Docente: **Giorgio Valentini**
Istruttore: **Matteo Re**

UNIVERSITÀ DEGLI STUDI DI MILANO

C.d.l. Biotecnologie Industriali e Ambientali

Biologia computazionale

A.A. 2010-2011 semestre II

1 INTRODUZIONE

Note amministrative:

Email: matteo.re@unimi.it
Sito web: put here the link

Inviatemi al più presto una email contenente queste informazioni:

- Nome
- Cognome
- Matricola
- Corso di laurea
- Come oggetto inserite : **corsobiocomp2011**

Motivazioni dell'esistenza della biologia computazionale:

Biologia Computazionale è l'**applicazione** di strumenti propri delle scienze dell'informazione (es. algoritmi, intelligenza artificiale, databases) a problemi di interesse biologico, biotecnologico e biomedico.

Attualmente suscita grande interesse perché:

- La dimensione dei problemi biologici è sufficiente a motivare lo sviluppo di algoritmi efficienti
- I problemi sono accessibili (elevata quantità di dati pubblici e letteratura inerente) ed interessanti
- Le scienze biologiche si avvalgono sempre più spesso di strumenti computazionali

Biologia computazionale: Scienze dell'Informazione o Biologia? (I)

Gli sviluppi delle scienze biomediche, (in particolare per quanto riguarda la biologia molecolare) si verificano ad un ritmo tale da porre seri problemi:

La nostra capacità tecnologica di acquisire nuovi dati (spesso in quantità elevate) rende impossibile la loro analisi **in assenza di strumenti efficienti**.

Biologia computazionale: Scienze dell'Informazione o Biologia? (II)

La biologia computazionale gode di un interesse sempre crescente sia nei dipartimenti di area biomedica che nei dipartimenti di scienze dell'informazione.

- Molti problemi biologici si avvalgono di strumenti informatici (es. assemblaggio di frammenti genomici, analisi di biosequenze, costruzione alberi filogenetici…)

- Molti strumenti biologici sono stati proposti per rispondere a problemi informatici (avete mai pensato che il DNA offre incredibili possibilità di calcolo parallelo? … cercate in google 'DNA COMPUTING')

Attacco al virus SARS

La reazione della comunità scientifica alla comparsa del virus **SARS** in Asia nel **Febbraio 2003** illustra il ruolo critico che genomica e informatica giocano nella biologia moderna.

- **Marzo 2003**: analisi di dati di microarray di DeRisi rivelano che si tratta di un coronavirus.
- **12 Aprile 2003**: Il Michael Smith Genome Sciences Centre in Canada annuncia il sequenziamento del genoma del virus.
- **15 Aprile 2003**: Microarray SARS-specifici vengono messi in commercio.
- **1 Maggio 2003**: Analisi di predizione delle posizioni dei geni SARS ed analisi del genoma SARS pubblicati in Science.
- **Fine Maggio 2003**: Pubblicazione analisi filogenetica per trovare la posizione del virus SARS nella famiglia dei coronavirus.
- **Novembre 2008**: Pubblicato esperimento in cui viene riportata la costruzione di un virus SARS sintetico(**!**)

In questo corso di Biologia Computazionale

Verranno introdotti **PROBLEMI COMPUTAZIONALI** inerenti a:

- Assemblaggio di sequenze
- Predizione dei geni
- Analisi di dati microarray
- Predizione della classe di oggetti biologici mediante integrazione di dati eterogenei
- Predizione della classe farmacologica

e studieremo l'**APPLICAZIONE** degli strumenti disponibili per la soluzione di questi problemi (**NBB:** l'applicazione non è possibile senza il disegno di un esperimento).

‘Natura’ della Biologia Computazionale (I)

Biologia Computazionale è altamente interdisciplinare: questo pone dei problemi.

Informatici vs Biologi/Biotecnologi/Medici…

Esistono vari tipi di scienziati in area biomedica (biologi, biotecnologi, ecologi, medici,…) come in area informatica (algoritmisti, ingegneri del software,…). Questa area di ricerca suscita anche molto interesse da parte di matematici e statistici.

PROBLEMA:

NULLA è completamente vero / falso nelle scienze biomediche mentre **TUTTO** è vero / falso in scienze dell’informazione / matematica.

‘Natura’ della Biologia Computazionale (II)

Esistono delle fondamentali differenze culturali tra informatici, matematici e scienziati di area biomedica:

Biotecnologi/Medici ecc. : cercano di capire un mondo reale estremamente complicato (e sono consapevoli di averne una visione parziale).

Vogliono **SCOPRIRE** qualcosa …

Informatici/Matematici ecc. : cercano di costruire dei ‘micromondi’ organizzati ed ordinati che funzionano da rappresentazioni ridotte del mondo reale. Queste rappresentazioni virtuali del mondo reale sono **INDISPENSABILI** per sviluppare soluzioni efficienti a problemi altrimenti intrattabili.

Vogliono **INVENTARE** qualcosa …

‘Natura’ della Biologia Computazionale (III)

Altre differenze importanti:

Scienziati area Biomedica (BM)	**Scenziati area computazionale o matematica (CM)**	**Conseguenze ?**
Data driven	Algorithm driven	Pagine web CM migliori di pagine web BM
Bagaglio di conoscenze più vasto e più vario (biochimica/genetica/chimica/fisica…)	Bagaglio di conoscenze più specializzato (sono in grado di svolgere operazioni complesse in modo efficiente)	
Sono a loro agio rispetto al fatto che **OGNI DATO** contiene degli errori	Non lo sono	
Costi sperimentali estremamente elevati … (**pianificazione sperimentale accurata**)	Metriche fondamentali per pianificare un esperimento sono stime del **tempo** necessario e **qualità** delle soluzioni prodotte	

Scienze dell'Informazione per scienziati di area biomedica

Cercheremo molto spesso di valutare due aspetti degli algoritmi utilizzati in biologia computazionale:

- **Correttezza**
- **Efficienza**

Descriveremo algoritmi che ritornano, in modo DIMOSTRABILE, la miglior soluzione possibile ad un problema combinatorio BEN DEFINITO. Esistono inoltre altre procedure dette EURISTICHE che restituiscono buoni risultati in problemi pratici ma di cui non è dimostrabile la correttezza.

Una grossa difficoltà è quella di "estrarre" problemi ben definiti da problemi reali complessi in modo da rendere questi ultimi affrontabili. Questo processo è analogo alle sperimentazioni *in-vivo* ed *in-vitro*.

Insidie in esperimenti di Biologia Computazionale

Spesso (se non sempre) gli esperimenti di biologia computazionale sono svolti in parti separate, ognuna affidata ad una categoria differente di figure professionali.

Il rischio maggiore è che le parti svolgano operazioni che non convergano ad una soluzione (o, peggio, che la soluzione finale non risponda ai problemi iniziali).

Possibile soluzione: potenziare i contatti tra le parti in gioco in modo che il team Biomedico formalizzi il problema in esame nel modo più stringente possibile ed il team informatico sia quindi in grado di costruire una soluzione adatta al problema e che permetta di salvaguardare contemporaneamente **correttezza** (quando possibile), **efficienza** e **coerenza** della soluzione proposta con il problema in esame.

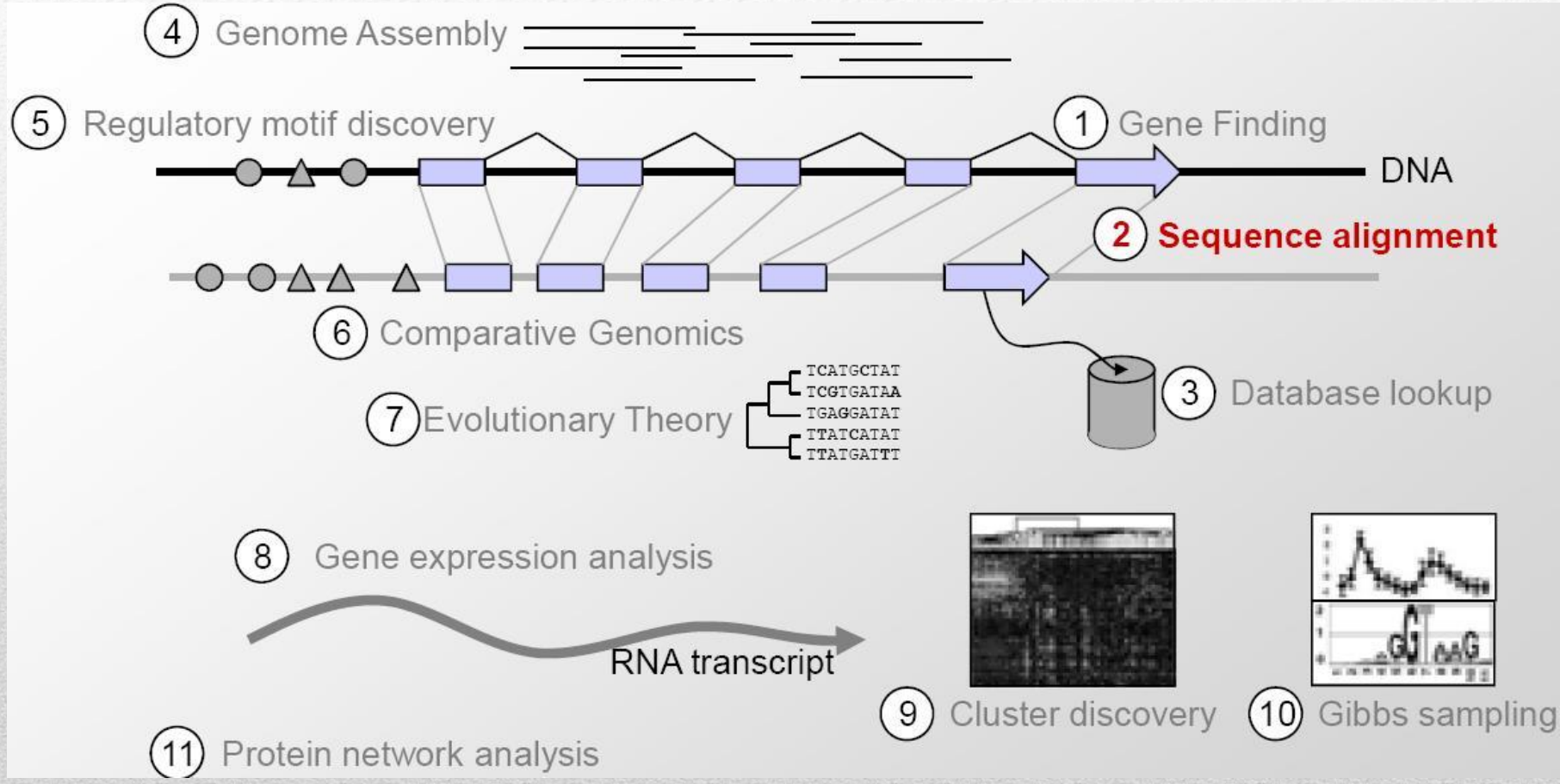
Obiettivi del corso

Presentazione di problemi di interesse biologico/biotecnologico e della loro **formalizzazione**.

Descrizione di algoritmi di uso corrente utilizzati per rispondere ai problemi presentati (analisi delle caratteristiche dei vari algoritmi).

Ricerca sistematica del punto di contatto tra le varie discipline coinvolte nella soluzione dei problemi (**questo è l'obiettivo fondamentale del corso**)

