Informatica per le Biotecnologie
Algoritmica Lezione 6

Studiato l'algoritmo per determinare la **edit distance** tra due sequenze, che ricordiamo nell'imagine seguente, esamineremo ora **nuovi problemi di allineamento**

	Ø	L	A	B	B	R	O
Ø	Ø	1	2	3	4	5	6
A	1	1	1	2	3	4	5
L	2	1	2	2	3	4	5
B	3	2	2	2	2	3	4
E	4	3	3	3	3	3	4
R	5	4	4	4	4	3	4
O	6	5	5	5	5	4	3

Ricerca approssimata di un pattern **in un** testo

Il pattern X e il testo Y sono sequenze
di n e m caratteri, con $n << m$

La X può apparire in Y come **sottosequenza** S
di questa, eventualmente con qualche differenza
(la edit distance tra X e S può non essere zero)

La X può apparire in diversi punti di Y
e vogliamo individuarli tutti

Adattiamo l'algoritmo ED:

M[i,j] ora indica la edit distance tra

il prefisso $x_1 x_2 \ldots x_i$ **di X**

e la sottosequeza $y_k y_{k+1} \ldots y_j$ **di Y**

che termina in posizione **j** e minimizza tale distanza (il valore k ≥ 1 non è noto a priori).

X: E A C E D R

```
                    k         j
Y    . . A A B C — D . .
X        E A — C E D R
         1         i
```

M[i,j] = 0

L'unica variazione necessaria nell'algoritmo
ED è inizializzare la riga zero con tutti Ø

$$M[Ø,j]=Ø, \quad \text{per } Ø \leq j \leq m$$

per assegnare edit distance zero a
l'allineamento tra il prefisso vuoto della X
e qualunque prefisso della Y

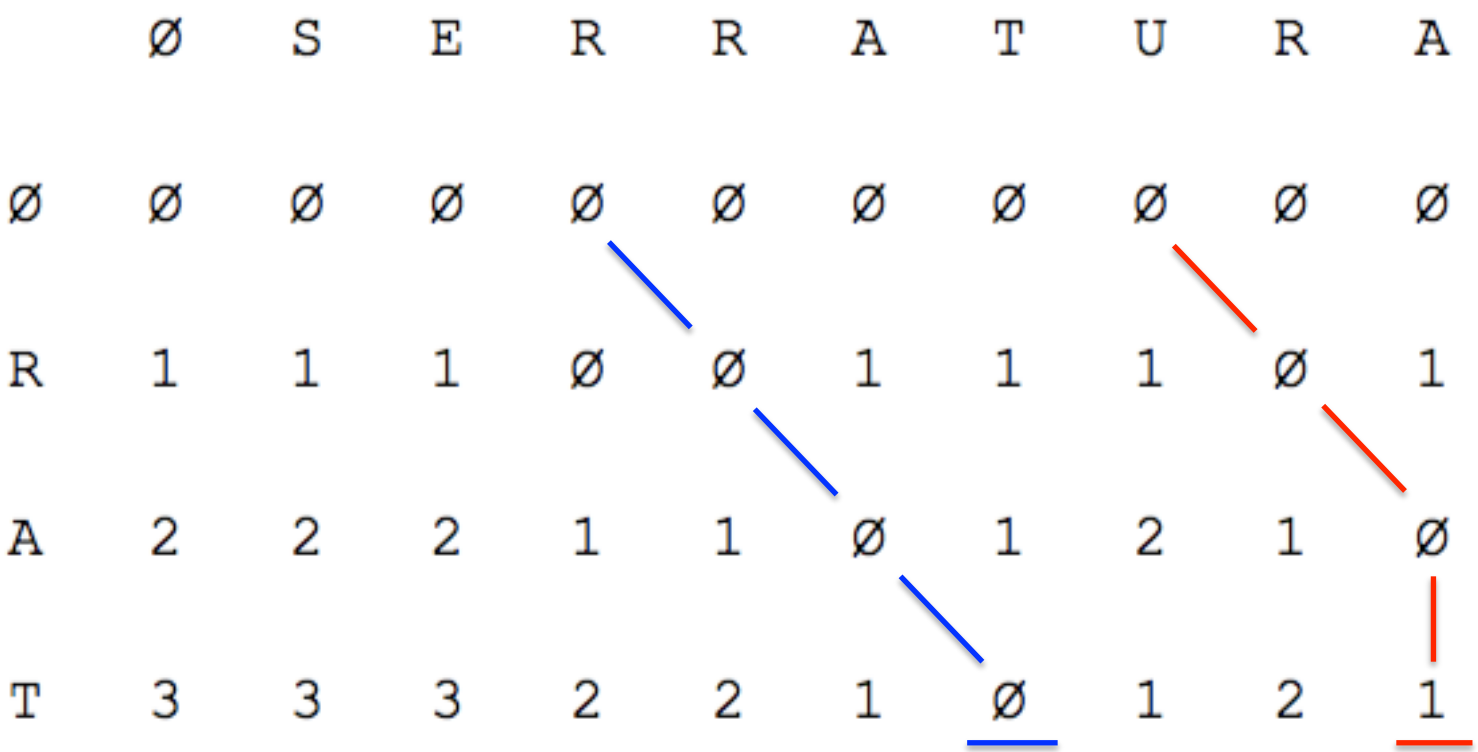
infatti X può apparire in qualsiasi punto di Y

I valori di output sono quelli nell'ultima riga della matrice, che indicano la minima edit distance tra la X e tutte le sottosequenze della Y che terminano in quella colonna

Tra questi valori si sceglieranno i più bassi, o quelli con valore inferiore a un limite prefissato.

