Dynamic programming

CMPSC 465 - Yana Safonova

Edit distance

Edit distance

Definition

The **edit distance** between x and y, denoted by $d(x, y)$, is the minimum number of insertions, deletions, and substitutions needed to transform x to y

```
X = PLACE
    ..|||
Y = SPACE

dist = 2 (two mismatches)
```

```
X = TOAD-
    |.||
Y = TRADE

dist = 2 (1 mismatch, 1
insertion)
```

Edit distance and alignment

Alignment is a way of arranging DNA sequences to identify regions of similarity that show functional, structural, or evolutionary relationships between the sequences

```
                 0123456789      16
                 ACGAGCGAGTCGCGAG-
distance = 3     |||||||.| ||||||
                 ACGAGCGCG-CGCGAGC
```

```
Mutations:
7: (A, C)
9: (T, -)
16: (-, C)
```

Edit distance and alignment

Alignment is a way of arranging DNA sequences to identify regions of similarity that show functional, structural, or evolutionary relationships between the sequences

```
                 0123456789      16        Mutations:
                 ACGAGCGAGTCGCGAG-         7: (A, C)
distance = 3     ||||||||.| ||||||         9: (T, -)
                 ACGAGCGCG-CGCGAGC         16: (-, C)

                 ACGAGCGAGTCGCGAG
distance = 8     ||||||||.|.......
                 ACGAGCGCGCGCGAGC
```

Edit distance and alignment

Alignment is a way of arranging DNA sequences to identify regions of similarity that show functional, structural, or evolutionary relationships between the sequences

```
0123456789      16         Mutations:
ACGAGCGAGTCGCGAG-          7: (A, C)
|||||||.| ||||||           9: (T, -)
ACGAGCGCG-CGCGAGC          16: (-, C)
```

Edit distance can be viewed as the number of non-matching operations in the best alignment

Edit distance can be computed using DP

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

Edit distance can be computed using DP

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

Subproblem: consider prefix $x_1 \cdots x_i$ and $y_1 \cdots y_j$ $(i \leq m,\ j \leq n)$

Define

$$E(i, j) = d(x_1 \cdots x_i, y_1 \cdots y_j)$$

Edit distance can be computed using DP

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

Subproblem: consider prefix $x_1 \cdots x_i$ and $y_1 \cdots y_j$ $(i \leq m,\ j \leq n)$

Define

$$E(i,j) = d(x_1 \cdots x_i, y_1 \cdots y_j)$$

Optimal solution: $E(m,n)$

How to use the solution to the subproblems to solve $E(i,j)$?

Edit distance recurrence

Look at the rightmost column:

Edit distance recurrence

Look at the rightmost column:

Case 1 $\begin{array}{cccc} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{array}$

Contributes 1 to the cost plus the cost of alignment $\begin{array}{ccc} x_1 & \cdots & x_{i-1} \\ y_1 & \cdots & y_j \end{array}$

$$E(i, j) = 1 + E(i-1, j)$$

Edit distance recurrence

Look at the rightmost column:

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{matrix}$

Contributes 1 to the cost plus the cost of alignment $\begin{matrix} x_1 & \cdots & x_{i-1} \\ y_1 & \cdots & y_j \end{matrix}$

$$E(i,j) = 1 + E(i-1, j)$$

Case 2 $\begin{matrix} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{matrix}$

Contributes 1 to the cost plus the cost of alignment $\begin{matrix} x_1 & \cdots & x_i \\ y_1 & \cdots & y_{j-1} \end{matrix}$

$$E(i,j) = 1 + E(i, j-1)$$

Edit distance recurrence

Look at the rightmost column:

Case 1
$$\begin{array}{cccc} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{array}$$

Contributes 1 to the cost plus the cost of alignment $\begin{array}{ccc} x_1 & \cdots & x_{i-1} \\ y_1 & \cdots & y_j \end{array}$

$$E(i,j) = 1 + E(i-1,j)$$

Case 2
$$\begin{array}{cccc} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{array}$$

Contributes 1 to the cost plus the cost of alignment $\begin{array}{ccc} x_1 & \cdots & x_i \\ y_1 & \cdots & y_{j-1} \end{array}$

$$E(i,j) = 1 + E(i,j-1)$$

Case 3
$$\begin{array}{cccc} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_{j-1} & y_j \end{array}$$

$$E(i,j) = \begin{cases} E(i-1,j-1) & \text{if } x_i = y_j \\ 1 + E(i-1,j-1) & \text{otherwise} \end{cases}$$

Edit distance recurrence

The recurrence:

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

where

$$\mathrm{diff}(i,j) = \begin{cases} 1 & \text{if } x_i \neq y_j \\ 0 & \text{otherwise} \end{cases}$$

Edit distance recurrence

The recurrence:

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

where

$$\mathrm{diff}(i,j) = \begin{cases} 1 & \text{if } x_i \neq y_j \\ 0 & \text{otherwise} \end{cases}$$

Optimal solution: $E(m,n)$

Base case: $E(0,0) = 0$, $E(i,0) = i$, $E(0,j) = j$

Edit distance table

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

Edit distance table: filling base cases

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

$$\begin{array}{ccccc} E(0,0) & E(0,1) & \cdots & E(0,n-1) & E(0,n) \\ E(1,0) & & & & \\ \vdots & & & & \\ E(m-1,0) & & & & \\ E(m,0) & & & & \end{array}$$

m + 1 rows
n + 1 columns

Edit distance table: computing E(1, 1)

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

$$\begin{array}{ccccc}
E(0,0) & E(0,1) & \cdots & E(0,n-1) & E(0,n) \\
 & \downarrow & & & \\
E(1,0) & E(1,1) & & & \\
\vdots & & & & \\
E(m-1,0) & & & & \\
E(m,0) & & & &
\end{array}$$

E(1, 1) = min{
1 + E(0, 1),

}

Edit distance table: computing E(1, 1)

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

$$\begin{array}{ccccccc}
E(0,0) & & E(0,1) & \cdots & E(0,n-1) & & E(0,n) \\
 & & \downarrow & & & & \\
E(1,0) & \rightarrow & E(1,1) & & & & \\
\vdots & & & & & & \\
E(m-1,0) & & & & & & \\
E(m,0) & & & & & &
\end{array}$$

E(1, 1) = min{
1 + E(0, 1),
1 + E(1, 0),

}

Edit distance table: computing E(1, 1)

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \text{diff}(i,j) + E(i-1,j-1)\},$$

$$\begin{array}{ccccc}
E(0,0) & & E(0,1) & \cdots & E(0,n-1) & E(0,n) \\
& \searrow & \downarrow & & & \\
E(1,0) & \rightarrow & E(1,1) & & & \\
\vdots & & & & & \\
E(m-1,0) & & & & & \\
E(m,0) & & & & &
\end{array}$$

E(1, 1) = min{
1 + E(0, 1),
1 + E(1, 0),
diff(1, 1) + E(0, 0)
}

Edit distance table: completing row 1

$$E(i,j) = \min\{1 + E(i-1,j), 1 + E(i,j-1), \mathrm{diff}(i,j) + E(i-1,j-1)\},$$

$$\begin{tikzcd}
E(0,0) \arrow[dr] & E(0,1) \arrow[d] & \cdots \arrow[dr] & E(0,n-1) \arrow[d] \arrow[dr] & E(0,n) \arrow[d] \\
E(1,0) \arrow[r] & E(1,1) & \cdots \arrow[r] & E(1,n-1) \arrow[r] & E(1,n)
\end{tikzcd}$$

Edit distance table: the complete version

$$E(i, j) = \min\{1 + E(i-1, j), 1 + E(i, j-1), \mathrm{diff}(i, j) + E(i-1, j-1)\},$$

$$\begin{array}{cccccccc}
E(0,0) & & E(0,1) & \cdots & & E(0,n-1) & & E(0,n) \\
 & \searrow & \downarrow & & & & & \\
E(1,0) & \rightarrow & E(1,1) & \cdots & & \cdots & & \cdots \\
\vdots & & & & & & & \\
 & & & & \searrow & \downarrow & \searrow & \downarrow \\
E(m-1,0) & & & & \rightarrow & E(m-1,n-1) & \rightarrow & E(m-1,n) \\
 & & & & \searrow & \downarrow & \searrow & \downarrow \\
E(m,0) & & & & \rightarrow & E(m,n-1) & \rightarrow & E(m,n)
\end{array}$$

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
AC
--
```

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
ACGTA
-----
```

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
ACGTA-
-----A
```

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
ACGTA--
-----AG
```

