# Dynamic Programming

- Sequence Analysis
- The Sequence Alignment Problem
- Dynamic Programming Formulation

Imdad ullah Khan

# Sequence Alignment: Problem

**Input:** Two sequences $X = x_1, \ldots, x_m$ and $Y = y_1, \ldots, y_n$ over $\Sigma$
and a score/penalty matrix

**Output:** Minimum cost alignment between $X$ and $Y$

Cost of the optimal alignment is the alignment distance between $X$ and $Y$

**Theorem:** If all penalties are 1, alignment distance = edit distance (otherwise it is equal to the weighted edit distance)

- It implies that alignment distance is a metric
- Edit distance does not specify the edits (it just counts)
- Optimal alignment does not only compute edit distance but also specify the edits

# Sequence Alignment: Problem

**Input:** Two sequences $X = x_1, \ldots, x_m$ and $Y = y_1, \ldots, y_n$ over $\Sigma$

**Output:** Minimum cost alignment between $X$ and $Y$

Cost of the optimal alignment is the alignment distance between $X$ and $Y$

- $\text{OPT}(i,j)$: optimal alignment between $x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots, y_j$
- $D(i,j)$: alignment distance between $x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots, y_j$
  - ▷ cost of opt alignment of $i$ and $j$ length prefixes of $X$ and $Y$

We want to find $\text{OPT}(m, n)$ and $D(m, n)$

# Computing Optimal Alignment

Argue about structure of the optimal alignment (can't compute it)

In $\text{OPT}(m, n)$

- Either $x_m$ is paired with $y_n$
- Or $y_n$ is unpaired and $x_m$ may be paired with $y_p$, $p < n$
- Or $x_m$ is unpaired and $y_n$ may be paired with $x_q$, $q < m$
- Or $x_m$ is paired with $y_p$, $p < n$ and $y_n$ is paired with $x_q$, $q < m$

| | |
|---|---|
| $x_1 \; x_2 \; \cdots \; x_{m-1}$ | $x_m$ |
| $y_1 \; y_2 \; \cdots \; y_{n-1}$ | $y_n$ |

| | |
|---|---|
| $x_1 \; x_2 \; \cdots \; x_m$ | — |
| $y_1 \; y_2 \; \cdots \; y_{n-1}$ | $y_n$ |

| | |
|---|---|
| $x_1 \; x_2 \; \cdots \; x_{m-1}$ | $x_m$ |
| $y_1 \; y_2 \; \cdots \; y_n$ | — |

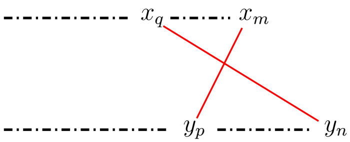
# Computing Optimal Alignment

Argue about structure of the optimal alignment (can't compute it)

In $\mathrm{OPT}(m, n)$

- Either $x_m$ is paired with $y_n$
- Or $y_n$ is unpaired and $x_m$ may be paired with $y_p$, $p < n$
- Or $x_m$ is unpaired and $y_n$ may be paired with $x_q$, $q < m$
- Or $x_m$ is paired with $y_p$, $p < n$ and $y_n$ is paired with $x_q$, $q < m$

Since $p < n$ and $q < m$ the pairs $(x_m, y_p)$ and $(x_q, y_n)$ cross



# Computing Optimal Alignment

- **Either $x_m$ is paired with $y_n$**

- $\alpha(x_m, y_n)$ added in $D(m, n)$
- $\text{OPT}(m, n) \setminus \{(x_m, y_n)\}$: some alignment of $x_1, \ldots, x_{m-1}$ and $y_1, \ldots, y_{n-1}$
- $\text{OPT}(m, n) \setminus \{(x_m, y_n)\}$: an optimal alignment of $x_1, \ldots, x_{m-1}$ and $y_1, \ldots, y_{n-1}$
- Because if $\mathcal{A}$ is a better alignment of $x_1, \ldots, x_{m-1}$ and $y_1, \ldots, y_{n-1}$

$$cost(\mathcal{A}) < D(m, n) - \alpha(x_m, y_n) \implies cost(\mathcal{A} \cup \{(x_m, y_n)\}) < D(m, n)$$