		Ø	S	E	R	R	A	T	U	R	A
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
R	1	1	1	Ø	Ø	1	1	1	Ø	1
A	2	2	2	1	1	Ø	1	2	1	Ø
T	3	3	3	2	2	1	Ø	1	2	1

Per determinare gli allineamenti tra la X e le sottosequenze di Y si procede come in ALLINEA, risalendo dalla posizione considerata nell'ultima riga fino alla riga Ø



M[3,6] = 0

Y . . R A T . .
X R A T

M[3,9] = 1

Y . . R A -
X R A T

Studiamo ora gli allineamenti tra sequenze misurandone la qualità con un nuovo parametro detto **similarità**, spesso utilizzato in biologia molecolare al posto della **edit distance**.

Come per la edit distance, la similarità di un allineamento è la somma dei pesi tra coppie di caratteri corrispondenti, ove i pesi sono definiti come:

$P(i,j) = +1$ se $x_i = y_j$ (il match ha segno positivo)

$P(i,j) = -1$ se $x_i \neq y_j$ (mismatch ha segno negativo)

carattere-spazio ha similarità - 1

Due sequenze con data edit distance hanno similarità tanto più alta quanto maggiore è la loro lunghezza, poiché coincidono in un numero maggiore di posizioni e ciascuna di queste dà contributo +1 alla somma.

```
A A A              A A A A A A A A A A A A
B A A              B A A A A A A A A A A A
ed 1 sim 1         ed 1  sim 10
```

Vediamo come si modifica l'algoritmo di EDIT-DISTANCE impiegando la similarità

Problema di Global Comparison

Date due sequenze X e Y trovare un allineamento di massima similarità

All'opposto della edit distance, la similarità è il peso complessivo dell'allineamento che massimizza tale peso

Impieghiamo l'algoritmo di base sulla matrice M già impiegato per la edit distance, con le seguenti modifiche:

M[i,j] indica la similarità tra il prefisso $x_1\ x_2\ \ldots x_i$ di X e il prefisso $y_1\ y_2\ \ldots y_j$ di Y contenenti eventualmente degli spazi.

M[n,m] è la similarità tra le due sequenze.

La riga e la colonna Ø della M corrispondenti a prefissi vuoti di X e Y si inizializzano come:

