the digest with enzyme B .

dX – fragment lengths from the digest with *both* A and B .

Output: A – location of the cuts in the restriction map for the enzyme A .

B – location of the cuts in the restriction map for the enzyme B .

Double Digest: Multiple Solutions

