

Restriction Maps

MPA-PRG: Programming in Bioinformatics

Exercise 6

Restriction Motifs

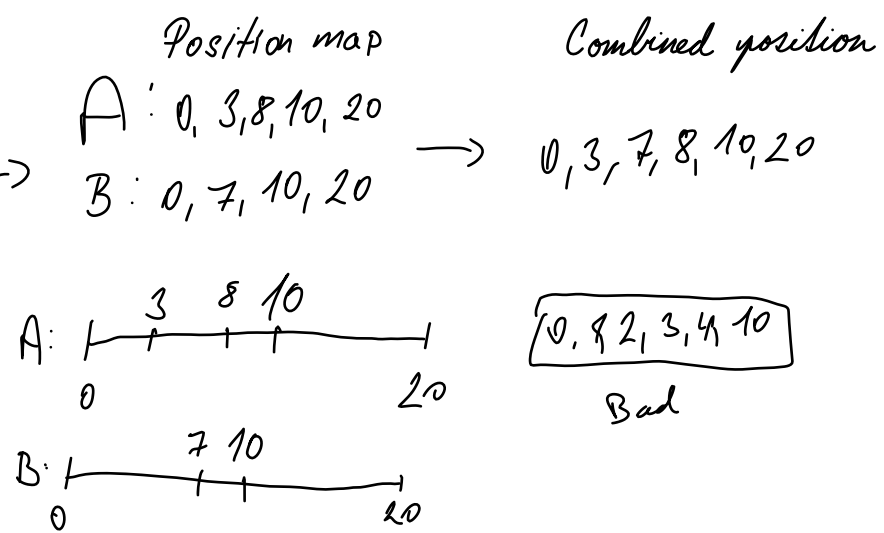
- specific recognition sites on DNA strand recognized by restriction endonuclease that cleaves DNA into fragments
- Double Digest Problem (DDP)
 - digestion of a single sequence by two different enzymes
- Partial Digest Problem (PDP)
 - incomplete (partial) digestion of a single sequence by a single enzyme

DDP

- assume sets of fragment sizes:
 - $\Delta X_A = \{a_1, a_2, \dots, a_m\}$ – products generated by digestion by enzyme A
 - $\Delta X_B = \{b_1, b_2, \dots, b_n\}$ – products generated by digestion by enzyme B
 - $\Delta X_{AB} = \{c_1, c_2, \dots, c_{m+n-1}\}$ – products generated by digestion by enzymes A + B
- enumeration of all possible combinations of fragment arrangements for both enzymes
- for each combination:
 - take the fragments sizes
 - create position maps
 - combine positions
 - calculate successive differences
 - sort differences
 - compare results with products of double digestion (ΔX_{AB})

DDP – Example

- $\Delta X_A = \{2, 3, 5, 10\}$
- $\Delta X_B = \{3, 7, 10\}$
- $\Delta X_{AB} = \{1, 2, 2, 5, 5, 5\}$

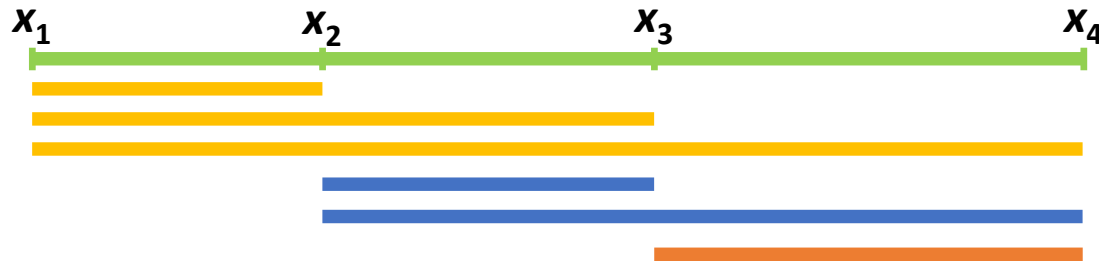


Task 1

- in R, create a function `DoubleDigestProblem()` for the brute-force DDP algorithm for one possible arrangement of fragments
- input:
 - a multiset of fragments generated by digestion by enzyme A
 - a multiset of fragments generated by digestion by enzyme B
 - a multiset of fragments generated by digestion by enzymes A + B
- output:
 - positions of restriction sites for enzyme A
 - positions of restriction sites for enzyme B
- modify it to search through all possible arrangements of fragments

PDP

- assume a multiset of fragment sizes ΔX and we search for a set of positions $X = \{x_1 = 0, x_2, \dots, x_n\}$, that generates $\Delta X = \{x_j - x_i: 1 \leq i < j \leq n\}$



- the number of restriction sites:

$$\frac{N(N - 1)}{2} = m$$

N – the number of restriction sites

m – the number of fragments in the multiset ΔX

PDP

- $x_1 = 0$
- find maximum value in ΔX = corresponds to the longest fragment
- deduce possible position of restriction site
- remove fragments sizes from ΔX that are generated by this position
- repeat until all fragments are exhausted

PDP – Example

- $\Delta X = \{2, 2, 3, 3, 4, 5, 6, 7, 8, 10\}$

Task 2

- in R, implement a function `Place()` according to the following pseudocode
- input:
 - a multiset of fragments lengths
 - a list of the current positions of the restriction sites
 - length of the DNA molecule i.e. the size of the longest fragment
- output:
 - a list of the positions of the restriction sites
- $\Delta(y, X)$ is a multiset of lengths between value y and all values in X
- **hint:** create additional function `Remove()`, which removes from ΔX all used lengths $\Delta(y, X)$

Task 2

```
Place(deltaX, X, width)
1   if deltaX is empty
2       output X
3       return
4   y ← the maximum element from deltaX
5   if  $\Delta(y, X) \subseteq \text{deltaX}$ 
6       add y to X and remove the lengths  $\Delta(y, X)$  from deltaX
7       Place(deltaX, X, width)
8       remove y from X and add the lengths  $\Delta(y, X)$  to deltaX
9   if  $\Delta(\text{width} - y, X) \subseteq \text{deltaX}$ 
10      add width - y to X and remove the lengths  $\Delta(\text{width} - y, X)$  from deltaX
11      Place(deltaX, X, width)
12      remove width - y from X and add the lengths  $\Delta(\text{width} - y, X)$  to deltaX
13  return
```

Task 3

- in R, implement a function `PartialDigestProblem()` according to the following pseudocode
- input:
 - a multiset of fragments lengths

`PartialDigestProblem(deltaX)`

```
1   width ← the maximum element from deltaX
2   delete the maximum element from deltaX
3   X ← {0, width}
4   Place(deltaX, X, width)
```