$M[Ø,j] = -j$ per $Ø \leq j \leq m$

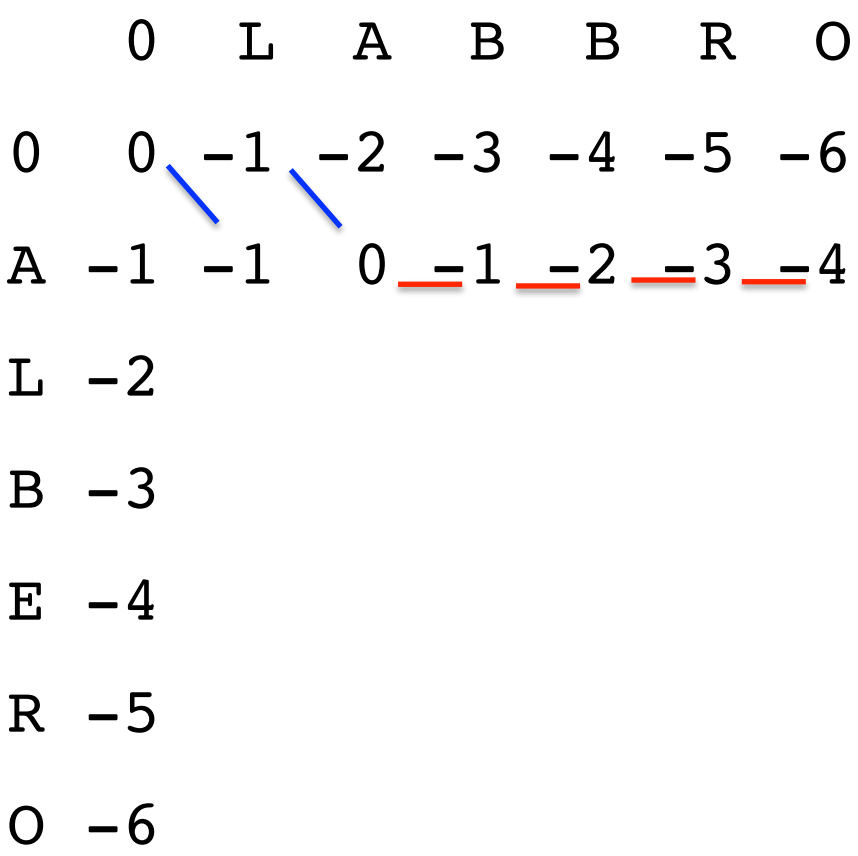
$M[i,Ø] = -i$ per $Ø \leq i \leq n$

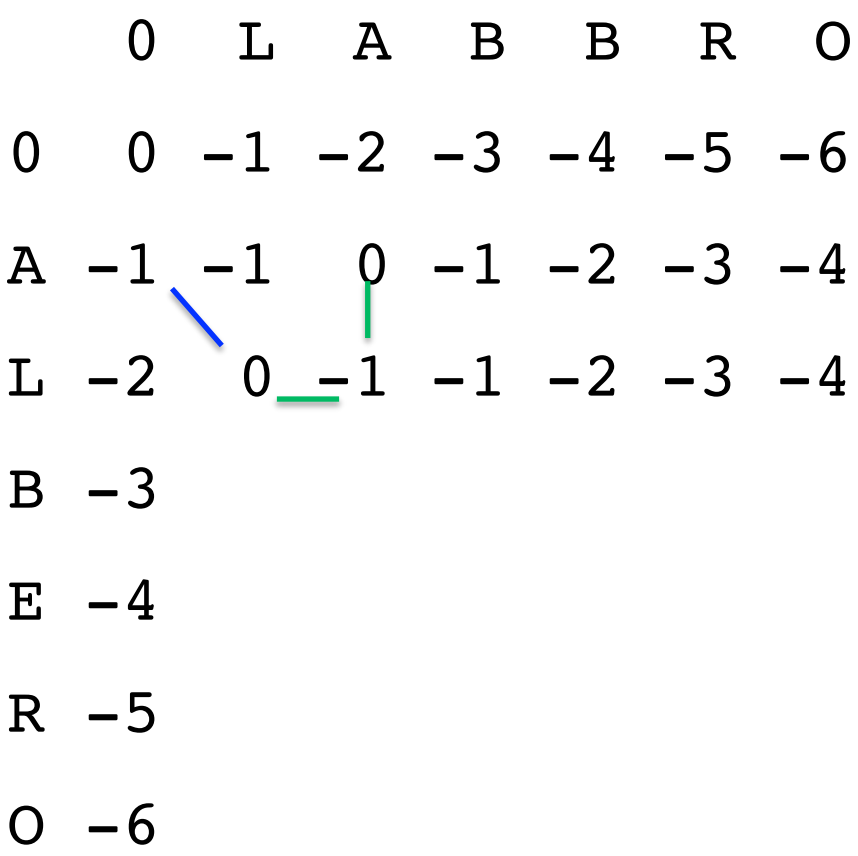
Gli elementi M[i,j] si calcolano con la formula ricorsiva (funzione **max** al posto di **min**):

```
M[i,j] = max{ M[i,j-1] - 1,
              M[i-1,j] - 1,
              M[i-1,j-1] + p(i,j)}
```

La formula può essere immediatamente modificata per diversi valori dei pesi

		0	L	A	B	B	R	O
	0	0	-1	-2	-3	-4	-5	-6
A	-1							
L	-2							
B	-3							
E	-4							
R	-5							
O	-6							


0 L A B B R O
0 0 -1 -2 -3 -4 -5 -6
A -1 -1 0 -1 -2 -3 -4
L -2
B -3
E -4
R -5
O -6


0 L A B B R O
0 0 -1 -2 -3 -4 -5 -6
A -1 -1 0 -1 -2 -3 -4
L -2 0 -1 -1 -2 -3 -4
B -3
E -4
R -5
O -6

		0	L	A	B	B	R	O
	0	0	-1	-2	-3	-4	-5	-6
A	-1	-1	0	-1	-2	-3	-4	
L	-2	0	-1	-1	-2	-3	-4	
B	-3	-1	-1	0	0	-1	-2	
E	-4							
R	-5							
O	-6							