- Hence $\mathcal{A} \cup \{(x_m, y_n)\}$ is a better alignment of $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$

| | | | | |
|---|---|---|---|---|
| $x_1$ | $x_2$ | $\cdots$ | $x_{m-1}$ | $x_m$ |
| $y_1$ | $y_2$ | $\cdots$ | $y_{n-1}$ | $y_n$ |

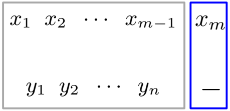
# Computing Optimal Alignment

- **Or $x_m$ is paired with $y_p$, $p < n$ and $y_n$ is unpaired**
  - $\delta$ added in $D(m,n)$
  - $\text{OPT}(m,n)$ is some alignment of $x_1,\ldots,x_m$ and $y_1,\ldots,y_{n-1}$
  - $\text{OPT}(m,n)$ is an optimal alignment of $x_1,\ldots,x_m$ and $y_1,\ldots,y_{n-1}$
  - A better alignment $\mathcal{A}$ of $x_1,\ldots,x_m$ and $y_1,\ldots,y_{n-1}$ contradicts optimality

| | | | | |
|---|---|---|---|---|
| $x_1$ | $x_2$ | $\cdots$ | $x_m$ | $-$ |
| $y_1$ | $y_2$ | $\cdots$ | $y_{n-1}$ | $y_n$ |

# Computing Optimal Alignment

- **Or $y_n$ is paired with $x_q$, $q < m$ and $x_m$ is unpaired**
- $\delta$ added in $D(m, n)$
- $\text{OPT}(m, n)$ is some alignment of $x_1, \ldots, x_{m-1}$ and $y_1, \ldots, y_n$
- $\text{OPT}(m, n)$ is an optimal alignment of $x_1, \ldots, x_{m-1}$ and $y_1, \ldots, y_n$



# Computing Optimal Alignment

$$D(i,j) = \begin{cases} D(i-1,j-1) + \alpha(x_i, y_j) & \text{if } (x_i, y_j) \in \textsc{opt}(i,j) \\ D(i-1,j) + \delta & \text{if } x_i \text{ is unpaired in } \textsc{opt}(i,j) \\ D(i,j-1) + \delta & \text{if } y_j \text{ is unpaired in } \textsc{opt}(i,j) \\ j \cdot \delta & \text{if } i = 0 \\ i \cdot \delta & \text{if } j = 0 \end{cases}$$

We do not know which branch to take (cannot evaluate the conditions)

$$\text{For } i,j \geq 1 \quad D(i,j) = \max \begin{cases} D(i-1,j-1) + \alpha(x_i, y_j) \\ D(i-1,j) + \delta \\ D(i,j-1) + \delta \end{cases}$$

# $D$ matrix computation

**Algorithm** Bottom-Up Evaluation of $D$

```
function COMPUTED(i, j)
    if i = 0 AND j = 0 then
        return 0
    if i = 0 then
        return j × δ
    if j = 0 then
        return i × δ
    pairIt ← COMPUTED(i − 1, j − 1) + α(x_i, y_j)
    addGapX ← COMPUTED(i − 1, j) + δ
    addGapY ← COMPUTED(i, j − 1) + δ
    return max{pairIt, addGapX, addGapY}
```

# Backtracking for pairwise alignment

To find the optimal alignment itself, backtrack from $D(n, m)$ to $D(0, 0)$, following the path of min score at each step

Append the corresponding symbols or gaps to the alignment

**Algorithm** Backtracking to get Optimal Alignment

$X' \leftarrow ""$ $\quad Y' \leftarrow ""$

$i \leftarrow n \quad j \leftarrow m$ ▷ Start from the bottom-right corner of the matrix

**while** $i > 0$ OR $j > 0$ **do**

  **if** $D(i,j) = D(i-1,j-1) + \alpha(x_i, y_j)$ **then**

    $X' \leftarrow x_i + X' \quad Y' \leftarrow y_i + Y' \quad i \leftarrow i-1 \quad j \leftarrow j-1$