(alcune) Sfide della Biologia Computazionale



Docente: **Giorgio Valentini**
Istruttore: **Matteo Re**

UNIVERSITÀ DEGLI STUDI DI MILANO

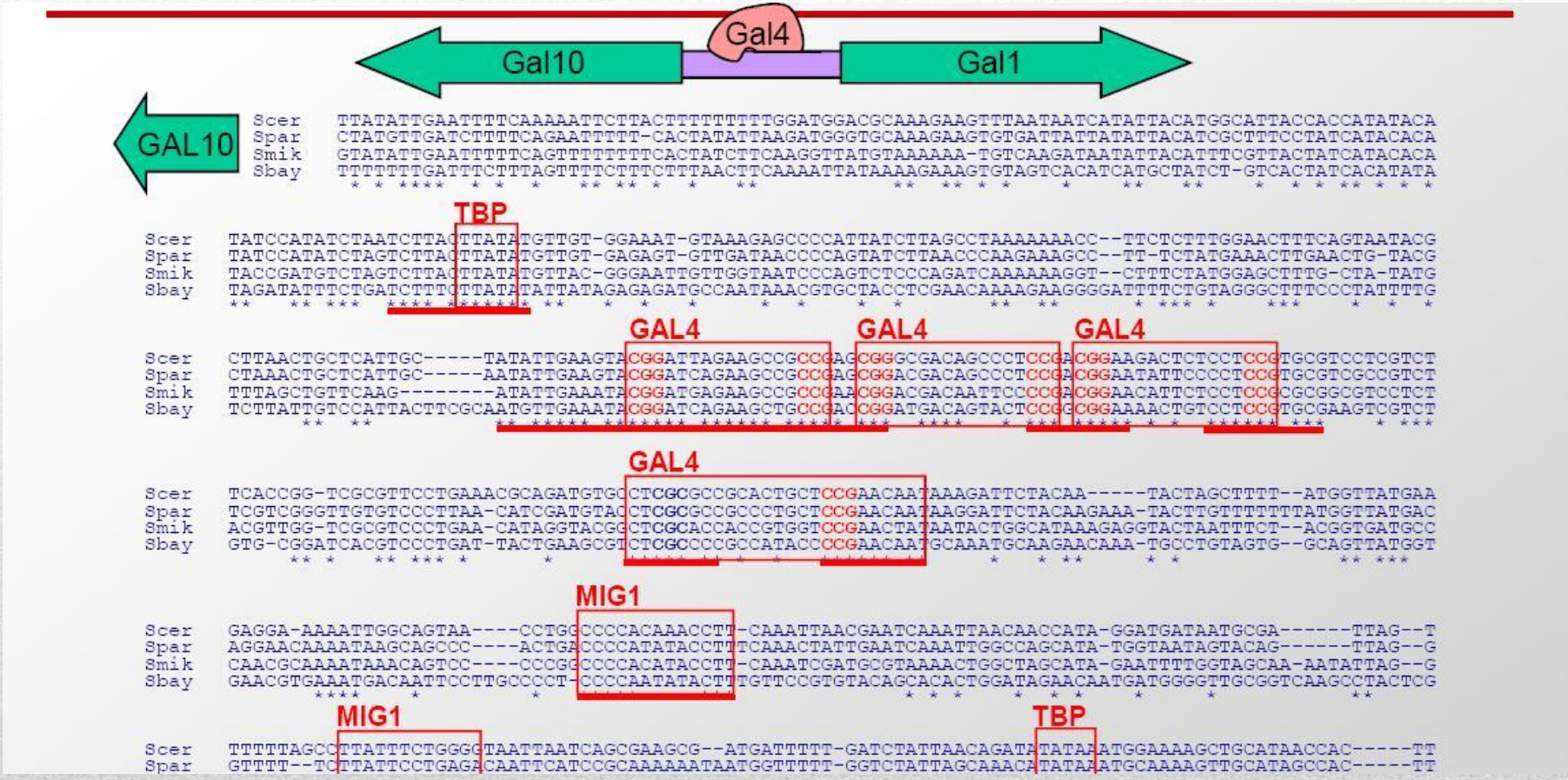
C.d.l. Biotecnologie Industriali e Ambientali

Biologia computazionale

A.A. 2010-2011 semestre II

2 ALLINEAMENTO DI BIOSEQUENZE

Elementi conservati nel corso dell'evoluzione

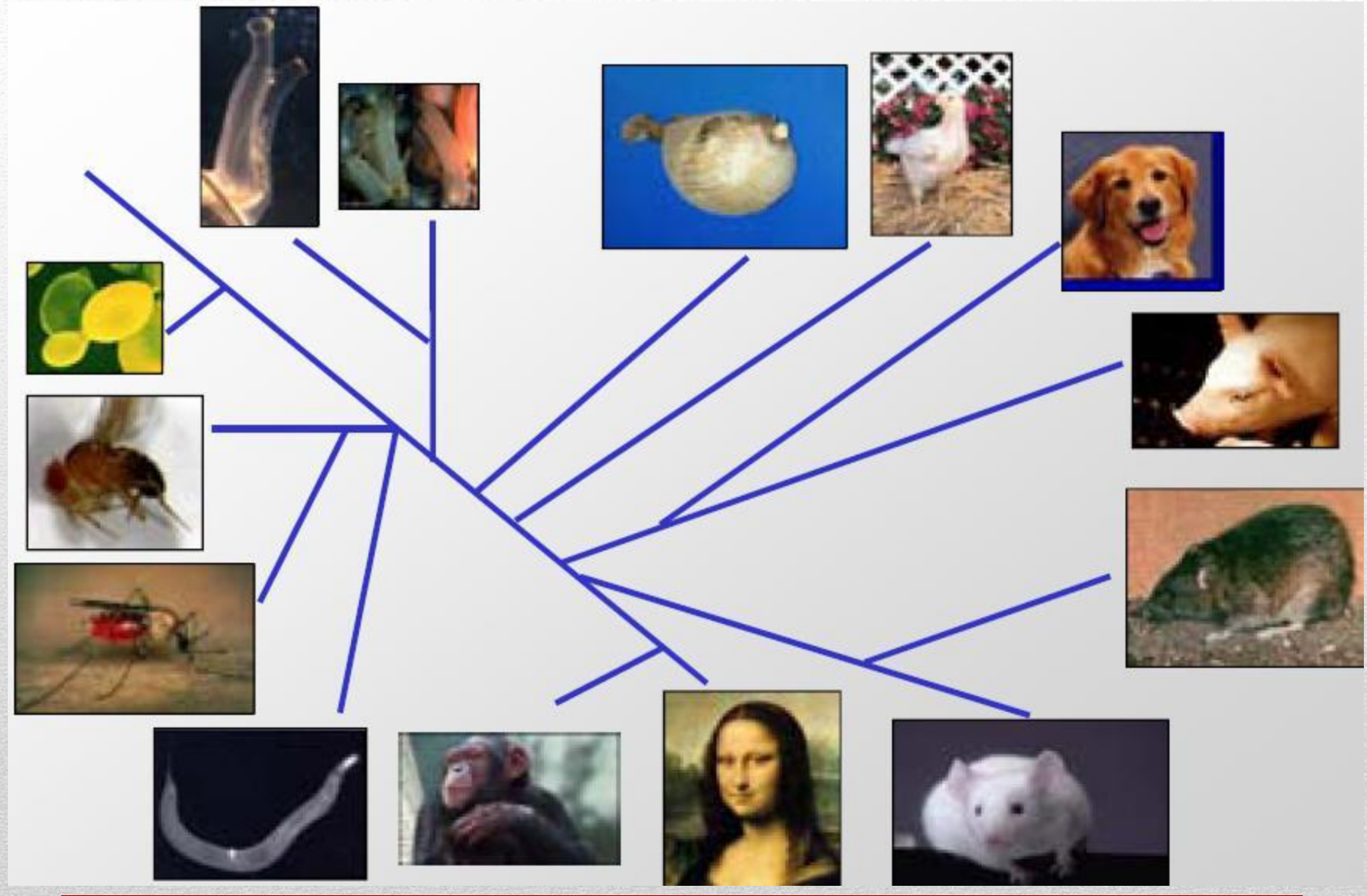


Possiamo **"LEGGERE"** l'evoluzione
per trovare elementi funzionali

Kellis et al. Nature 2003

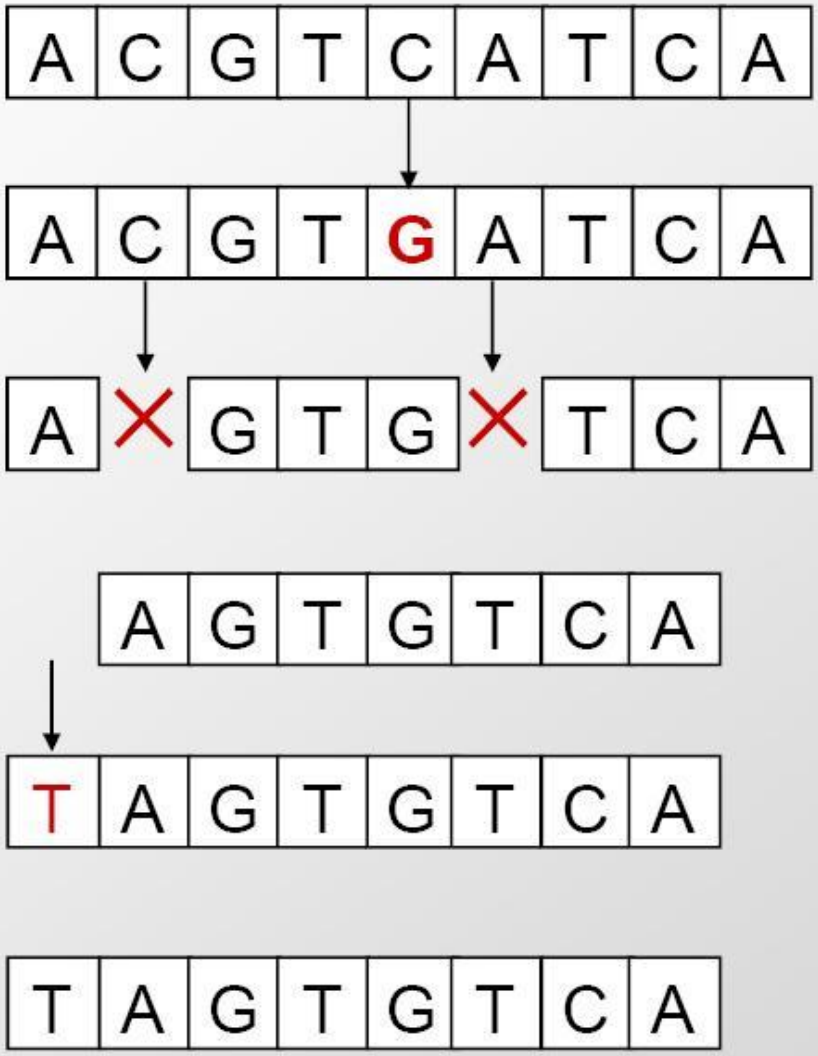
Obiettivo di oggi:

Come possiamo ALLINEARE le sequenze di due geni?



I genomi cambiano nel tempo

Stato iniziale



mutazione

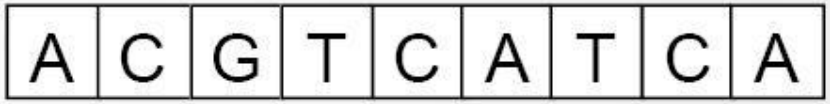
delezione

inserzione

Stato finale

Obiettivo dell'allineamento:

Stato iniziale



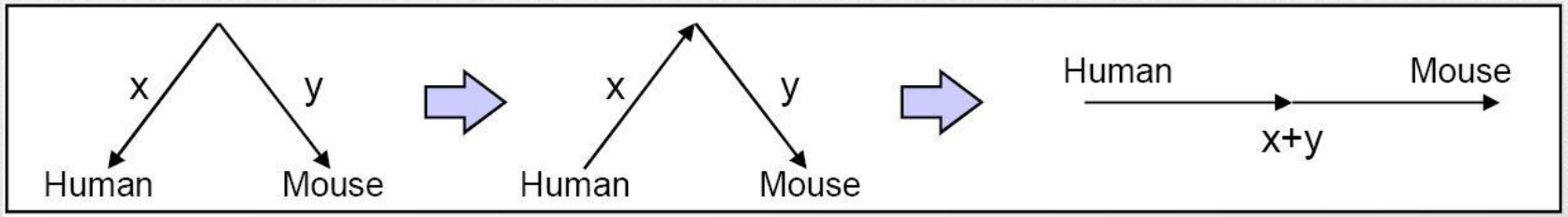
Stato finale

INFERENZA DELLE OPERAZIONI DI MODIFICA (editing)

Bio CS

FORMALIZZAZIONE DEL PROBLEMA

1. Definizione di un **set di operazioni evolutive** (mutazione, inserzione e delezione)



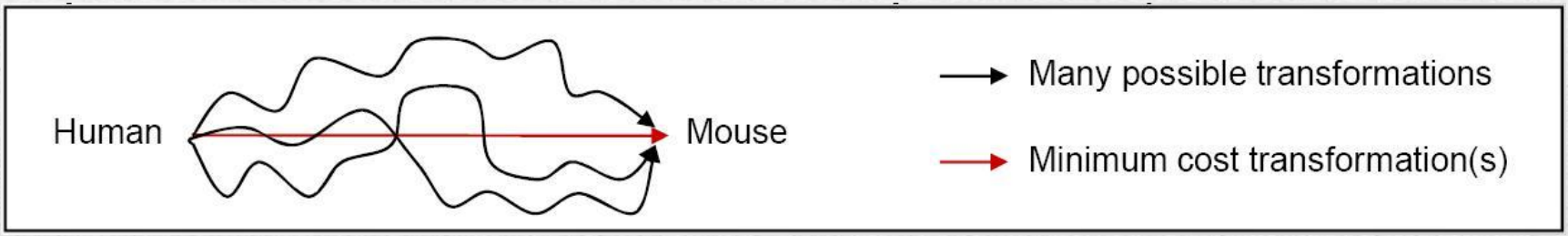
Simmetria delle operazioni (A → C, C → A) permette reversibilità rispetto al tempo. Questa è una scelta di "disegno" della formalizzazione del problema.

Eccezione(**Bio**) : dinucleotidi CpG metilati → TpG/CpA (perdita simmetria)

Bio CS

FORMALIZZAZIONE DEL PROBLEMA

2. Definizione di un **criterio di ottimalità (minimo numero operazioni, minimo costo complessivo,…)**



E’ **IMPOSSIBILE** inferire la serie **ESATTA** di operazioni che portano dalla sequenza A alla sequenza B!

Rasoio di Occam: “Avendo a disposizione diverse ipotesi equivalentemente competitive rispetto ad un dato problema è buona norma scegliere la più semplice e scartare le altre”.