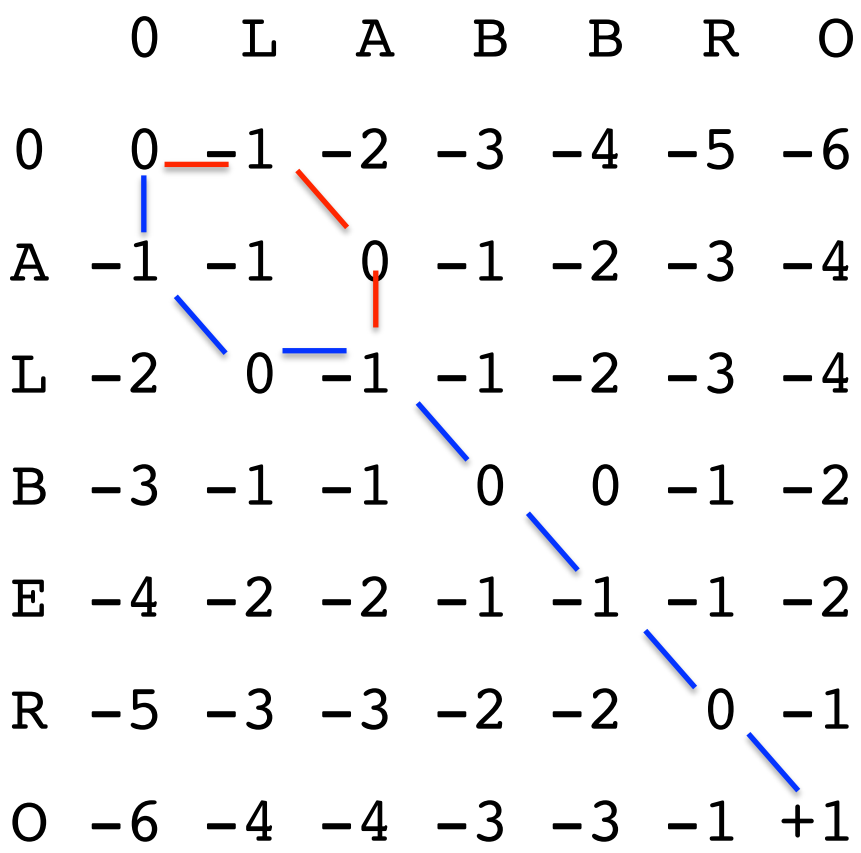
		0	L	A	B	B	R	O
	0	0	-1	-2	-3	-4	-5	-6
A		-1	-1	0	-1	-2	-3	-4
L		-2	0	-1	-1	-2	-3	-4
B		-3	-1	-1	0	0	-1	-2
E		-4	-2	-2	-1	-1	-1	-2
R		-5						
O		-6						

	0	L	A	B	B	R	O
0	0	-1	-2	-3	-4	-5	-6
A	-1	-1	0	-1	-2	-3	-4
L	-2	0	-1	-1	-2	-3	-4
B	-3	-1	-1	0	0	-1	-2
E	-4	-2	-2	-1	-1	-1	-2
R	-5	-3	-3	-2	-2	0	-1
O	-6						

		L	A	B	B	R	O
0	0	-1	-2	-3	-4	-5	-6
A	-1	-1	0	-1	-2	-3	-4
L	-2	0	-1	-1	-2	-3	-4
B	-3	-1	-1	0	0	-1	-2
E	-4	-2	-2	-1	-1	-1	-2
R	-5	-3	-3	-2	-2	0	-1
O	-6	-4	-4	-3	-3	-1	+1

M[6,6] = +1 è la similarità tra le due sequenze

Anche per la global comparison possono esserci diversi allineamenti ottimi che si ricostruiscono con l'algoritmo già descritto per la edit distance risalendo dalla cella M[n,m} alla M[0,0], tenendo conto della nuova regola di calcolo di ogni cella M[i,j]



Due allineamenti con similarità +1

```
A  L  -  B  E  R  O
-  L  A  B  B  R  O
-1 +1 -1 +1 -1 +1 +1
```

```
-  A  L  B  E  R  O
L  A  -  B  B  R  O
-1 +1 -1 +1 -1 +1 +1
```

Questi sono due degli allineamenti già trovati con edit distance = 3

Notare che l' allineamento trovato con edit distance = 3

```
 A  L  B  E  R  O
 L  A  B  B  R  O
-1 -1 +1 -1 +1 +1
```

ha similarità = 0

La similarità è anche impiegata in un problema fondamentale in biologia molecolare che si risolve attraverso la matrice M, con una variazione del solito metodo.

PATTERN COMPARISON. Ricerca con **massima similarità** di una sottosequenza X in una sequenza Y in genere molto più lunga.

L'algoritmo è una ovvia estensione di quello di ricerca di un pattern in un testo ove si sostituisce la similarità alla distanza. Si inizializzano con zero tutti gli elementi della prima riga e con 0, -1, -2, . . . quelli della prima colonna, e si cerca il valore di massima similarità nell'ultima riga.

	Y	A	T	G	T	G	A	C	G	A	A	T	C	A
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	−1	−1	+1	0	+1	0	−1	−1	−1	−1	−1	+1	0	−1
C	−2	−2	0	0	0	0	−1	0	−1	−2	−2	0	+2	+1
A	−3	−1	−1	−1	−1	−1	+1	0	−1	0	−1	−1	+1	+3
G	−4	−2	−2	0	−1	0	−1	0	+1	0	−1	−2	0	+2
A	−5	−3	−3	−1	−1	−1	+1	0	0	+2	+1	0	−1	+1

Massima similarità in M[5,9]

A	T	C	T	G	A	C	G	A	A	T	C	A
			T	C	A	-	G	A				
			+1	−1	+1	−1	+1	+1				

Un nuovo problema di massima importanza:

LOCAL COMPARISON. Tra due sequenze X, Y trovare una sottosequenza S_x di X e una sottosequenza S_y di Y che hanno un allineamento **di massima similarità.**

Il problema sembra molto più difficile dei precedenti perché non si sa nulla a priori sulle due sottosequenze cercate. Tuttavia si risolve attraverso la matrice M, con una variazione del solito metodo.

