Computational Biology

Lecture 12

Saad Mneimneh

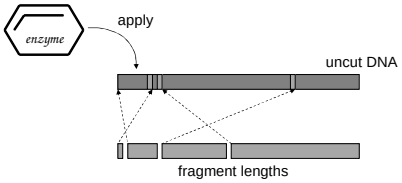
Physical Mapping

A physical map of a DNA tells the location of precisely defined sequences along the molecule.

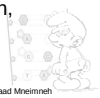
- Restriction mapping: mapping of restriction sites of a cutting enzyme based on lengths of fragments
  - Double Digest Problem DDP
  - Partial Digest Problem PDP
- Hybridization mapping: mapping clones based on hybridization data with probes
  - Non-unique probes
  - Unique probes

Saad Mneimneh

Restriction Mapping

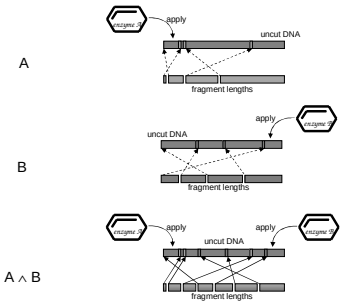


- Ordering the fragments maps the restriction sites on the DNA.
- But lengths of fragments is not enough information, any order would satisfy the experiment data.
- Use two cutting enzymes



Saad Mneimneh

Double Digest



Saad Mneimneh

Double Digest Problem

Double Digest Problem DDP

given

1. multiset *A* of lengths from enzyme A
2. multiset *B* of lengths from enzyme B
3. multiset *C* of lengths from enzymes A ∧ B

determine an ordering of the fragments that is consistent with the three experiments

Saad Mneimneh

Example

- : {3, 6, 8,10}
- : {4, 5, 7, 11}
- ∧ : {1, 2, 3, 3, 5, 6, 7}

A	3		8		6		10	
C = A ∧ B	3	1	5	2	6	3	7	
B	4		5		11		7	

Saad Mneimneh

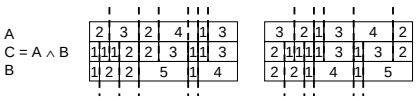
Number of solutions

- The solution for a DDP might not be unique.
- The number of solutions grows exponentially

Saad Mneimneh

Example

-  : {1, 2, 2, 3, 3, 4}
-  : {1, 1, 2, 2, 4, 5}
-  ∧  : {1, 1, 1, 1, 1, 2, 2, 3, 3}



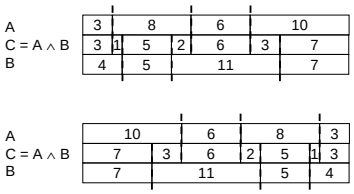
Saad Mneimneh

Equivalence of Solutions

- Some different solutions might be equivalent.
- For instance, if (a_1, a_2, …, a_m) (b_1, b_2, …, b_n) is a solution, then (a_m, a_{m-1}, …, a_1) (b_n, b_{n-1}, …, b_1) is also a solution.
- This is a Reflection. No fragment length data could possibly distinguish between the two, they only differ by orientation.

Saad Mneimneh

Reflection



Saad Mneimneh

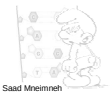
Overlap Equivalence

- Let's define a more general type of equivalence called overlap equivalence.
- Let $\{A_i\}$ be the set of fragments from A and $\{B_j\}$ be the set of fragments from B.
- A solution defines an overlap matrix O, s.t. $O_{ij} = 1$ if A_i overlaps with B_j.
- Two solutions are *overlap equivalent* if they define the same overlap matrix O.
- Reflections are overlap equivalent.

Saad Mneimneh

Equivalence class

- A solution with all its overlap equivalent solutions form an equivalence class (this is an equivalence relation).
- Given a solution,
  - What is the size of its equivalence class?
  - Can we generate all solutions in the class?