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

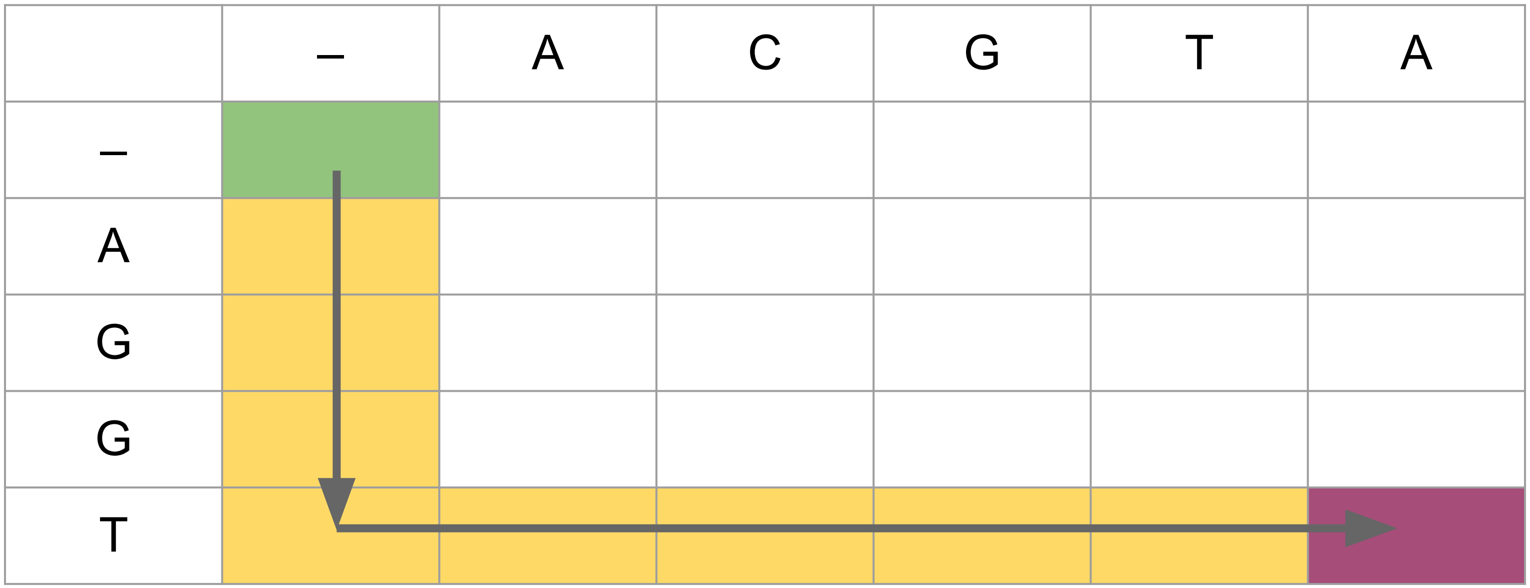
Each path from the green cell to the purple cell corresponds to an alignment

Edit distance = 9

```
ACGTA----
-----AGGT
```

One of two worst alignments in terms of the edit distance

Example



→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

Edit distance = 9

```
----ACGTA
AGGT-----
```

Example

	–	A	C	G	T	A
–						
A						
G						
G						
T						

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

Edit distance = 2

ACGTA
AGGT -

Is it the best alignment?

Example: base cases

Indices / letters		0	1	2	3	4	5
		–	**A**	**C**	**G**	**T**	**A**
0	**–**	0	1	2	3	4	5
1	**A**	1					
2	**G**	2					
3	**G**	3					
4	**T**	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1					
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

min{E(1, 0) + 1, E(0, 0) + d(A, A), E(0, 1) + 1}

Example

Indices / letters		0	1	2	3	4	5
		–	**A**	**C**	**G**	**T**	**A**
0	**–**	0	1	2	3	4	5
1	**A**	1					
2	**G**	2					
3	**G**	3					
4	**T**	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
min{1 + 1, 0 + 0, 1 + 1}
```

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0				
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

min{1 + 1, 0 + 0, 1 + 1}

Example

Indices / letters		0	1	2	3	4	5
		–	**A**	**C**	**G**	**T**	**A**
0	**–**	0	1	2	3	4	5
1	**A**	1	0				
2	**G**	2					
3	**G**	3					
4	**T**	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

-A
A-

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0				
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
-A      A
A-      A
```

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0				
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
-A    A    A-
A-    A    -A
```

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0				
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

A
A

The best choice for aligning prefixes of lengths 1 and 1

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0				
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
min{E(1, 1) + 1, E(0, 1) + d(A, C), E(0, 2) + 1}
```

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0				
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

$$\min\{0 + 1, 1 + 1, 2 + 1\}$$

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1			
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

$$\min\{0 + 1, 1 + 1, 2 + 1\}$$

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1			
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT

↓ - insertion to ACGTA

↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

AC

A-

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	**–**	0	1	2	3	4	5
1	**A**	1	0	1			
2	**G**	2					
3	**G**	3					
4	**T**	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

AC
A-

AC
-A

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1			
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
AC
A-
```