MA ATTENZIONE: LA VARIAZIONE E' SEMPLICE
MA LA SUA COMPRENSIONE UN PO' MENO !

S_x e S_y devono essere confrontate dal loro inizio, tralasciando i caratteri che le precedono in X in Y. Dunque la prima riga e la prima colonna di M sono inizializzate con tutti zeri.

M[i,j] ora indica la massima similarità tra due sottosequenze che terminano in i, j , cioè
$S_x = x_h \ldots x_i$ e $S_y = y_k \ldots y_j$ ove i valori h, k,
$1 \leq h \leq i$, $1 \leq k \leq j$, rendono massima la similarità.

Come sempre nell'allineamento si possono inserire degli spazi

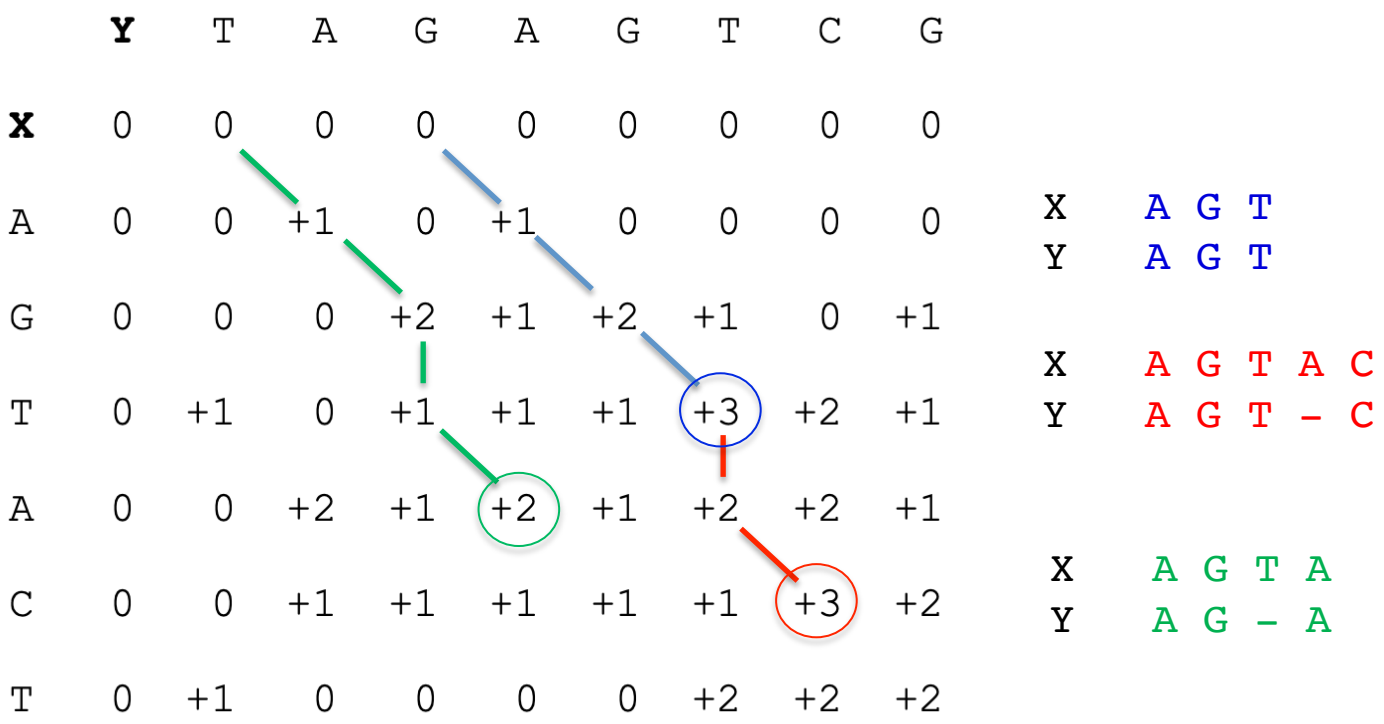
Osservazione chiave:
Se tutte le possibili S_x , S_y che terminano in i , j hanno similarità negativa, si scelgono due sottosequenze vuote che hanno similarità zero.

M[i,j] si calcola con la formula ricorsiva (si noti lo zero tra i valori del massimo):

```
M[i,j] = max{M[i,j-1] - 1,
             M[i-1,j] - 1,
             M[i-1,j-1] + p(i,j),
             0}
```

Il risultato apparirà nella cella M[i,j] che ha un valore rilevante per il caso che si sta studiando (spesso il valore massimo nella matrice), relativa agl'indici i, j di terminazione di S_x , S_y

	Y	T	A	G	A	G	T	C	G
X	0	0	0	0	0	0	0	0	0
A	0								
G	0								
T	0								
A	0								
C	0								
T	0								



La ricostruzione dell'allineamento avverrà con il metodo di tracciamento all'indietro arrestandosi quando si incontra una cella con valore zero, che può trovarsi in qualunque punto della M

Il problema del fragment assembly

Ricostruzione di una sequenza di DNA sulla base della conoscenza di alcuni (moltissimi) **frammenti**.
Il problema nel complesso è detto **sequenziamento**.