Bio CS

FORMALIZZAZIONE DEL PROBLEMA

3. Definizione di un **algoritmo in grado di <u>raggiungere questa condizione di ottimalità</u>** (o in grado di approssimarla)

Bio

Predicibilità Correttezza

Rilevanza

Casi speciali

Trade-off

CS

Algoritmi

Assunzioni

Implementazione

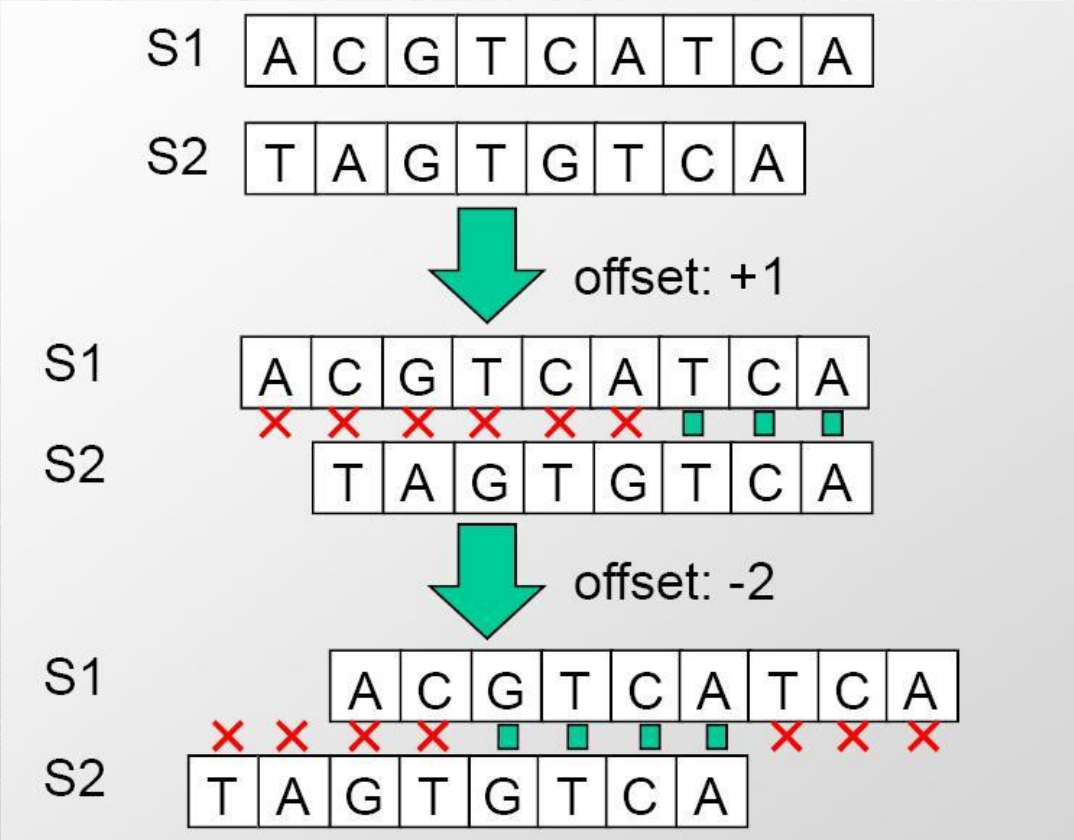
Trattabilità

Computabilità

CS

FORMULAZIONE 1 : longest common subsequence (LCS)

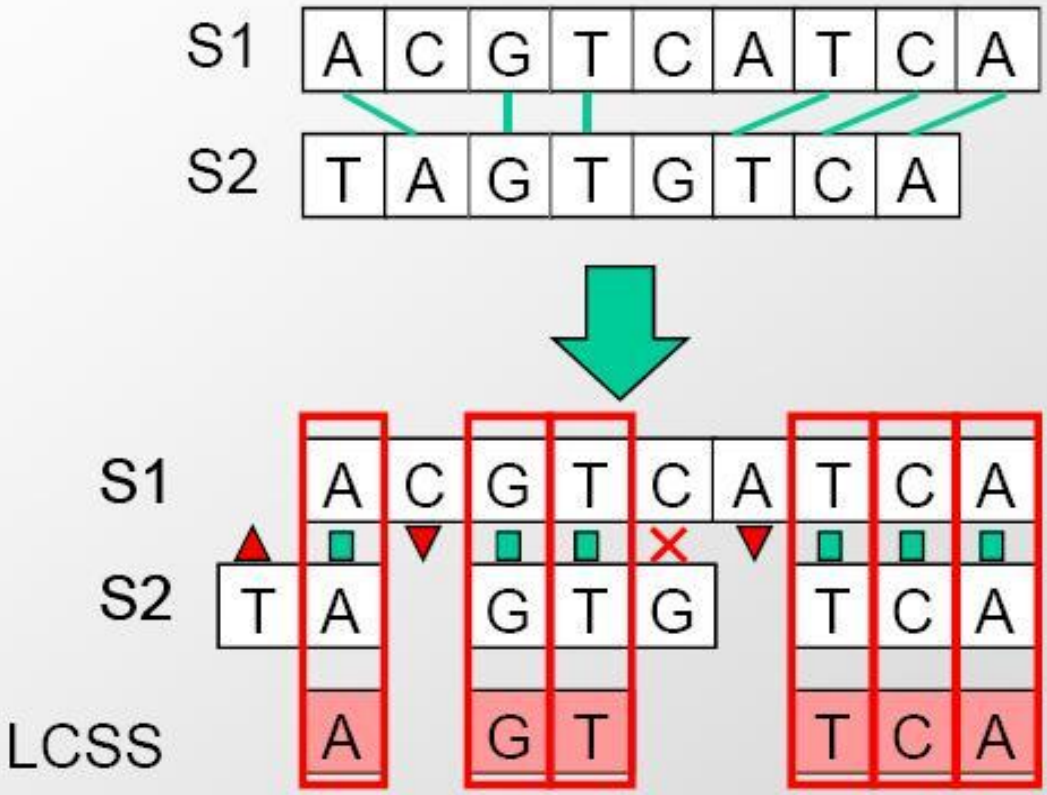
"Date due sequenze S1 ed S2 possibilmente originatesi da una medesima sequenza ancestrale quale è la più lunga sottosequenza in comune (LCS) ?" (gap non ammessi)



CS

FORMULAZIONE 2 : longest common subsequence (LCS)

"Date due sequenze S1 ed S2 possibilmente originatesi da una medesima sequenza ancestrale quale è la più lunga sottosequenza in comune (LCS) ?" (gap ammessi)



E' basata su **edit distance**:

- Numero di cambiamenti per passare da S1 a S2
- Funzione di scoring uniforme

CS

FORMULAZIONE 3 : Allineamento di sequenze

- **Gap ammessi (penalità fissa):**
  - Operazioni di inserzione e delezione
  - Medesimo costo per ogni carattere inserito/deleto

- **Penalità variabili per le operazioni di editing:**
  - Transizioni (Pirimidine↔Pirimidine, Purine↔Purine)
  - Trasversioni (variazioni Pirimidine↔Purine)
  - Polimerasi confonde “relativamente spesso” A/G e C/T

Scoring function:

Match(x,x) = +1

Mismatch(A,G)= -½

Mismatch(C,T)= -½

Mismatch(x,y) = -1

	A	G	T	C
A	+1	-½	-1	-1
G	-½	+1	-1	-1
T	-1	-1	+1	-½
C	-1	-1	-½	+1

purine pyrimid.

Transitions:

A⇔G, C⇔T common (lower penalty)

Transversions:

All other operations

CS

FORMULAZIONE 4,5,6,…

Molte variazioni:

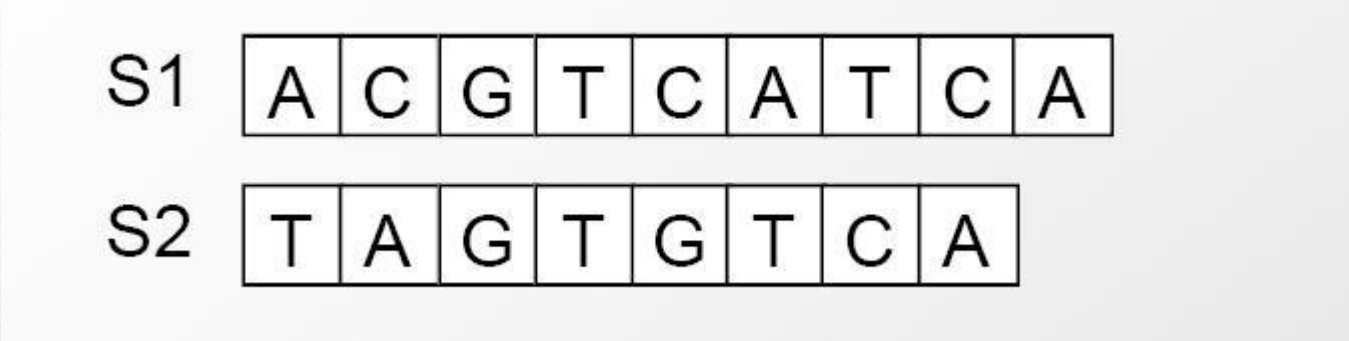
es. Penalità variabili in accordo con il numero di gap

…

(riuscite a proporre delle varianti?)

CS

Come possiamo costruire il “miglior” allineamento?



- **Data una funzione di scoring <u>additiva</u> :**
  - Costo mutazioni (AG, CT, altre)
  - Costo inserzione / delezione
  - Premio per match

- **Serve algoritmo per provare il <u>miglior</u> allineamento:**
  - Enumerazione di <u>tutti</u> i possibili allineamenti?
  - Come la realizzereste?
  - <u>Quanti sono</u> i possibili allineamenti di due sequenze?

CS

Modi di allineare due sequenze di lunghezza m, n:

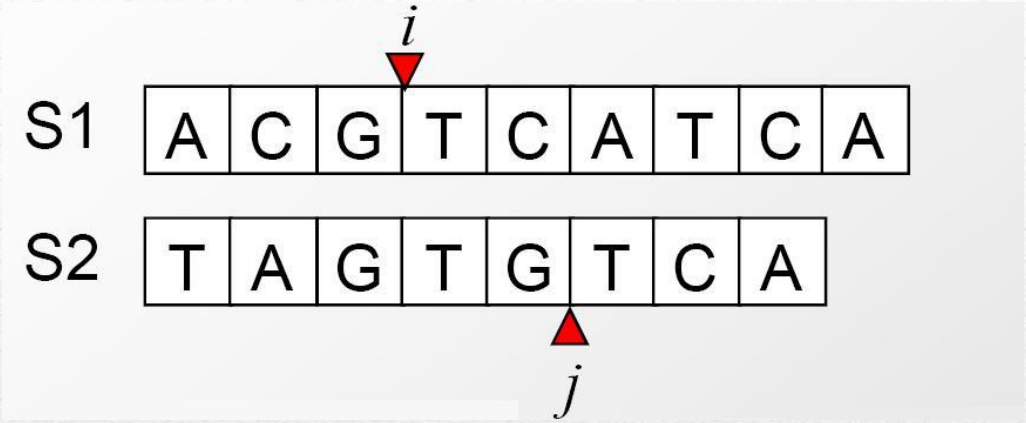
$$\binom{n+m}{m} = \frac{(m+n)!}{(m!)^2} \approx \frac{2^{m+n}}{\sqrt{\pi \cdot m}}$$

- **Per due sequenze di lunghezza n:**

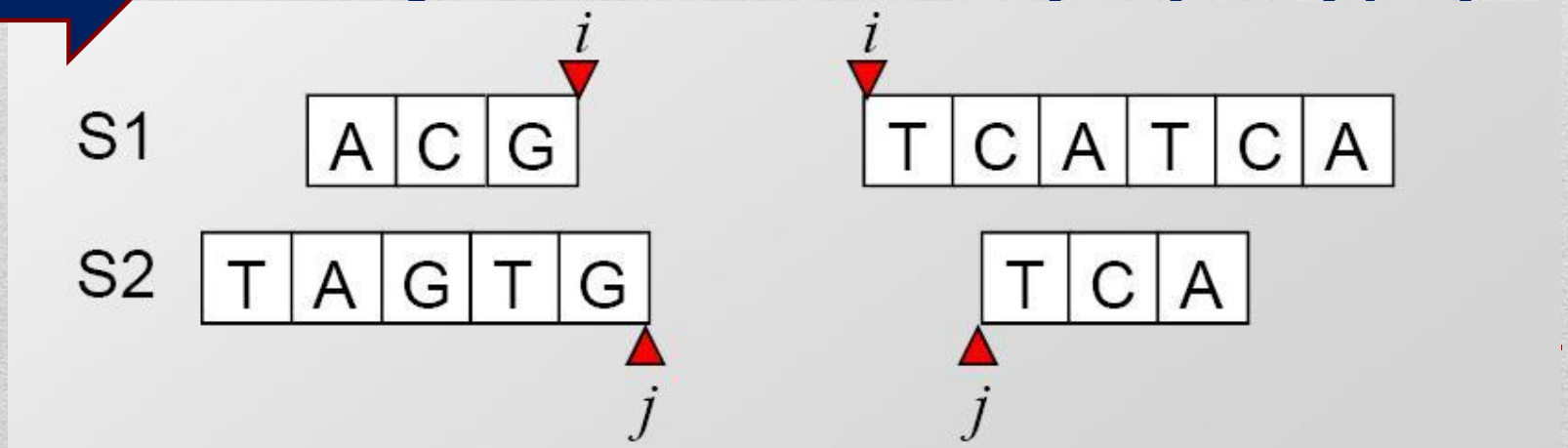
n	Enumerazione	Tecnica presentata oggi
10	184756	100
20	1.40E+11	400
100	9.00E+58	10000

CS

PUNTO CRUCIALE: gli score sono <u>additivi</u>:

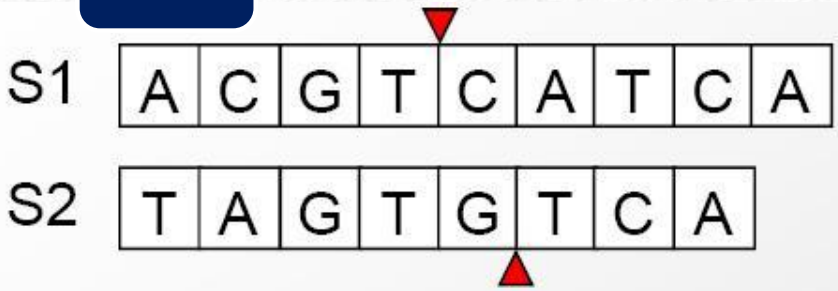


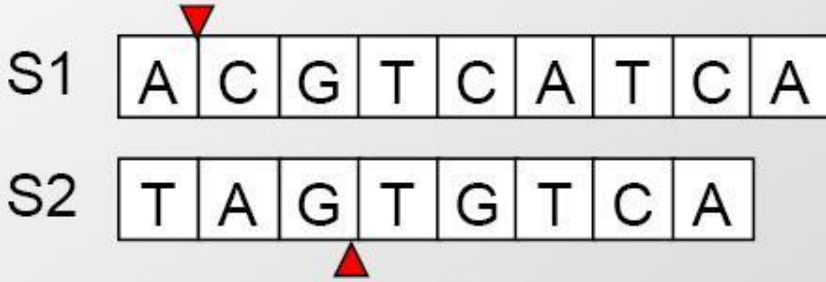
- **Calcolo RICORSIVO del miglior allineamento:**
  - Per una data coppia di sequenze allineate (S1,S2) il miglior allineamento è:
    - . Miglior allineamento di S1[1..i] e S2[1..j]
    - . **+** Miglior allineamento di S1[i..n] e S2[j..m]

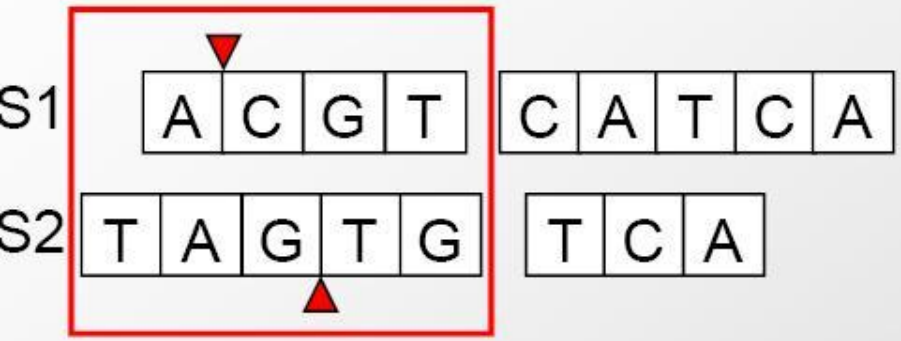


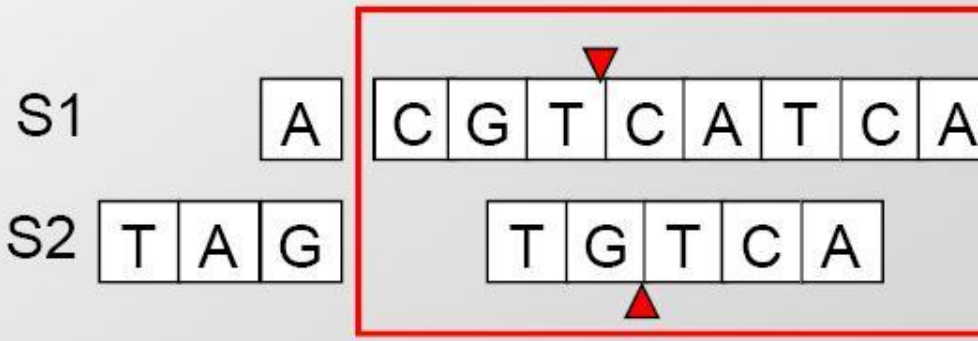
CS

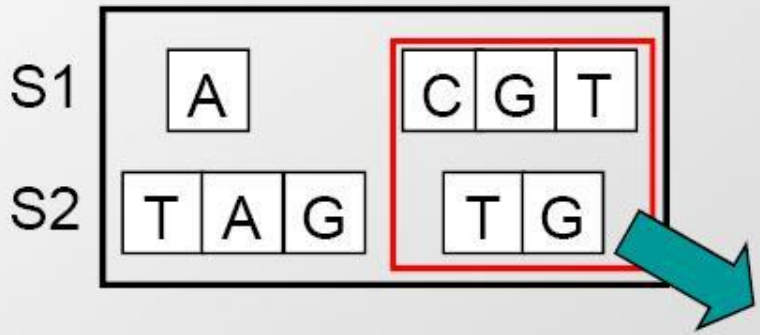
IDEA : riutilizzo delle operazioni svolte dal calcolatore

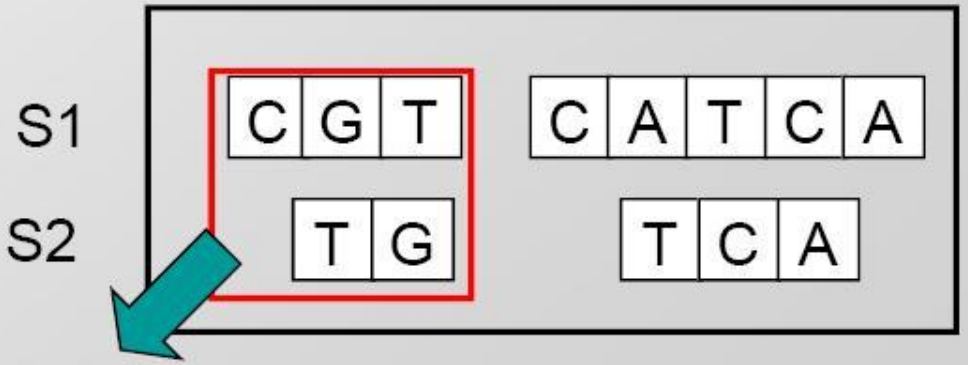












Sottoproblemi IDENTICI ! Possiamo riutilizzare più volte la stessa soluzione.

CS

SOLUZIONE #1 :

- Creazione di un DIZIONARIO (le chiavi sono le sequenze)

- Quando dobbiamo allineare due sequenze cerchiamo una soluzione corrispondente alle due sequenze
- SE ESISTE: ritorniamo il risultato
- SE NON ESISTE:
  - calcoliamo la soluzione
  - inseriamo la soluzione nel dizionario
  - restituiamo la soluzione

- Assicuriamoci di non duplicare i calcoli:
  è necessario calcolare un sottoallineamento una sola volta

LOGICA TOP-DOWN

CS

SOLUZIONE #2 : PROGRAMMAZIONE DINAMICA

- **Creazione di una grossa tabella con indici i,j**
  - Riempiamola compilando gli score di tutti i possibili sottoallineamenti (calcolati in passi che siano i più piccoli possibili)
  - ci serviranno tutti i possibili ”passi” dell’allineamento (per trovare la soluzione finale)

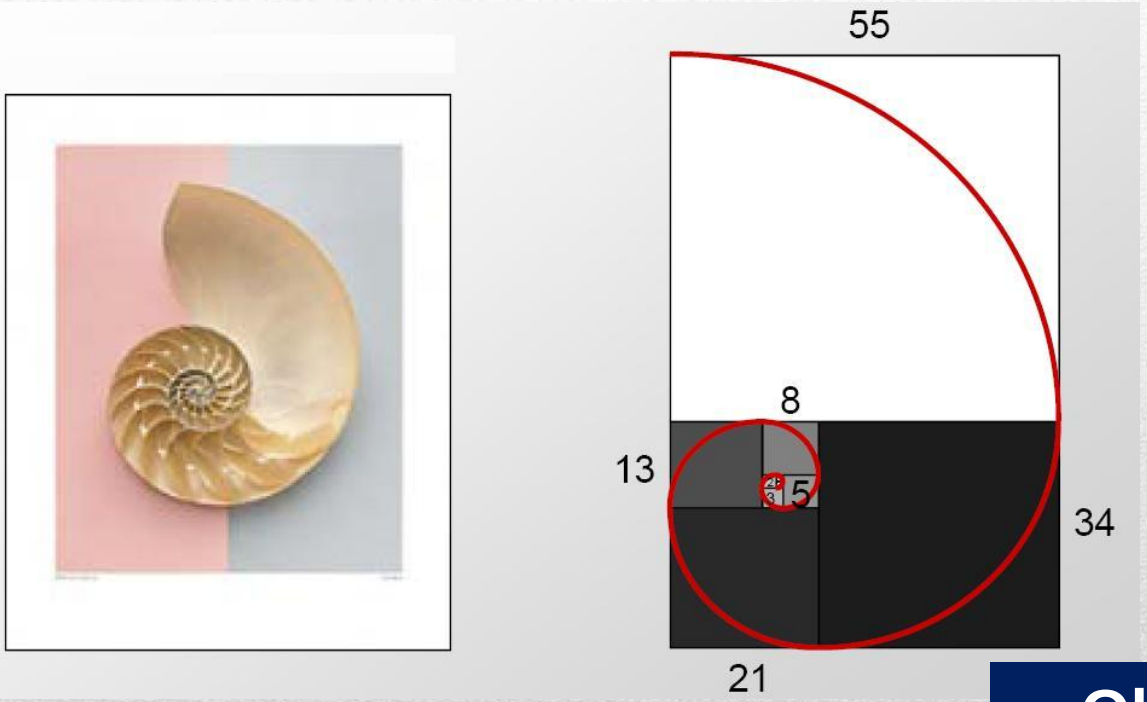
- Assicuriamoci di non duplicare i calcoli:
  è necessario calcolare un sottoallineamento una sola volta

- Soluzione MOLTO SEMPLICE DAL PUNTO DI VISTA COMPUTAZIONALE!

LOGICA BOTTOM-UP

Introduzione alla PROGRAMMAZIONE DINAMICA

I numeri di Fibonacci



… sono definiti in maniera ricorsiva:

F(0)=1, F(1)=1, **F(n) = F(n-1)+F(n-2)**

1,1,2,3,5,8,13,21,34,55,89,144,233,377,…

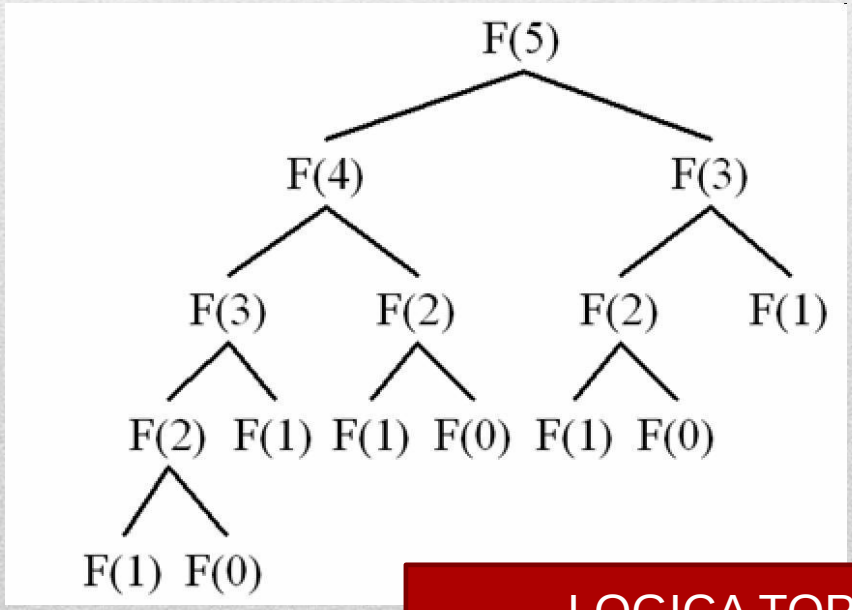
Obiettivo: calcolo dell' **n-esimo** numero di Fibonacci

CS

SOLUZIONE #1 :

- Codice R:

```
fibonacci_TD <- function(n){
        if(n==1 || n==2){
                return(1)
        }else{
                return ( fibonacci_TD(n-1)+fibonacci_TD(n-2) )
        }
}
```

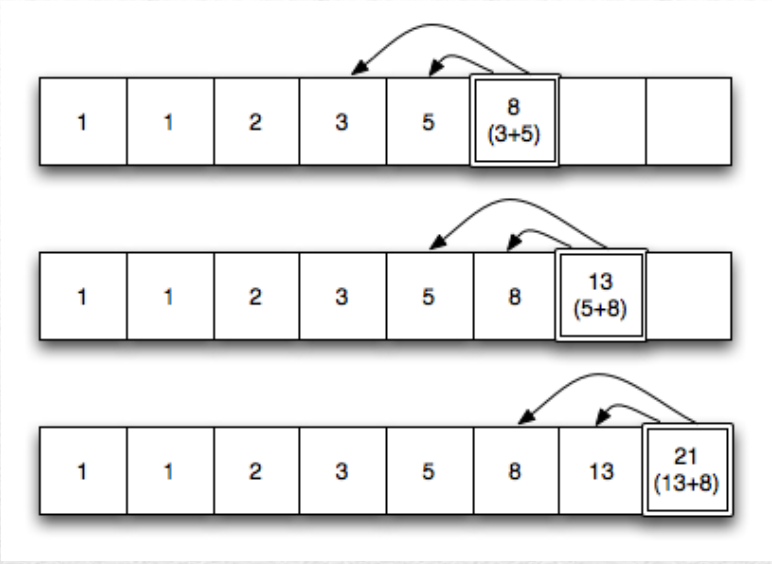


$T(n) = T(n-1)+T(n-2)= (\ldots) = O(2^n)$

complessità **esponenziale**
(in realtà è peggio ancora)

LOGICA TOP-DOWN

SOLUZIONE #2 :



- Codice R:

```
fibonacci_BU <- function(n){
        fib_table<-vector()

        fib_table[1] <- 1
        fib_table[2] <- 1

        for(i in 3:(n+1)){
                fib_table[i] <- fib_table[i-1]+fib_table[i-2]
        }

        return(fib_table[n])

}
```

$T(n) = O(n)$

complessità **lineare**

LOGICA BOTTOM-UP

CS

Cosa ci insegna l'algoritmo iterativo per il calcolo dell'ennesimo numero di Fibonacci? (progr. dinamica)

fib_table	
F[1]	1
F[2]	1
F[3]	2
F[4]	3
F[5]	5
F[6]	8
F[7]	13
F[8]	21
F[9]	34
F[10]	55
F[11]	89
F[12]	?

