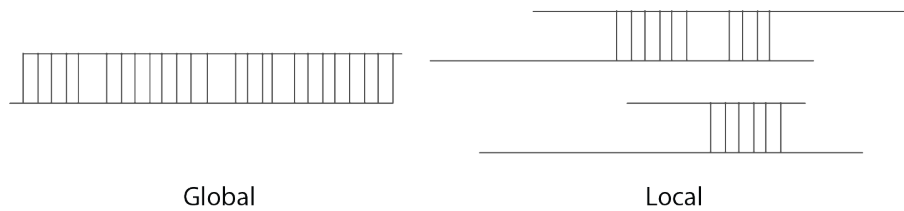


## 4 Local alignment

### 4.1 Local alignments

Local pairwise alignments are aligned pairs of sub-sequences that have certain level of similarities.

#### Deifference between global and local alignments



**Figure 4.1:** Global and local alignments

#### Elements of local alignment

- Segment: a substring of a sequence
- Segment pair: a pair of segments
- Local alignment: an alignment of a segment pair

#### Elements of local alignment

- Dynamic programming (Smith–Waterman)
- Dot matrix

#### Applications

- Sequence motifs
- Conserved regions
- Inverted repeats

### 4.2 Local alignment with DP

Dynamic programming can be used to find local alignments.

#### Requirements

- Find all local alignments between two sequences
- Assign scores to all local alignments

## Modification of DP for local alignments

- The minimum alignment score must be 0
- Some entries of the score matrix should be negative
- Backtracking also needs to be modified

## Update rule of DP cells

$$\begin{aligned}
 H_{i,j}^{(0)} &= H_{i-1,j} - g && (\textit{vertical}) \\
 H_{i,j}^{(1)} &= H_{i,j-1} - g && (\textit{horizontal}) \\
 H_{i,j}^{(2)} &= H_{i-1,j-1} + R_{a,b} && (\textit{diagonal}) \\
 H_{i,j}^{(2)} &= 0 && (\textit{minimumscore})
 \end{aligned}$$

## Example of cell update

	A	C
C	0.1	0.4
T	0.2	0

Scoring scheme:

Match: 0.5

Mismatch: -0.3

Gap penalty: 0.5

$$\begin{aligned}
 H_{i,j}^{(0)} &= -0.1 && (\textit{vertical}) \\
 H_{i,j}^{(1)} &= -0.3 && (\textit{horizontal}) \\
 H_{i,j}^{(2)} &= -0.2 && (\textit{diagonal}) \\
 H_{i,j}^{(2)} &= 0 && \checkmark (\textit{minimumscore})
 \end{aligned}$$

## Backtracking for local alignments

It starts from the cells with the maximum score instead of the right bottom cell.

- Start cells: cells with the maximum score
- End cells: cells with 0

**N.B.** the end cell with score 0 should not be included in the alignment.

## Example of backtracking

		A	C	G	C	
		0	0	0	0	
C		0	0	0.5	0	0.5
G		0	0	0	1	0.5
A		0	0.5	0	0.5	0.7

Local alignment  
**q: 2 CG 3**  
**d: 2 CG 3**

## Pseudo-code of updating DP table for local alignment

The cells in the first row and the first column are initialized with 0.

---

**Algorithm 4.1:** Update dynamic programming table for global alignment

---

```

Hi,j : Dynamic programming table
Ra,b: Match/mismatch scores
g   : Gap penalty

// Initialization
for i ← 0 to m do
    | Hi,0 ← 0;
end
for j ← 1 to n do
    | H0,j ← 0;
end

// Main loop for table update
for i ← 1 to m do
    for j ← 1 to n do
        | Hi,j ← max(0, Hi-1,j - g, Hi,j-1 - g, Hi-1,j-1 + Ra,b);
    end
end
end

```

---

## Exercise 4.1

Use DP to find a local alignment.

q/d		A	G	C	C
A					
G					
C					

Scoring scheme:  
 Match: 0.2  
 Mismatch: -0.2  
 Gap penalty: 0.2

## 4.3 Dot matrix

Using a dot matrix is an effective and easy way to find local similarities.

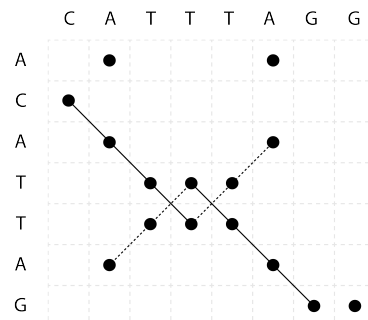
## Basic concept

It uses an  $m \times n$  binary matrix from two sequences.

- A dot: match
- Empty: mismatch

## Example of dot matrix

q: ACATTAG, d: CATTAGG



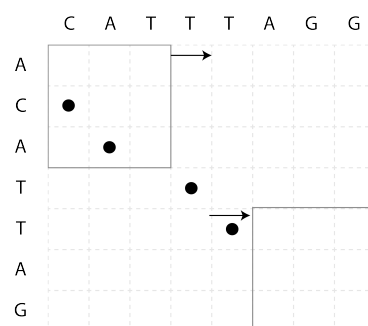
**Figure 4.2:** Dot matrix of  $7 \times 8$

It is easy to find segment pairs with a dot matrix. Contiguous dots along diagonals indicate local alignments. It is also easy to find other similarities. For instance, contiguous dots along anti-diagonals indicate reversed substrings.

## Filtering of dot matrix

Dot matrices usually get noisy with too many dots. Overlapping windows are usually applied to reduce the noise.

## Example of filtering



**Figure 4.3:** Filtered dot matrix with window size 3 and threshold 3.

## Exercise 4.2

Find local similarities between two DNA sequences, q: GATTACA and d: GGATTAC.

1. Create a dot matrix for the two sequences.
2. Filter dots with overlapping windows size 3 and threshold 3.