Consideriamo sequenze di quattro basi:
A, G - purine, C, T - pirimidine.
Le sequenze sono orientate come nella stuttura della molecola e riferite a uno dei due filamenti.

Gli esperimenti per *leggere* i filamenti di DNA sono in grado di operare su sequenze lunghe fino a qualche migliaio di basi.

Inoltre è tecnicamente impossibile ottenere un intero filamento di DNA senza che questo si rompa in un grande numero di frammenti, anche per organismi la cui intera sequenza è breve.

FRAGMENT ASSEMBLY. Dato un insieme di frammenti di una sequenza, tra loro indipendenti (cioè derivanti da parti distinte della sequenza) o parzialmente sovrapponibili (cioè derivanti da parti interamente o parzialmente sovrapposte), ricostruire la sequenza originale con il minimo numero di errori.

Per eseguire il sequenziamento i frammenti devono essere moltissimi e presentare moltissime sovrapposizioni. Ciò è ottenuto mediante PCR con cui da una molecola di DNA si ottengono repliche in numero che cresce esponenzialmente nel tempo.

Si noti che se tutti i segmenti fossero disgiunti non si avrebbe modo di stabilire in che ordine assembrarli.

Un esempio elementare: quattro brevissimi frammenti

```
T C C G A     C G A C T
A A T C     A T C C G A
```

e l'indicazione che la sequenza complessiva deve contenere circa dieci basi (dunque devono esserci sovrapposizioni di segmenti).

Un possibile allineamento e la risultante sequenza di **consenso**

```
- - T C C G A - -
- - - - C G A C T
A A T C - - - - -
- A T C C G A - -
-----------------
A A T C C G A C T
```

È un caso ideale senza errori.

In genere gli errori sono inevitabili e dipendono dagli strumenti di lettura delle basi e dalla possibile imprecisa replicazione nella PCR.

Errore nel secondo frammento:

C C A C T anziché C G A C T

```
- - T C C G A - -
- - - - C C A C T
A A T C - - - - -
- A T C C G A - -
-----------------
A A T C C G A C T
```

stesso allineamento

G scelta a maggioranza

Errore nel quarto frammento:

A T C G A anziché A T C C G A

(perdita della terza o quarta base)

```
- - T C C G A - -
- - - - C G A C T
A A T C - - - - -
- A T C - G A - -
-----------------
A A T C C G A C T
```

stesso allineamento con spazio nella quarta sequenza e C scelta a maggioranza

Errore nel secondo frammento:

C T G A C T anziché C G A C T

(inserzione nella seconda base)

```
- - T C C - G A - -
- - - - C T G A C T
A A T C - - - - - -
- A T C C - G A - -
-------------------
A A T C C - G A C T
```

diverso allineamento con spazi nella prima e quarta sequenza e spazio nel consenso scelto a maggioranza (ma non vi sono spazi agli estremi del consenso)

Vi sono casi più complessi, a cui solo accenniamo

- Due frammenti provenienti da parti distanti della sequenza originale e parzialmente sovrapponibili, si connettono tra loro dando luogo a un tratto di sequenza detta **chimera**.

Le chimere vanno escluse dalla sequenza ricostruita con tecniche specifiche.

- La sequenza originale può avere tratti ripetuti detti **repeat**. Per esempio nella sequenza **S**:

$$\mathbf{S} = T_1\ a\ b\ c\ T_2\ a\ b\ c\ T_3\ a\ b\ c\ T_4$$

T_1, T_2, T_3, T_4 sono tratti diversi, e **R** = a b c è un repeat composto da tre sotto-sequenze a, b, c.

Da eventuali frammenti

$\mathbf{F_1} = c\ T_2\ a$ $\mathbf{F_2} = c\ T_3\ a$

il consenso può risultare:

$\mathbf{S_1} = T_1\ \mathbf{R}\ T_2\ \mathbf{R}\ T_3\ \mathbf{R}\ T_4$ corretta

$\mathbf{S_2} = T_1\ \mathbf{R}\ T_3\ \mathbf{R}\ T_2\ \mathbf{R}\ T_4$ non corretta

- **Mancanza di copertura**: mancano i frammenti relativi a tratti della sequenza originale.

Si ricostruiscono porzioni disgiunte del consenso

l'unico rimedio è ripetere le operazioni su un maggior numero di copie della sequenza, tenendo presente che i costi complessivi crescono col numero di frammenti costruiti.

Ricostruire la sequenza non è semplice data l'assenza di informazioni sulle zone della sequenza da cui provengono i frammenti e la parziale sovrapposizione tra essi.