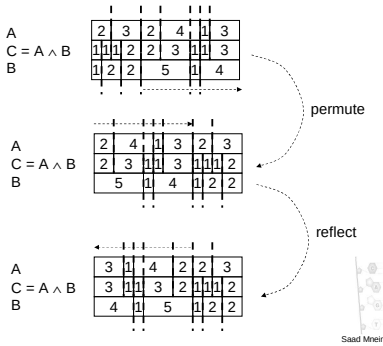
Saad Mneimneh

Observations

- If a solution has t – 1 coincident cuts sites, then it has t components.
- They can be permuted in $t!$ ways without changing the overlap data.
- Each component can also be reflected without changing the overlap.
- Therefore, we can generate $2^t t!$ solutions.

Saad Mneimneh

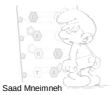
Example



Saad Mneimneh

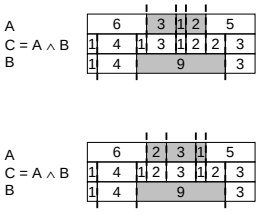
Another observation

- Given a solution, let
  - $\mathcal{A}_j = \{ A_k : A_k \subset Bj \}$
  - $\mathcal{B}_i = \{ B_k : B_k \subset A_i \}$
- Permuting $\mathcal{A}_j$ and $\mathcal{B}_i$ does not change the overlap data



Saad Mneimneh

Example



Saad Mneimneh

Size of equivalence class

- Is it $2^t t! \prod |\mathcal{A}_j|! \prod |\mathcal{B}_i|!$
- Not quite!
- If a component has only one fragment in either *A* or *B*, then a reflection is also a permutation.
- Let s be the number of such components, then the size of the equivalence class is:

$$2^{(t-s)}\, t! \prod |\mathcal{A}_j|! \prod |\mathcal{B}_i|!$$

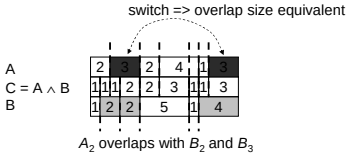
Saad Mneimneh

Other Equivalences?

- We can define other kinds of equivalences.
- Consider overlap size equivalence, i.e. two solutions are equivalent if they produce the same overlap sizes.
- Overlap equivalence => overlap size equivalence, but not the other way around.

Saad Mneimneh

Example



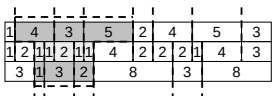
Saad Mneimneh

Cassette Transformation Equivalence

- Let $|C| = l$
- For $1 \le i \le j \le l$, $I_c = \{ C_k : i \le k \le j \}$ is the set of fragments from C_i to C_j.
- The cassette defined by I_C is the set of fragments (I_A, I_B) that contain a fragment from I_C.

Saad Mneimneh

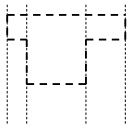
Cassette



Cassette for $I_c = \{ C_3, C_4, C_5, C_6, C_7 \}$

Saad Mneimneh

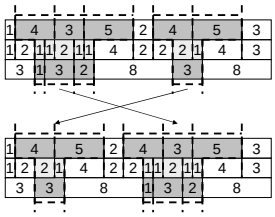
Cassette
left and right overlap



left overlap = $m_a - m_b$ $m_a' - m_b'$ = right overlap

Saad Mneimneh

Cassette exchange

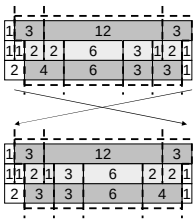


Left overlap = -2
Right overlap = 4

Two cassettes with the same left and right overlap can be exchanged

Saad Mneimneh

Cassette Reflection



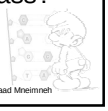
Left overlap = -1
Right overlap = 1

A cassettes with the same left and right overlap (but different signs) can be reflected

Saad Mneimneh

Cassette Equivalence

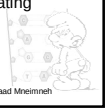
- Two solutions are *cassette equivalent* if there exists a series of cassette transformations (exchanges and reflections) that take on to the other.
- What is the size of an equivalence class?