- **Cosa fa la soluzione iterativa?**

  - rivela sottoproblemi identici
  - ordina le operazioni in modo da favorirne il riutilizzo
  - riempie una tabella di risultati (intermedi)
  - esprime problemi complessi come insiemi di sottoproblemi più semplici

- L'ordine delle operazioni **conta**

  - approccio top-down molto lento (risultati intermedi non disponibili)
  - approccio bottom-up efficace (risolve sottoproblemi, salva risultati, non ricalcola la soluzione ma la costruisce da valori già disponibili)

CS

Basi della programmazione dinamica:

- **Il problema può essere rappresentato come insieme di sottoproblemi**

- **I sottoproblemi non sono sovrapposti (problemi indipendenti che si ripresentano molte volte: calcolo una volta e riutilizzo le soluzioni)**

- **In problemi di ottimizzazione reali:**
  - scelte ottimali a livello **locale**
  - score sommato rispetto a tutto lo spazio dei sottoproblemi
  - soluzione: **traceback** per trovare il “percorso migliore” tra le soluzioni dei sottoproblemi

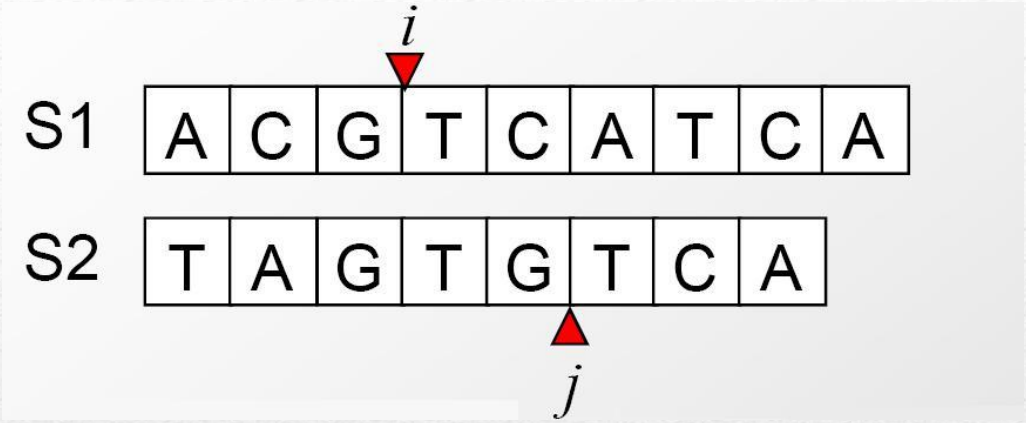
CS

Programmazione dinamica sembra una buona soluzione (finora è stata presentata solo con l'esempio dei numeri di Fibonacci) …

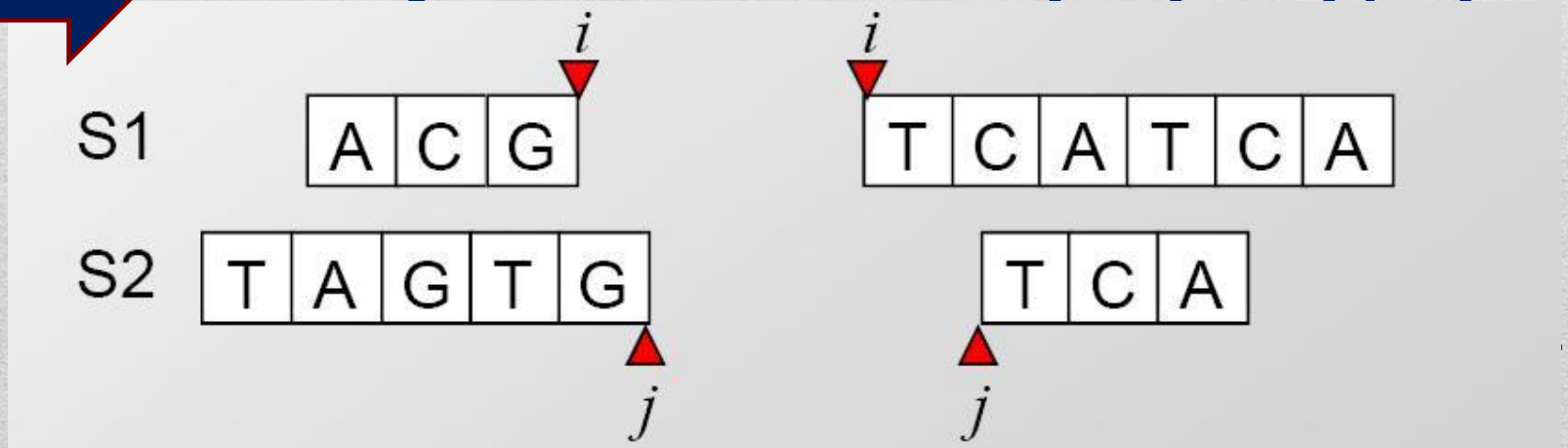
COME POSSIAMO APPLICARLA AL PROBLEMA DELL'ALLINEAMENTO DI SEQUENZE?

CS

PUNTO CRUCIALE: gli score sono <u>additivi</u>:



- **Calcolo RICORSIVO del miglior allineamento:**
  - Per una data coppia di sequenze allineate (S1,S2) il miglior allineamento è:
    - . Miglior allineamento di S1[1..i] e S2[1..j]
    - . **+** Miglior allineamento di S1[i..n] e S2[j..m]



CS

Programmazione dinamica in pratica:

1. **Trovare i 'parametri' della matrice (# dimensioni, variabili)**

2. **Assicurarsi che lo spazio dei sottoproblemi sia finito**
   - se il riutilizzo non è estensivo può darsi che prog. din. non sia la scelta migliore
3. **Ordine di calcolo: i sottorisultati devono essere disponibili quando ci servono**
   - (bottom-up è meglio di top-down … ma non è sempre così ovvio)
4. **Ricorsione: sovraproblema = F(sottoproblema)**
5. **Ricordare le scelte effettuate in precedenza** (tipicamente *F()* include *min()* o *max()*)
6. **Iniziamo a calcolare:**
   - Riempiamo la matrice di sottorisultati
   - Definiamo un "punto di partenza" e ripercorriamo i passi all'indietro fino a trovare la **migliore soluzione**.

CS Bio

ALLINEAMENTO LOCALE DI SEQUENZE NUCLEOTIDICHE

Algoritmo più noto per soluzione del problema:

Smith-Waterman (1981)

Obiettivo:

“Date due sequenze relativamente lunghe, determinare la sottosequenza della prima sequenza che realizza il *maggior grado di similarità* con una sottosequenza della seconda sequenza”.

CS Bio

ALLINEAMENTO LOCALE DI SEQUENZE NUCLEOTIDICHE

Smith-Waterman (1981)

1) Inizializzazione

Data una coppia di sequenze $S_A=a_1,a_2,\ldots,a_i,a_n$ e $S_B=b_1,b_2,\ldots,b_j,b_m$ costruire una matrice $\mathbf{H}=||H_{i,\,j}||$, costituita da n+1 righe e m+1 colonne che viene inizializzata ponendo:

$H_{i,0} = H_{0,j}$ per $0 \leq i \leq n$ e $0 \leq j \leq m$

CS Bio

ALLINEAMENTO <u>LOCALE</u> DI SEQUENZE NUCLEOTIDICHE

Smith-Waterman (1981)

2) Calcolo

Per ciascuna coppia di elementi a_i e b_j appartenenti alle sequenze S_A e S_B si definisce un punteggio di **similarità s** .

Per il confronto di sequenze nucleotidiche tale punteggio viene generalmente definito come segue:

$s(a_i , b_j) = \alpha$ se $a_i = b_j$ (usualmente > 0)
$s(a_i , b_j) = \beta$ se $a_i \neq b_j$ (usualmente ≤ 0)
$W_k = \gamma + \delta(k-1)$ dove γ=costo fisso apertura gap
δ=penalità aggiuntiva per estensione gap

CS Bio

ALLINEAMENTO LOCALE DI SEQUENZE NUCLEOTIDICHE

Smith-Waterman (1981)

2) Calcolo (continua)

I valori della matrice H_{ij} vengono determinati secondo l'equazione:

$$\mathbf{H_{i,j}} = \mathbf{max}(H_{i-1,j-1} + \mathbf{s}(a_i , b_j) , H_{i-1,j} - \mathbf{pgap}, H_{i,j-1} - \mathbf{pgap} , 0)$$

NB: in questo esempio invece di usare $\mathbf{W_k} = \boldsymbol{\gamma}+\boldsymbol{\delta}(k-1)$ come penalità dinamica per i gap di diversa lunghezza abbiamo utilizzato una penalità *costante* (**pgap**) indipendente dall'estensione dei gap.

CS Bio

ALLINEAMENTO LOCALE DI SEQUENZE NUCLEOTIDICHE

Smith-Waterman (1981)

2) Calcolo (continua)

I valori della matrice H_{ij} vengono determinati secondo l'equazione:

$$\mathbf{H_{i,j}} = \mathbf{max}(H_{i-1,j-1} + \mathbf{s}(a_i , b_j) , H_{i-1,j} - \mathbf{pgap}, H_{i,j-1} - \mathbf{pgap} , 0)$$

Confronto tra nucleotidi (match/mismatch) score ≠

delezione in sequenza B di lunghezza 1

delezione in sequenza A di lunghezza 1

CS Bio

ALLINEAMENTO <u>LOCALE</u> DI SEQUENZE NUCLEOTIDICHE

Smith-Waterman (1981)

2) Calcolo (continua)

I valori della matrice H_{ij} vengono determinati secondo l'equazione:

$$\mathbf{H_{i,j}} = \mathbf{max}(H_{i-1,j-1} + \mathbf{s}(a_i , b_j) , H_{i-1,j} - \mathbf{pgap}, H_{i,j-1} - \mathbf{pgap} , 0)$$

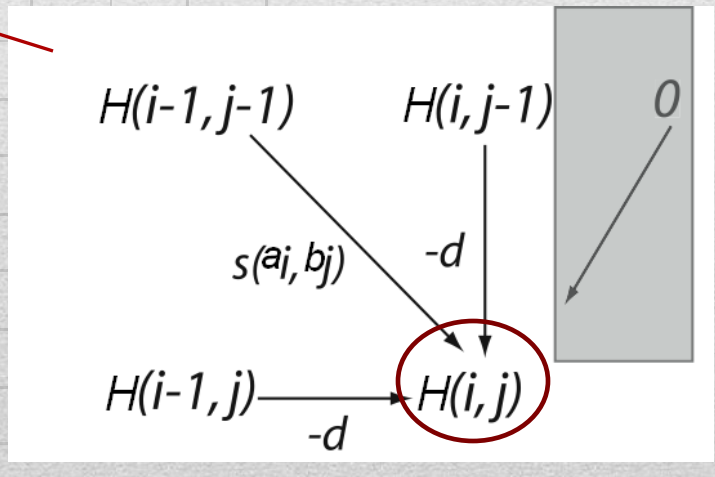
<u>INTERPRETAZIONE</u>: Questa formula **ricorsiva** considera ***tutte le possibili terminazioni*** di segmenti allineati alle cui estremità si trovano $\mathbf{a_i}$ e $\mathbf{b_j}$.