Un problema di base, cuore dell'intero processo, si risolve come estensione dei precedenti, in particolare della local comparison.

SUFFIX-PREFIX COMPARISON.

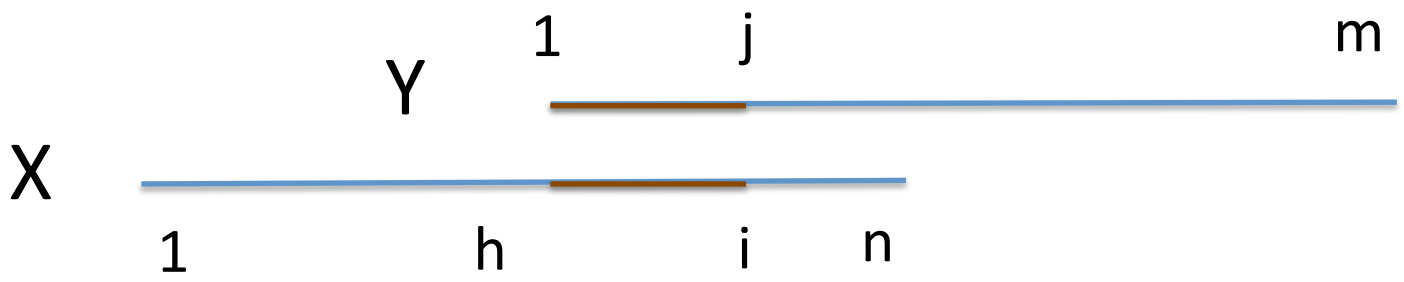
Date due sequenze X,Y trovare un suffisso S_X di X e un prefisso P_y di Y che presentino la **massima similarità.**

Infatti due frammenti si compongono sovrapponendo una parte finale (suffisso) di uno a una parte iniziale (prefisso) del'altro.

Ma non si conosce a priori la lunghezza delle sotto-sequenze S_X e P_y cercate (cioè in quali punti di X e Y inizia la prima e termina la seconda)

Adottiamo di nuovo i pesi p[i,j] = +1 o -1 per match o mismatch, e peso -1 per carattere-spazio, e impieghiamo l'algoritmo standard sulla matrice M con le seguenti variazioni.

- Il valore M[i,j] indica la massima similarità tra un tratto $x_h \ldots x_i$ di X, e un prefisso $y_1 \ldots y_j$ di Y.



Inizializzazione:

$M[\emptyset,j] = -j$, per $\emptyset \leq j \leq m$

possibili spazi iniziali prima di S_X

$M[i,\emptyset] = \emptyset$, per $\emptyset \leq i \leq n$

S_X può iniziare in ogni posizione di X

La formula ricorsiva è quella relativa alla global comparison

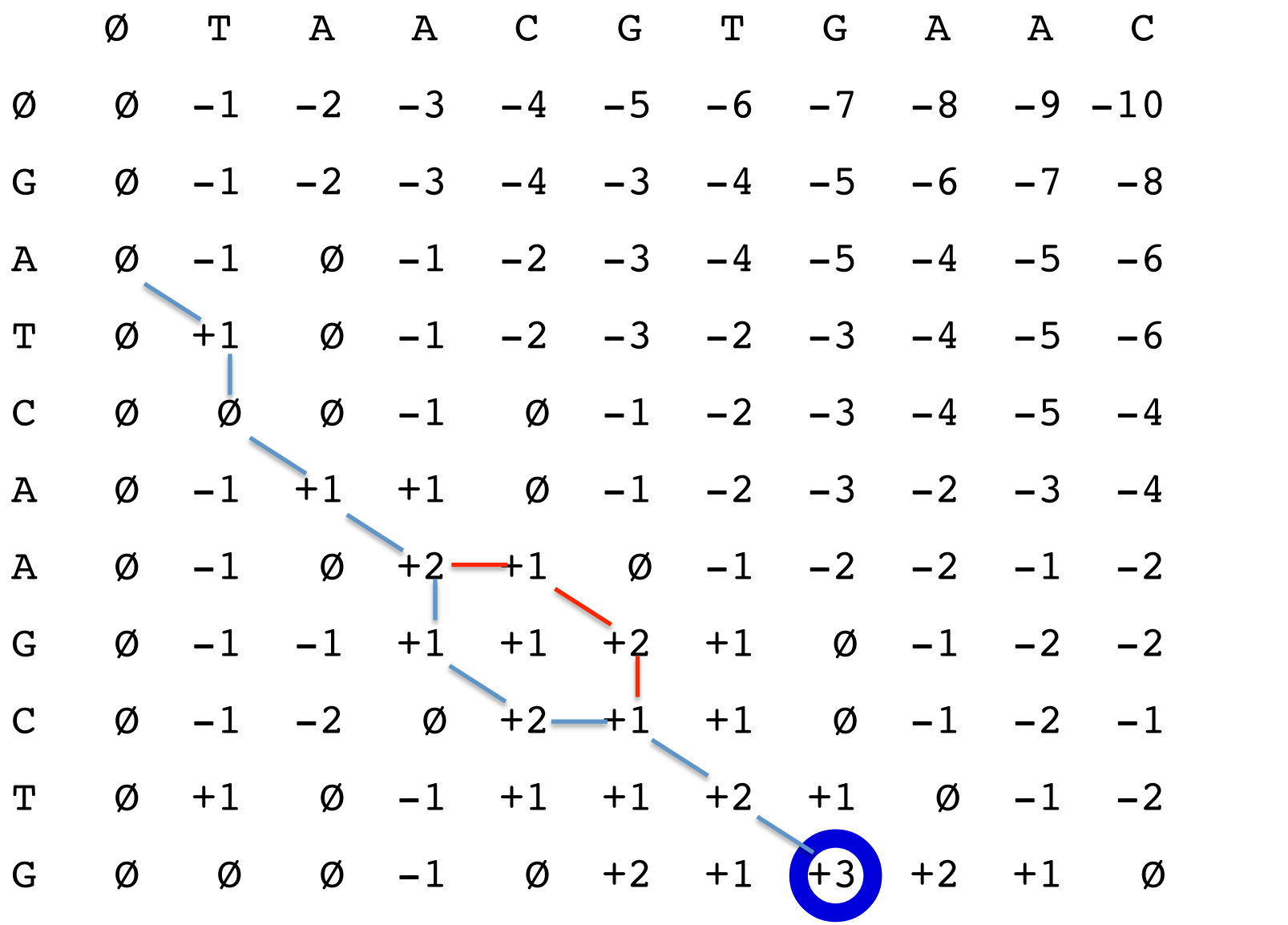
$$M[i,j] = \max\{M[i,j-1]-1,\ M[i-1,j]-1,\ M[i-1,j-1]+p(i,j)\}$$