Saad Mneimneh

Alternating Euler Paths

- Consider a graph with colored edges
- An Euler path (cycle) is a path (cycle) that goes through every edge once
- An alternating Euler path (cycle) is an Euler path (cycle) such that consecutive edges on the path (cycle) have different colors
- Pevzner 1995 showed that given a solution, we can construct a special bi-colored graph called the border block graph.
- Each cassette equivalent solution corresponds to an alternating Euler path (cycle) in the graph and vice-versa.



Saad Mneimneh

Fact

- Let $d_i(v)$ be the number of edges of color i incident to v.
- An edge bi-colored connected graph with $d_A(v) = d_B(v)$ has an alternating Euler cycle.
- Proof:
  - Every vertex has even degree; therefore, the graph contains an Euler cycle.
  - Construct the Euler cycle the usual way, but by using only distinct color edges when traversing a vertex.

Saad Mneimneh

Exchange

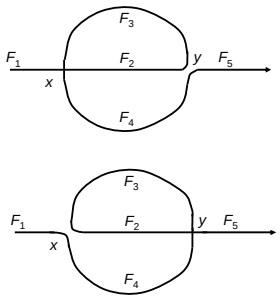
- Consider an alternating path

  …x…y…x…y…

- It consists of 5 parts $F_1F_2F_3F_4F_5$
- $F_1F_2F_3F_4F_5 \rightarrow F_1F_4F_3F_2F_5$ is called an exchange if $F_1F_4F_3F_2F_5$ is an alternating path

Saad Mneimneh

Illustration



Saad Mneimneh

Reflection

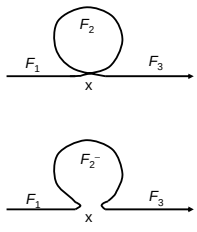
- Consider an alternating path

  …x…x…

- It consists of 3 parts $F_1F_2F_3$
- $F_1F_2F_3 \rightarrow F_1F_2^-F_3$ is called a reflection if $F_1F_2^-F_3$ is an alternating path, where F_2^- is the reverse of F_2.

Saad Mneimneh

Illustration



Saad Mneimneh

Fact

- Every two alternating Euler cycles in a bi-colored graph can be transformed into each other by a series of exchanges and reflections.
- Proof: Pevzner p. 29

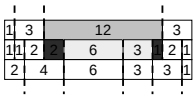
Saad Mneimneh

The border blocks

- Let $I(A_i) = \{ C_k : C_k \subset A_i \}$
- Let $I(B_j) = \{ C_k : C_k \subset B_j \}$
- If $|I(X)| > 1$, define the border blocks of X to be the left most and right most block in $I(X)$.
- C_i is a border block is it is a border block for some fragment X.

Saad Mneimneh

Example



$I(A_3) = \{ C_4, C_5, C_6, C_7 \}$
Border blocks of A_3: C_4 and C_7

Saad Mneimneh

Lemma

- Each fragment X with $|I(X)| > 1$ contains exactly two border blocks.
- $I(A_i) \cap I(B_j) \leq 1$
- Assume no cuts in A and B coincide, then each border block is a border block for some A_i and some B_j, except C_1 and C_l.

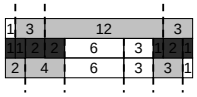
Saad Mneimneh

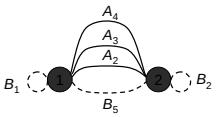
Border graph

- Let $\mathcal{B} = \{ C_k : C_k$ is a border block $\}$
- $V = \{ |C_k| : C_k \in \mathcal{B} \}$ vertices
- $E = \{ (|C_i|, |C_j|) : C_i$ and $C_j \in \mathcal{B} \cap I(X)$ for some $X \}$
- Each edge labeled by its X and colored A if $X \in A$ and B if $X \in B$.