CS Bio

ALLINEAMENTO <u>LOCALE</u> DI SEQUENZE NUCLEOTIDICHE

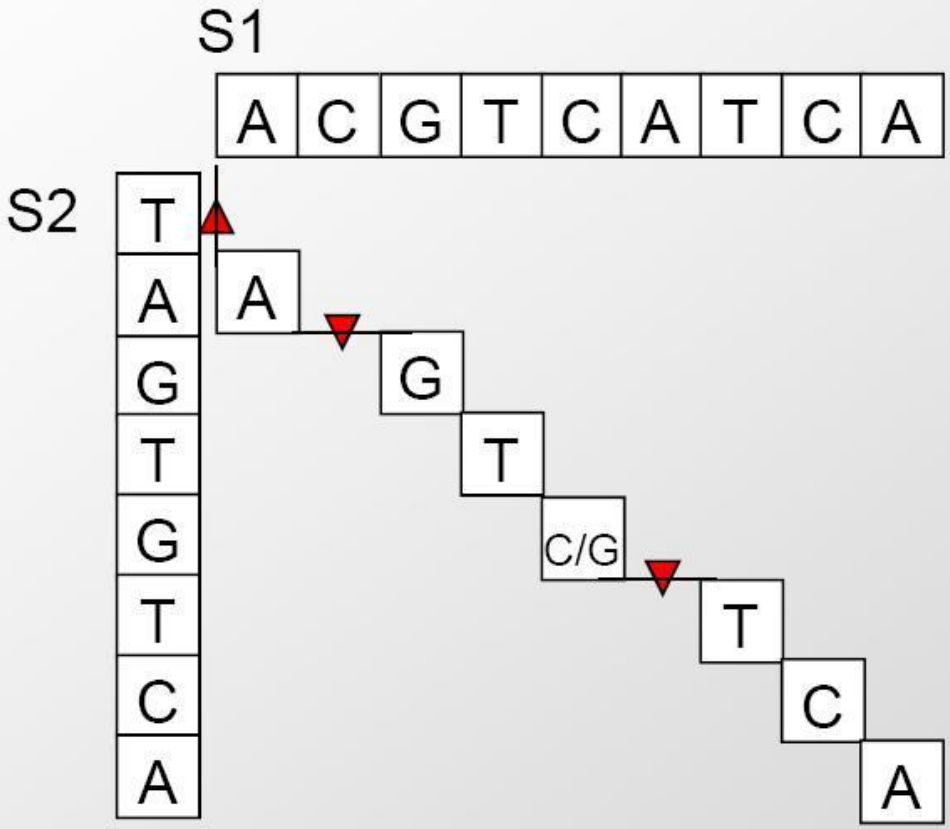
2) Calcolo

Sequence B \ Sequence A		C	A	G	C	C	U	C	G	C	U	U	A	G
		0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0
A		0,0	1,0	0,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,7
A		0,0	1,0	0,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,7
U		0,0	0,0	0,8	0,3	0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,0	0,7
G		0,0	0,0	1,0	0,3	0,0	0,0	0,7	1,0	0,0	0,0	0,7	0,7	1,0
C		1,0	0,0	0,0	2,0	1,3	0,3	1,0	0,3	2,0	0,7	0,3	0,3	0,3
C		1,0	0,7	0,0	1,0	3,0	1,7	?						
A														
U														
U														
G														
A														
C														
G														
G														



CS Bio

DUALITA' :
"miglior allineamento" ↔ miglior percorso attraverso la matrice !



NUOVO OBIETTIVO:

Trovare il <u>miglior percorso</u> attraverso la matrice !

CS Bio

ALLINEAMENTO LOCALE DI SEQUENZE NUCLEOTIDICHE

3) Backtracking

-Trovare valore massimo

- Ripercorrere la matrice fino a quando non raggiungo l' angolo in alto a sinistra o trovo **0** sulla diagonale

		Sequence A												
		C	A	G	C	C	U	C	G	C	U	U	A	G
Sequence B	A	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0
	A	0,0	1,0	0,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,7
	U	0,0	0,0	0,8	0,3	0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,0	0,7
	G	0,0	0,0	**1,0**	0,3	0,0	0,0	0,7	1,0	0,0	0,0	0,7	0,7	1,0
	C	1,0	0,0	0,0	**2,0**	1,3	0,3	1,0	0,3	2,0	0,7	0,3	0,3	0,3
	C	1,0	0,7	0,0	1,0	**3,0**	1,7	1,3	1,0	1,3	1,7	0,3	0,0	0,0
	A	0,0	2,0	0,7	0,3	1,7	2,7	1,3	1,0	0,7	1,0	1,3	1,3	0,0
	U	0,0	0,7	1,7	0,3	1,3	**2,7**	2,3	1,0	0,7	1,7	2,0	1,0	1,0
	U	0,0	0,3	0,3	1,3	1,0	2,3	**2,3**	2,0	0,7	1,7	2,7	1,7	1,0
	G	0,0	0,0	1,3	0,0	1,0	1,0	2,0	**3.3**	2,0	1,7	1,3	2,3	2,7
	A	0,0	1,0	0,0	1,0	0,3	0,7	0,7	2,0	3,0	1,7	1,3	2,3	2,0
	C	1,0	0,0	0,7	1,0	2,0	0,7	1,7	1,7	3,0	2,7	1,3	1,0	2,0
	G	0,0	0,7	1,0	0,3	0,7	1,7	0,3	2,7	1,7	2,7	2,3	1,0	2,0
	G	0,0	0,0	1,7	0,7	0,3	0,3	1,3	1,3	2,3	1,3	2,3	2,0	2,0

CS Bio

ALLINEAMENTO LOCALE DI SEQUENZE NUCLEOTIDICHE

3) Backtracking

-Trovare valore massimo

- Ripercorrere la matrice fino a quando non raggiungo l' angolo in alto a sinistra o trovo **0** sulla diagonale

		Sequence A												
		C	A	G	C	C	U	C	G	C	U	U	A	G
Sequence B	A	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0
	A	0,0	1,0	0,7	0,0	0,0	0,0							
	U	0,0	0,0	0,8	0,3	0,0	0,0							
	G	0,0	0,0	**1,0**	0,3	0,0	0,0							
	C	1,0	0,0	0,0	**2,0**	1,3	0,3							
	C	1,0	0,7	0,0	1,0	**3,0**	1,7	1,3	1,0	1,3	1,7	0,3	0,0	0,0
	A	0,0	2,0	0,7	0,3	1,7	2,7	1,3	1,0	0,7	1,0	1,3	1,3	0,0
	U	0,0	0,7	1,7	0,3	1,3	**2,7**	2,3	1,0	0,7	1,7	2,0	1,0	1,0
	U	0,0	0,3	0,3	1,3	1,0	2,3	**2,3**	2,0	0,7	1,7	2,7	1,7	1,0
	G	0,0	0,0	1,3	0,0	1,0	1,0	2,0	**3.3**	2,0	1,7	1,3	2,3	2,7
	A	0,0	1,0	0,0	1,0	0,3	0,7	0,7	2,0	3,0	1,7	1,3	2,3	2,0
	C	1,0	0,0	0,7	1,0	2,0	0,7	1,7	1,7	3,0	2,7	1,3	1,0	2,0
	G	0,0	0,7	1,0	0,3	0,7	1,7	0,3	2,7	1,7	2,7	2,3	1,0	2,0
	G	0,0	0,0	1,7	0,7	0,3	0,3	1,3	1,3	2,3	1,3	2,3	2,0	2,0

```
G C C A U U G
| | |   | . |
G C C – U C G
```

CS Bio

ALLINEAMENTO GLOBALE DI SEQUENZE

Algoritmo più noto per soluzione del problema:

Needleman-Wunsch (1970)

Obiettivo:

“Date due sequenze A e B, determinare il miglior allineamento (avente quindi il maggior grado si similarità tra le sequenze considerate) che comprenda **ogni elemento della sequenza A ed ogni elemento della sequenza** B. ”

CS Bio

ALLINEAMENTO GLOBALE DI SEQUENZE

Algoritmo più noto per soluzione del problema:

Needleman-Wunsch (1970)

Costrizioni proprie dell'allineamento globale:

1) Lo score similarità calcolato confrontando seq A con Seq B, **S(A,B)**,**DEVE** essere uguale allo score di similarità che si ottiene confrontando seq B con seq A, **S(B,A)**.
2) lo score di similarità tra due sequenze A e B, **S(A,B) DEVE** essere minore o uguale alla somma degli score di similarità ottenuti confrontando sia seq A che seq B con una terza sequenza (seq C): **$S(A,B) \leq S(A,C) + S(B,C)$**

CS Bio

ALLINEAMENTO GLOBALE DI SEQUENZE

Algoritmo più noto per soluzione del problema:

Needleman-Wunsch (1970)

Anche l'algoritmo di Needleman-Wunsh, come quello di Smith-Waterman, è basato su tecniche di programmazione dinamica.

CS Bio

ALLINEAMENTO <u>GLOBALE</u> DI SEQUENZE

Needleman-Wunsch (1970)

1) Date due sequenze A e B di lunghezza n ed m, rispettivamente, viene costruita una matrice n per m. Nel caso più semplice i valori delle celle M_{ij} vengono inizializzati come segue:

1 (caratteri identici)
0 (altrimenti)

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						1					1		
C			1					1		1			
K													
C			1					1		1			
R						1					1		
B		1											
P												1	

CS Bio

ALLINEAMENTO GLOBALE DI SEQUENZE

Needleman-Wunsch (1970)

2) Una volta inizializzata, la matrice viene **trasformata** a partire dalla cella **in basso a destra**.

Trasformazione:
Aggiungere allo score della cella corrente il massimo valore della celle appartenenti alla coppia riga-colonna avente l'angolo in alto a sinistra posizionato in basso e a destra della cella considerata.

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						1					1		
C			1					1		1			
K													
C			1					1		1			
R						1					2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

CS Bio

ALLINEAMENTO <u>GLOBALE</u> DI SEQUENZE

Needleman-Wunsch (1970)

2) **Trasformazione …**

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

CS Bio

ALLINEAMENTO <u>GLOBALE</u> DI SEQUENZE

Needleman-Wunsch (1970)

2) **Trasformazione …**

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	*8*	7	6	6	5	4	4	3	3	2	1	0	0
Y	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
Y	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

CS Bio

ALLINEAMENTO <u>GLOBALE</u> DI SEQUENZE

Needleman-Wunsch (1970)

3) **Traceback**

Partendo dal valore massimo (8) si procede verso verso la cella in basso a destra seguendo il percorso che realizza il **punteggio più alto**.

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	**8**	7	6	6	5	4	4	3	3	2	1	0	0
Y	7	**7**	6	6	6	4	4	3	3	2	1	0	0
C	6	6	**7**	**6**	5	4	4	4	3	3	1	0	0
Y	6	6	6	5	**6**	4	4	3	3	2	1	0	0
N	5	5	5	6	**5**	4	4	3	3	2	1	0	0
R	4	4	4	4	4	**5**	**4**	3	3	2	2	0	0
C	3	3	4	3	3	3	3	**4**	3	3	1	0	0
K	3	3	3	3	3	3	3	3	**3**	2	1	0	0
C	2	2	3	2	2	2	2	3	2	**3**	1	0	0
R	2	1	1	1	1	2	1	1	1	1	**2**	0	0
B	1	2	1	1	1	1	1	1	1	1	**1**	0	0
P	0	0	0	0	0	0	0	0	0	0	0	**1**	0

CS Bio

ALLINEAMENTO GLOBALE DI SEQUENZE

Needleman-Wunsch (1970)

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	**8**	7	6	6	5	4	4	3	3	2	1	0	0
Y	7	**7**	6	6	6	4	4	3	3	2	1	0	0
C	6	6	**7**	**6**	5	4	4	4	3	3	1	0	0
Y	6	6	6	5	**6**	4	4	3	3	2	1	0	0
N	5	5	5	6	**5**	4	4	3	3	2	1	0	0
R	4	4	4	4	4	**5**	**4**	3	3	2	2	0	0
C	3	3	4	3	3	3	3	**4**	3	3	1	0	0
K	3	3	3	3	3	3	3	3	**3**	2	1	0	0
C	2	2	3	2	2	2	2	3	2	**3**	1	0	0
R	2	1	1	1	1	2	1	1	1	1	**2**	0	0
B	1	2	1	1	1	1	1	1	1	1	**1**	0	0
P	0	0	0	0	0	0	0	0	0	0	0	**1**	0

```
ABCNY-RQCLCR-PM
| | | | | | | |
AYC-YNR-CKCRBP-
```

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	**8**	7	6	6	5	4	4	3	3	2	1	0	0
Y	7	**7**	6	6	6	4	4	3	3	2	1	0	0
C	6	6	**7**	6	5	4	4	4	3	3	1	0	0
Y	5	6	**6**	5	***6***	4	4	3	3	2	1	0	0
N	5	5	5	**6**	**5**	4	4	3	3	2	1	0	0
R	4	4	4	4	4	**5**	**4**	3	3	2	2	0	0
C	3	3	4	3	3	3	3	**4**	3	3	1	0	0
K	3	3	3	3	3	3	3	3	**3**	2	1	0	0
C	2	2	3	2	2	2	2	3	2	**3**	1	0	0
R	2	1	1	1	1	2	1	1	1	1	**2**	0	0
B	1	2	1	1	1	1	1	1	1	1	**1**	0	0
P	0	0	0	0	0	0	0	0	0	0	0	**1**	0

```
ABC-NYRQCLCR-PM
| | | | | | | |
AYCYN-R-CKCRBP-
```

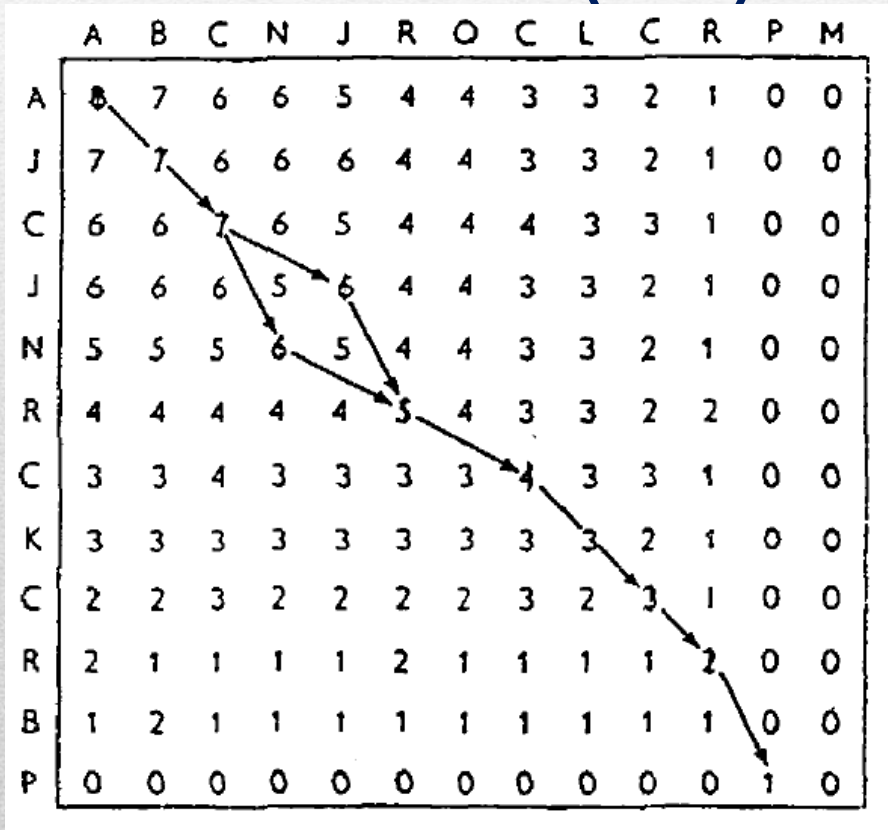
Entrambi i percorsi (**allineamenti**) realizzano il **massimo punteggio** (8)

CS Bio

ALLINEAMENTO <u>GLOBALE</u> DI SEQUENZE

Needleman-Wunsch (1970)

NB:
Non sono ammessi spostamenti **<u>diretti</u>** verso destra o verso il basso.