I risultati si cercano nell'ultima riga della matrice: M[n,j] indica che l'esame di un suffisso S_X è stato completato paragonandolo al prefisso P_y che termina nella colonna j.

L'allineamento si costruisce risalendo nella matrice fino a raggiungere la colonna zero, nella riga ove il prefisso S_X ha inizio, a parte eventuali spazi iniziali.

	Ø	T	A	A	C	G	T	G	A	A	C
Ø	Ø	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
G	Ø	-1	-2	-3	-4	-3	-4	-5	-6	-7	-8
A	Ø	-1	Ø	-1	-2	-3	-4	-5	-4	-5	-6
T	Ø	+1	Ø	-1	-2	-3	-2	-3	-4	-5	-6
C	Ø	Ø	Ø	-1	Ø	-1	-2	-3	-4	-5	-4
A	Ø	-1	+1	+1	Ø	-1	-2	-3	-2	-3	-4
A	Ø	-1	Ø	+2	+1	Ø	-1	-2	-2	-1	-2
G	Ø	-1	-1	+1	+1	+2	+1	Ø	-1	-2	-2
C	Ø	-1	-2	Ø	+2	+1	+1	Ø	-1	-2	-1
T	Ø	+1	Ø	-1	+1	+1	+2	+1	Ø	-1	-2
G	Ø	Ø	Ø	-1	Ø	+2	+1	+3	+2	+1	Ø

M[10,7] = +3


Ø T A A C G T G A A C
Ø Ø -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
G Ø -1 -2 -3 -4 -3 -4 -5 -6 -7 -8
A Ø -1 Ø -1 -2 -3 -4 -5 -4 -5 -6
T Ø +1 Ø -1 -2 -3 -2 -3 -4 -5 -6
C Ø Ø Ø -1 Ø -1 -2 -3 -4 -5 -4
A Ø -1 +1 +1 Ø -1 -2 -3 -2 -3 -4
A Ø -1 Ø +2 +1 Ø -1 -2 -2 -1 -2
G Ø -1 -1 +1 +1 +2 +1 Ø -1 -2 -2
C Ø -1 -2 Ø +2 +1 +1 Ø -1 -2 -1
T Ø +1 Ø -1 +1 +1 +2 +1 Ø -1 -2
G Ø Ø Ø -1 Ø +2 +1 +3 +2 +1 Ø

M[10,7] = +3

```
Y:       T – A A - C G T G A A C
X:   G A T C A A G C - T G
```

```
Y:       T – A A C G – T G A A C
X:   G A T C A A – G C T G
```

Due allineamenti con similarità 3 e lughezza 9

Nelle applicazioni biologiche una sovrapposizione accettabile deve avere in genere lunghezza k e similarità s relativamente alte.

Tra due sovrapposizioni con valori (k_1,s_1) e (k_2,s_2) tali che $k_1 > k_2$ e $s_1 < s_2$ il criterio di scelta è stabilito dal biologo.

Nelle sequenze geniche degli eucarioti, i frammenti possono contenere migliaia di basi e le sovrapposizioni tra loro devono contenere fino a centinaia di basi, con un numero massimo di differenze di qualche unità.