  **else if** $D(i,j) = D(i-1,j) + \delta$ **then**

    $X' \leftarrow "-" + X' \quad Y' \leftarrow y_i + Y' \quad i \leftarrow i-1$

  **else**

    $X' \leftarrow x_i + X' \quad Y' \leftarrow "-" + Y' \quad j \leftarrow j-1$

  **return** $X'$ , $Y'$

# Pairwise alignment: Example

Let $X = ATGCT$ and $Y = AGCT$

Using the penalty and rewards: Match = +1, Mismatch = -1, Gap = -2
The matrix $D$ of order $(n+1) * (m+1)$ is filled as follows:

| | - | A | T | G | C | T |
|---|---|---|---|---|---|---|
| - | 0 | -2 | -4 | -6 | -8 | -10 |
| A | -2 | ? | ? | ? | ? | ? |
| G | -4 | | | | | |
| C | -6 | | | | | |
| T | -8 | | | | | |

Initialization:

- $D(0,0) = 0$
- $D(i,0) = D(i-1,0) - \delta$
- $D(0,j) = D(0,j-1) - \delta$

# Pairwise alignment: Example

Match = +1, Mismatch = -1, Gap = -2

The second row of $D$ is filled as follows:

Update rule:

| | - | A | T | G | C | T |
|---|---|---|---|---|---|---|
| - | 0 | -2 | -4 | -6 | -8 | -10 |
| A | -2 | 1 | -1 | -3 | -5 | -7 |
| G | -4 | ? | ? | ? | ? | ? |
| C | -6 | | | | | |
| T | -8 | | | | | |

$$D(1,1) = \max \begin{cases} D(0,0) + \alpha(A,A) \\ D(0,1) + \delta \\ D(1,0) + \delta \end{cases}$$

$$= \max \begin{cases} 0 + 1 \\ -2 + (-2) \\ -2 + (-2) \end{cases} = 1$$

# Pairwise alignment: Example

Let $X = ATGCT$ and $Y = AGCT$

The matrix $D$ is filled as follows:

| | - | A | T | G | C | T |
|---|---|---|---|---|---|---|
| - | 0 | -2 | -4 | -6 | -8 | -10 |
| A | -2 | 1 | -1 | -3 | -5 | -7 |
| G | -4 | -1 | 0 | 0 | -2 | -4 |
| C | -6 | -3 | -2 | -1 | 1 | -2 |
| T | -8 | -5 | -2 | -3 | -2 | 2 |

Termination:

$D(4,5) = 2$

# Pairwise alignment: Backtracking Example

The optimal alignment score is $D(4,5) = 2$ and is obtained by backtracking from $D(4,5)$ to $D(0,0)$

The traceback matrix is filled as below:

| | - | A | T | G | C | T |
|---|---|---|---|---|---|---|
| - | . | . | . | . | . | . |
| A | . | $\nwarrow$ | $\leftarrow$ | $\leftarrow$ | $\leftarrow$ | $\leftarrow$ |
| G | . | $\uparrow$ | $\nwarrow$ | $\nwarrow$ | $\leftarrow$ | $\leftarrow$ |
| C | . | $\uparrow$ | $\nwarrow\uparrow$ | $\nwarrow$ | $\nwarrow$ | $\leftarrow$ |
| T | . | $\uparrow$ | $\nwarrow$ | $\nwarrow\uparrow$ | $\nwarrow\uparrow$ | $\nwarrow$ |

- Diag $\nwarrow$: the letters from two sequences are aligned/paired
- Left $\leftarrow$: gap is introduced into the left sequence
- Up $\uparrow$: a gap is introduced into the top sequence

The optimal alignment is:

| | | | | |
|---|---|---|---|---|
| $A$ | $T$ | $G$ | $C$ | $T$ |
| $A$ | $-$ | $G$ | $C$ | $T$ |

# Sequence Alignment: Variations

- Local/Global Alignment
- Multiple Sequence Alignment
- Short Read Alignment
- Gap Penalties

This, and heuristic approximations to it like BLAST, are workhorse tools in molecular biology, and elsewhere.

BLAST Demo `http://www.ncbi.nlm.nih.gov/blast/`

Try it!

pick any protein, e.g. hemoglobin, insulin, exportin,... BLAST to find distant relatives