	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

La soluzione non è **unica …** **(+ <u>percorsi</u> con lo score massimo)**

Allineamento globale : GOTOH 1982

L’algoritmo di Needleman-Wunsch è stata la prima soluzione ampiamente utilizzata per risolvere problemi di allineamento globale. Ma, come soluzione, non era efficiente.

Nel 1982 Gotho propose una variante simile a quella proposta in precedenza da Smith e Waterman. Lo schema è molto simile ma ci sono alcune differenze importanti.

CS Bio

ALLINEAMENTO DI SEQUENZE

Allineamento globale : GOTOH 1982

- Costruzione matrice m per n
- Inizializzazione:

H(0,0) = 0 H(i,0) = i * pgap H(0,j) = j * pgap

- Riempimento matrice:

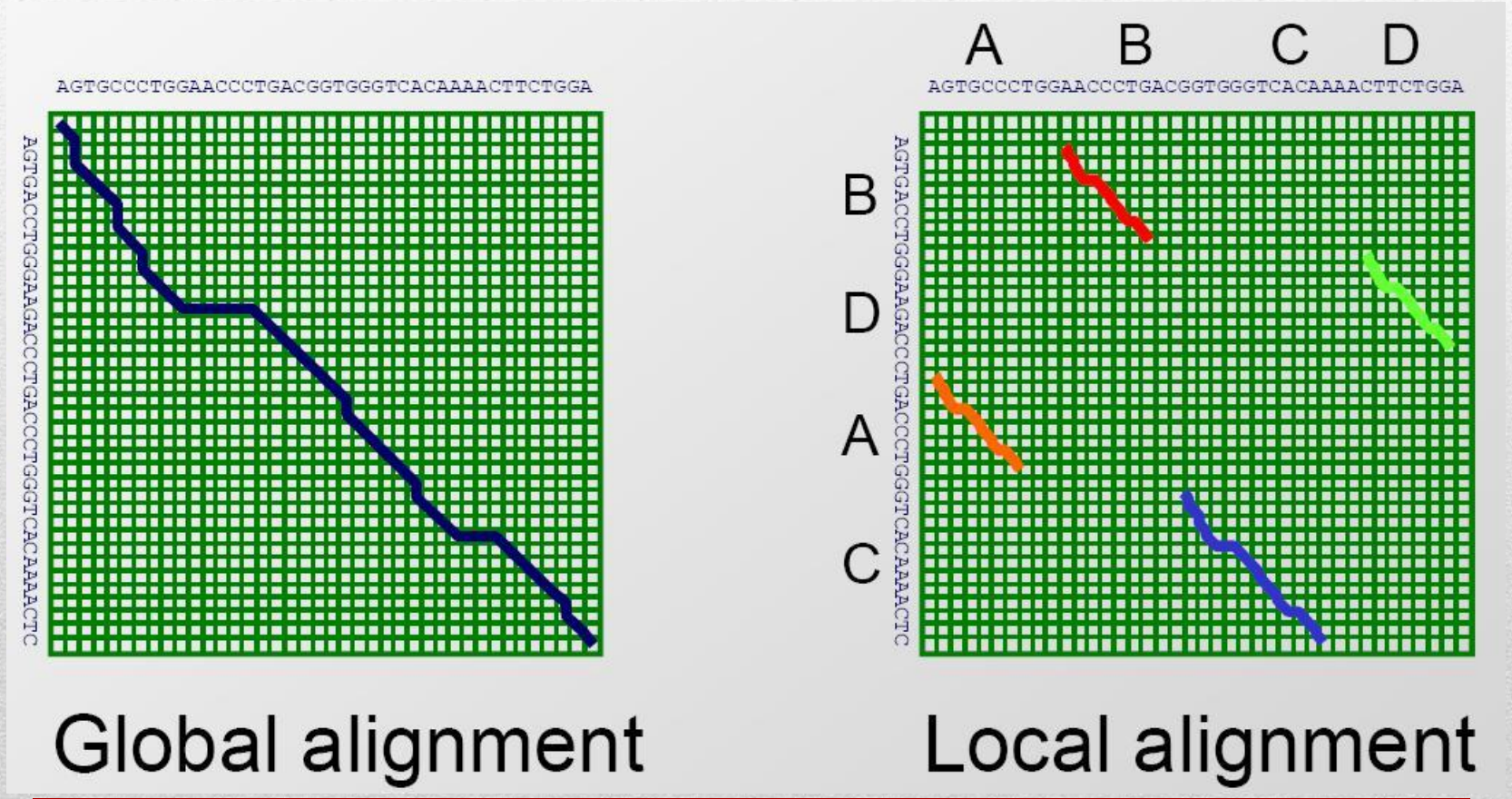
Come SW ma senza considerare lo **0** (possibili valori < 0)

- Inizio traceback : **sempre da H(m,n)**

CS Bio

ALLINEAMENTO DI SEQUENZE

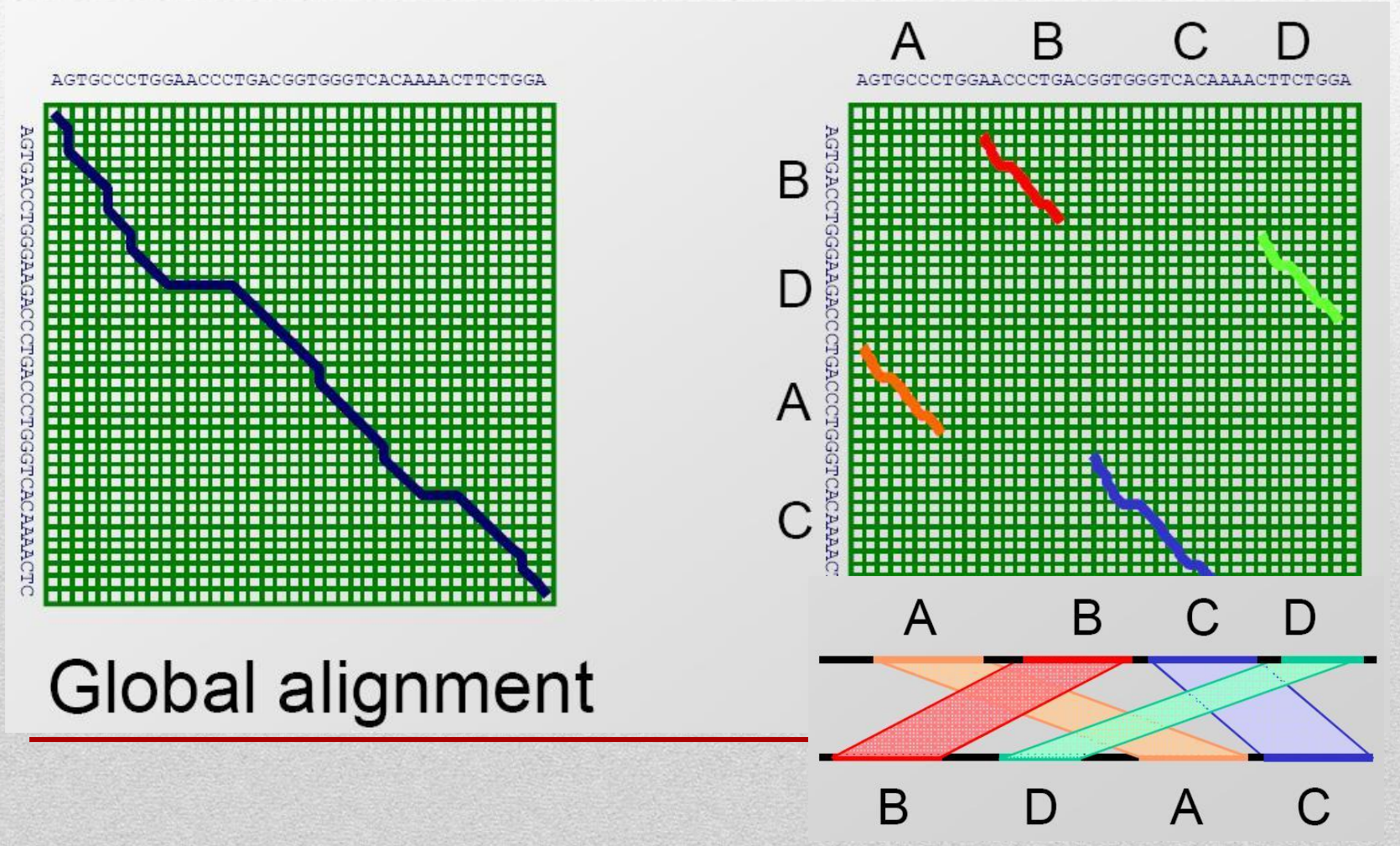
Allineamento globale vs locale



CS Bio

ALLINEAMENTO DI SEQUENZE

Allineamento globale vs locale



CS Bio

ALLINEAMENTO DI SEQUENZE

Allineamento globale vs locale

Needleman-Wunsch algorithm

Initialization: $H(0, 0) = 0$

Iteration:

$$H(i, j) = \max \begin{cases} H(i-1, j) - d \\ H(i, j-1) - d \\ H(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: Bottom right

Smith-Waterman algorithm

Initialization: $H(0, j) = H(i, 0) = 0$

Iteration:

$$H(i, j) = \max \begin{cases} 0 \\ H(i-1, j) - d \\ H(i, j-1) - d \\ H(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: *max*

CS Bio

ALLINEAMENTO DI SEQUENZE

Riassunto:

Il processo di allineamento cerca di quantificare la **similarità** (numero di variazioni esistenti) tra le sequenze confrontate.

I metodi presentati si basano su tecniche di **programmazione dinamica** (soluzione di problemi grandi una volta che questi ultimi sono stati scomposti in problemi più piccoli)

Diversi tipi di allineamento: locale e globale

Bio

ALLINEAMENTO DI SEQUENZE:

CS

I genomi cambiano nel tempo. **Pattern evolutivi simili** possono aiutarci a identificare sequenze **funzionalmente simili**.

Confronto di biosequenze (nt, aa) : quantificare la **similarità** sulla base delle variazioni necessarie per trasformare una sequenza nell'altra (**allineamento**).

Molti modi di allineare due sequenze. E' necessaria una soluzione per trovare il **miglior allineamento** possibile.

Soluzione efficiente: suddividere il problema in molti sottoproblemi ripetitivi e calcolare le soluzioni dei sottoproblemi seguendo un ordine che garantisca il possibile riutilizzo delle soluzioni già calcolate → **programmazione dinamica**.

Bio

ALLINEAMENTO DI SEQUENZE:

CS

Alla fine del processo di allineamento *COSA OTTENIAMO*?

CS: Allineamento con score migliore (più allineamenti possono avere il massimo score), calcolato in modo efficiente.