Saad Mneimneh

Example





Saad Mneimneh

Alternating Euler path in border block graph

- Each vertex has equal number of edges of each color, except possibly for $|C_1|$ and $|C_l|$.
- By adding one or two edges (depending on the colors) we can fix this. Therefore, the graph has an alternating Euler path or cycle.
- Let $C_1 \ldots C_m$ be the ordered set of border blocks, then $P = |C_1| \ldots |C_m|$ is an alternating Euler path (cycle).

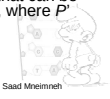
Saad Mneimneh

Result

Cassette transformations do not change the border graph.

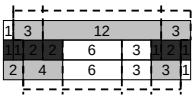
Let P be the alternating Euler path (cycle) corresponding a solution $[A, B]$.

- If a solution $[A', B']$ is obtained from $[A, B]$ by cassette exchange (reflection), it will have a path P' that can be obtained from P by an exchange (reflection).
- Let P' be an alternating Euler path (cycle) obtained from P by exchange (reflection). Then there is a solution $[A', B']$ that can be obtained from $[A, B]$ by cassette exchange (reflection), where P' corresponds to $[A', B']$



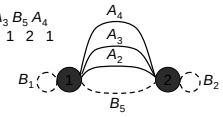
Saad Mneimneh

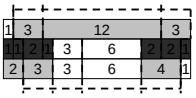
Example



reflection

This corresponds to the cycle: $B_1 A_2 B_2 A_3 B_5 A_4$
1 1 2 2 1 2 1

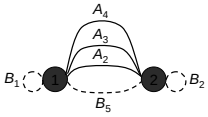




This corresponds to the cycle: $B_1 A_4 B_5 A_3 B_2 A_2$
1 1 2 1 2 2 1

Saad Mneimneh

Example



Cycle 1: $B_1 A_2 B_2 A_3 B_5 A_4$
1 1 2 2 1 2 1

Cycle 2: $B_1 A_4 B_2 A_3 B_5 A_2$
1 1 2 2 1 2 1

Cycle 1: $B_1 A_2 B_2 A_3 B_5 A_4$
1 1 2 2 1 2 1

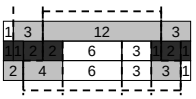
$B_1 A_2 B_5 A_3 B_2 A_4$
1 1 2 1 2 2 1

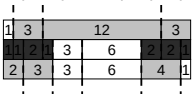
$B_1 A_2 B_5 A_3 B_2 A_4$
1 1 2 1 2 2 1

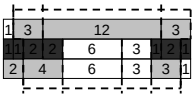
Cycle 2: $B_1 A_4 B_2 A_3 B_5 A_2$
1 1 2 2 1 2 1

Saad Mneimneh

Example (cont.)







$B_1 A_2 B_2 A_3 B_5 A_4$
1 1 2 2 1 2 1

$B_1 A_2 B_5 A_3 B_2 A_4$
1 1 2 1 2 2 1

$B_1 A_2 B_5 A_3 B_2 A_4$
1 1 2 1 2 2 1

$B_1 A_4 B_2 A_3 B_5 A_2$
1 1 2 2 1 2 1

Saad Mneimneh

DDP is NP-complete

Proof:

– *DPP* $\in$ NP. A solution for *DDP* can be verified in polynomial time.

– Set Partition problem (classical one), which is NP-complete, reduces to *DDP* in polynomial time.

Saad Mneimneh

DDP $\in$ NP

- Given
  - multiset *A* of lengths from enzyme A
  - multiset *B* of lengths from enzyme B
  - multiset *C* of lengths from enzymes A + B
- Solution
  - two sets of restriction sites, *a* and *b*.
- Verification:
  - Sort $g = a \cup b \cup \{0, L\}$, L = sum of all lengths in A
  - Compute multiset $c = \{ c_i: c_i = g_{i+1} - g_i, 0 < i < |g| \text{ and } g_{i+1} \neq g_i \}$
  - Sort *c* and *C* and compare them

Saad Mneimneh

Set Partition

Set Partition SP:

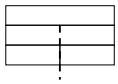
Given a set I of integers with total sum J

Can we partition I into two sets of sum $J/2$ each?

is NP-complete.