```
AC
-A
```

```
AC-
--A
```

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1			
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

AC
A-

The best choice for aligning prefixes of lengths 2 and 1

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1			
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

```
min{E(1, 2) + 1, E(2, 0) + d(A, G), E(0, 3) + 1}
```

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1			
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

$$\min\{1 + 1,\ 2 + 1,\ 3 + 1\}$$

Example

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1	2		
2	G	2					
3	G	3					
4	T	4					

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

$$\min\{1 + 1,\ 2 + 1,\ 3 + 1\}$$

Example

Indices / letters		0	1	2	3	4	5
		–	**A**	**C**	**G**	**T**	**A**
0	**–**	0	1	2	3	4	5
1	**A**	1	0	1	2	3	4
2	**G**	2	1	1	1	2	3
3	**G**	3	2	2	1	2	3
4	**T**	4	3	3	2	1	2

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

Example

Indices / letters		0	1	2	3	4	5
		–	**A**	**C**	**G**	**T**	**A**
0	**–**	0	1	2	3	4	5
1	**A**	1	0	1	2	3	4
2	**G**	2	1	1	1	2	3
3	**G**	3	2	2	1	2	3
4	**T**	4	3	3	2	1	2

→ - insertion to AGGT
↓ - insertion to ACGTA
↘ - match / mismatch

Each path from the green cell to the purple cell corresponds to an alignment

edit distance between AGGT and ACGTA = 2

Pseudocode

```
def EDIT_DISTANCE(x, y):
    for i = 0, ..., m:
        E(i, 0) = i;
    for j = 0, ..., n:
        E(0, j) = j;
    for i = 1, ..., m:
        for j = 1, ..., n:
            E(i, j) =
             min{1+E(i−1, j), 1+E(i, j−1), diff(i, j)+E(i−1, j−1)};
    return E(m, n);
```

Running time: O(mn)
Memory usage: O(mn)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0	1	2	3	4	5
1	A	1	0	1	2	3	4
2	G	2	1	1	1	2	3
3	G	3	2	2	1	2	3
4	T	4	3	3	2	1	2

Edit distance modification

We use an extra table prev to record where each entry of $E(i,j)$ was coming from:

$$\mathrm{prev}(i,j) = \begin{cases} (i-1,j) & \text{if } E(i,j) = 1 + E(i-1,j) \\ (i,j-1) & \text{if } E(i,j) = 1 + E(i,j-1) \\ (i-1,j-1) & \text{if } E(i,j) = \mathrm{diff}(i,j) + E(i-1,j-1) \end{cases}$$

backtracking

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Edit distance modification

We use an extra table prev to record where each entry of $E(i, j)$ was coming from:

$$\text{prev}(i,j) = \begin{cases} (i-1, j) & \text{if } E(i,j) = 1 + E(i-1,j) \\ (i, j-1) & \text{if } E(i,j) = 1 + E(i,j-1) \\ (i-1, j-1) & \text{if } E(i,j) = \text{diff}(i,j) + E(i-1,j-1) \end{cases}$$

def $\textsc{Pring\_Alignment}(x, y, \text{prev})$**:**

- Set $i = m, j = n$;
- **while** $i \geq 1$ *and* $j \geq 1$**:**
  - **if** $\text{prev}(i,j) = (i-1, j-1)$**:**
    - $\text{print\_back}\binom{y_i}{x_i}$;
    - $i = i - 1,\ j = j - 1$;
  - **if** $\text{prev}(i,j) = (i-1, j)$**:**
    - $\text{print\_back}\binom{-}{x_i}$;
    - $i = i - 1$;
  - **if** $\text{prev}(i,j) = (i, j-)$**:**
    - $\text{print\_back}\binom{y_j}{-}$;
    - $j = j - 1$;

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

Example: how to reconstruct the best alignment?

Indices / letters		0	1	2	3	4	5
		–	A	C	G	T	A
0	–	0 (-, -)	1 (0, 0)	2 (0, 1)	3 (0, 2)	4 (0, 3)	5 (0, 4)
1	A	1 (0, 0)	0 (0, 0)	1 (1, 1)	2 (1, 2)	3 (1, 3)	4 (1, 4)
2	G	2 (1, 0)	1 (1, 1)	1 (1, 1)	1 (1, 2)	2 (2, 3)	3 (2, 4)
3	G	3 (2, 0)	2 (2, 1)	2 (2, 2)	1 (2, 2)	2 (2, 3)	3 (3, 4)
4	T	4 (3, 0)	3 (3, 1)	3 (3, 1)	2 (3, 3)	1 (3, 3)	2 (4, 4)

```
ACGTA
|.||
AGGT-
```