Given an *SP* instance, construct the following DDP:

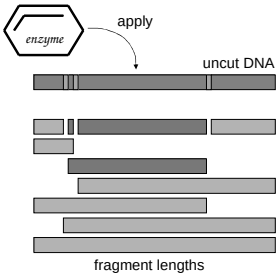
$A = I$
$B = \{J/2, J/2\}$
$C = I$



DPP has a solution iff *SP* has a solution

Saad Mneimneh

Partial Digest



- One restriction enzyme only
- Obtain fragments for
  - every pair of cuts
  - every cut and a boundary
- This can be achieved by *missing* a number of sites in every trial

Saad Mneimneh

Partial Digest Problem (turnpike problem)

Partial Digest Problem PDP

given

multiset Δx of distances between every pair of points on the line

$|\Delta x| = \binom{n}{2} = \frac{n(n-1)}{2}$

reconstruct the points on the line

Saad Mneimneh

PDP

- No polynomial time algorithm know
- Not known to be NP-complete
- Practical Backtracking algorithm due to Skiena et al. 1990

Saad Mneimneh

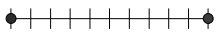
The algorithm

- Find longest distance in ΔX, this decides the two outermost points, delete that distance from ΔX.
- Repeatedly position the longest remaining distance of ΔX.
- Since the longer distance must be realized from one of the two outermost points, we have two possible positions (left and right) for the point.
- For each of these two positions, check whether all the distances from the position to the points already positioned are in ΔX.
- If they are, delete all those distances from ΔX and proceed.
- Backtrack if they are not for both of the two positions.

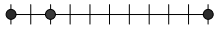
Saad Mneimneh

Example

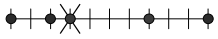
ΔX = { 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 }



ΔX = { 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 }



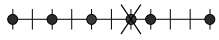
ΔX = { 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 }



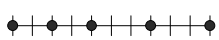
Saad Mneimneh

Example (cont.)

ΔX = { 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 }

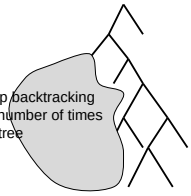


ΔX = { 2, 2, 3, 3, 4, 5, 6, 7, 8, 10 }



Saad Mneimneh

Worst case: Exponential



We could end up backtracking an exponential number of times in this decision tree

It is not trivial to show that an exponential number of backtracking could occur

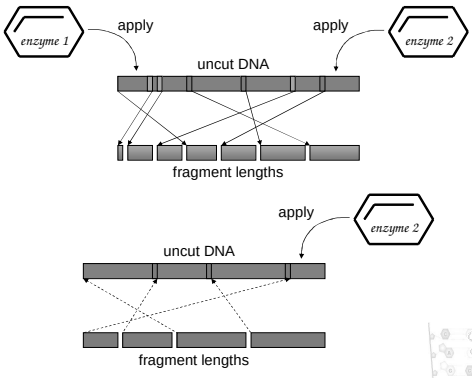
Zhang [1994] provided an example

Saad Mneimneh

In practice…

- If we have real points in general positions, then one of the two choices will be pruned with probability 1.
- The algorithm runs in $O(n^2 \log n)$ expected time since we have $O(n^2)$ positions and all operations can be done on ΔX using binary search which takes $O(\log n^2) = O(\log n)$.

Saad Mneimneh



Saad Mneimneh

Example

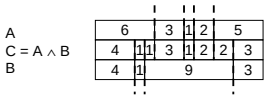
- 

: {1, 2, 3, 5, 6}
- 

: {1, 3, 4, 9}
- 

∧ 

: {1, 1, 1, 2, 2, 3, 3, 4}



Saad Mneimneh