Bio:

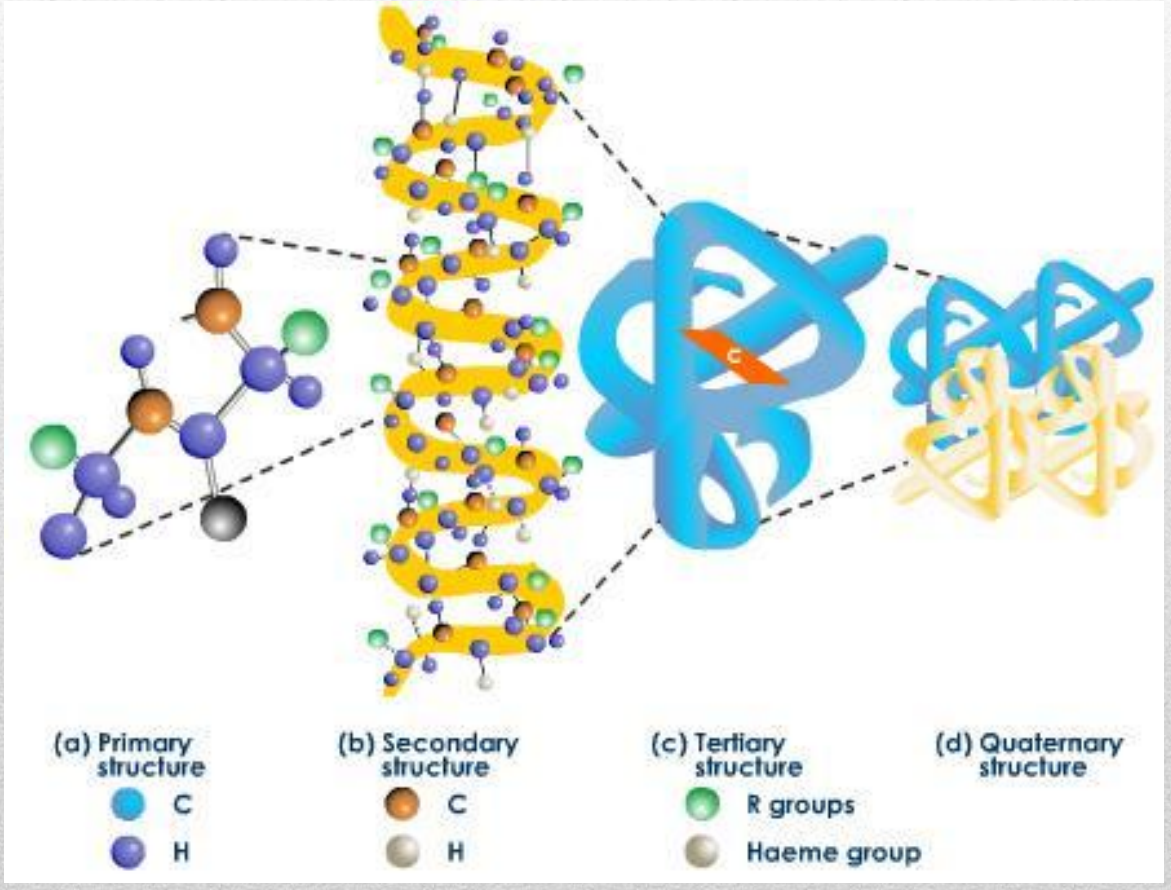
UN ALLINEAMENTO E' UNA ***IPOTESI EVOLUTIVA***

(avente come obiettivo quello di suggerire l'esistenza di relazioni funzionali tra le sequenze confrontate)

Bio

ALLINEAMENTO ↔ IPOTESI EVOLUTIVA

Se è vero che un allineamento **equivale** ad un'ipotesi evolutiva, cosa giustifica tale ipotesi? (dal punto di vista **biologico**)



(conservazione **crescente** durante l'evoluzione in caso di omologia funzionale)

Bio

ALLINEAMENTO ↔ IPOTESI EVOLUTIVA

La sequenza delle proteine è meno conservata rispetto a struttura secondaria, terziaria e quaternaria nel corso dell'evoluzione.

Questo ha due effetti:

1) Proteine **OMOLOGHE** possono avere sequenza molto diverse e quindi produrre allineamenti aventi un punteggio di similarità basso.

2) Se la **similarità** tra due sequenze proteiche è elevata (in maniera statisticamente significativa) è abbastanza ragionevole ipotizzare che tra di esse esista una relazione di **omologia funzionale**.

OMOLOGIA: carattere qualitativo, **SIMILARITA'**: carattere quantitativo

Bio

ALLINEAMENTO ↔ IPOTESI EVOLUTIVA

Per ora abbiamo discusso il problema dell'allineamento di biosequenze e abbiamo visto le soluzioni algoritmiche. Ma dobbiamo sempre ricordare che:

I metodi presentati (Needleman-Wunsch e Smith-Waterman) permettono di trovare il "miglior" allineamento in modo efficiente ma non garantiscono che il risultato sia **biologicamente sensato**!

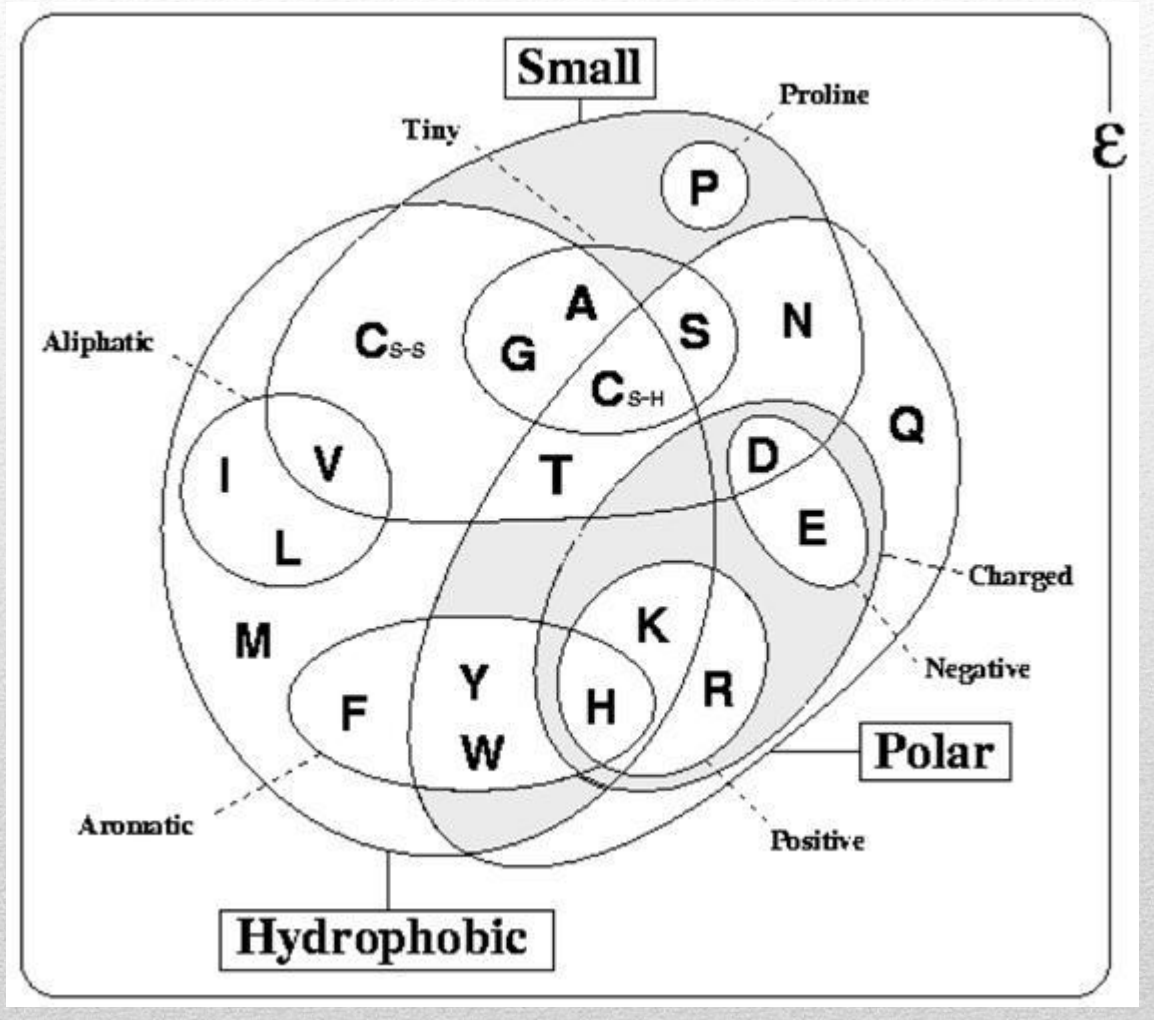
Perché un allineamento risulti informativo è di **FONDAMENTALE** importanza che gli schemi di **SCORING** abbiano un senso biologico (questo è particolarmente evidente nel caso di allineamenti di sequenze **PROTEICHE**).

Bio

ALLINEAMENTO ↔ IPOTESI EVOLUTIVA

Ha senso parlare di SCORE FISSI per match e mismatch quando cerchiamo di allineare due sequenze proteiche?

NO



Bio

ALLINEAMENTO ↔ IPOTESI EVOLUTIVA

Ha senso parlare di SCORE FISSI per match e mismatch quando cerchiamo di allineare due sequenze proteiche?

NO

COME E' POSSIBILE DECIDERE QUALE SCORE ASSEGNARE A DEI PARTICOLARI match/mismatch TRA AMINOACIDI?

E' un problema biologico (si)
E' un problema informatico (no)

… è un problema statistico!

Bio

ALLINEAMENTO DI SEQUENZE:
INTERPRETAZIONE PROBABILISTICA

Stat

In questa parte discuteremo di:

- Punteggi per i match
- Penalità per i mismatch

(sia nel caso di allineamenti nucleotidici che aminoacidici)

- **Origine** degli score utilizzati in applicazioni pratiche

Matrici di sostituzione:

-PAM

-BLOSUM

Bio

ALLINEAMENTO DI SEQUENZE: INTERPRETAZIONE PROBABILISTICA

Stat

Nucleotide space:

	A	G	T	C
A	+1	-½	-1	-1
G	-½	+1	-1	-1
T	-1	-1	+1	-½
C	-1	-1	-½	+1

purine pyrimid.

Transitions:

A⇔G, C⇔T common (lower penalty)

Transversions:

All other operations

Protein space: amino-acid match scores

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W

BLOSUM matrix of AA similarity scores

Bio

ALLINEAMENTO DI SEQUENZE:
INTERPRETAZIONE PROBABILISTICA

Stat

Modello probabilistico di allineamento di biosequenze:

- Ci concentriamo su allineamenti proteici senza gap
- Dato un allineamento possiamo considerare due opzioni:

C: Deriva da sequenze evolutivamente **C**orrelate
N: Deriva da sequenze evolutivamente **N**on correlate

- Come è possibile distinguere tra i due casi?
- C'è un collegamento tra questa domanda e le matrici di sostituzione amminoacidiche?

Bio

ALLINEAMENTO DI SEQUENZE:
INTERPRETAZIONE PROBABILISTICA

Stat

Sequenze evolutivamente NON correlate:

• Assumiamo che ogni aa in ogni posizione dell'allineamento derivi da un campionamento CASUALE di una distribuzione di aa.

q_a = probabilità dell' aminoacido a

• La probabilità di un allineamento di n caratteri tra le sequenze x ed y è data da:

$$Pr(x, y|N) = \prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}$$

Bio

ALLINEAMENTO DI SEQUENZE: INTERPRETAZIONE PROBABILISTICA

Stat

Sequenze evolutivamente Correlate:

• Assumiamo che ogni **coppia** di aa allineati derivi da un ancestor comune.

P_{ab} = probabilità che **l'evoluzione** abbia originato un aminoacido a nella sequenza x ed un aminoacido b nella sequenza y.

• La probabilità di un allineamento di n caratteri tra le sequenze x ed y è data da:

$$Pr(x, y|C) = \prod_{i=1}^{n} p_{x_i y_i}$$

Bio

ALLINEAMENTO DI SEQUENZE: INTERPRETAZIONE PROBABILISTICA

Stat

Come decidiamo quale dei modelli (C o N) è più probabile dato un allineamento?

• Consideriamo la probabilità relativa dei modelli dato l'allineamento:

$$\frac{Pr(x,y|C)}{Pr(x,y|N)} = \frac{\prod_{i=1}^{n} p_{x_i y_i}}{\prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}} = \frac{\prod_{i=1}^{n} p_{x_i y_i}}{\prod_{i=1}^{n} q_{x_i} q_{y_i}}$$

• Calcolando il logaritmo otteniamo:

$$log \frac{Pr(x,y|C)}{Pr(x,y|N)} = \sum_{i} \left(\frac{p_{x_i y_i}}{q_{x_i} q_{y_i}} \right)$$

Bio

ALLINEAMENTO DI SEQUENZE:
INTERPRETAZIONE PROBABILISTICA

Stat

• Gli elementi della matrice degli score di sostituzione per tutte le coppie di aminoacidi a b si calcolano come segue:

$$s(a,b) = log\left(\frac{P_{ab}}{q_a q_b}\right)$$

Ma come calcoliamo i valori P_{ab} (prob. che ***a*** e ***b*** si siano originati dal medesimo ancestor nel corso dell'evoluzione)?

DIPENDE DAL <u>TEMPO DI DIVERGENZA</u>:

- Divergenza **recente** $P_{ab} \approx 0$ se $a \neq b$
- Divergenza **lontana** $P_{ab} \approx q_a q_b$

Bio

MATRICI DI SOSTITUZIONE

Stat

Idea chiave:

Collezioni di sequenze che **sappiamo a priori** essere evolutivamente correlate possono fornire informazioni riguardanti le sostituzioni “biologicamente permesse.

Questo pone un problema:

Come possiamo costruire collezioni di sequenze “evolutivamente correlate” … che **criterio** possiamo utilizzare per selezionare sequenze evolutivamente correlate?

Bio

MATRICI DI SOSTITUZIONE

Stat

MATRICI PAM (Percent Accepted Mutation Matrices)

[Dayhoff ,1978] **Assunzione:** analizzando sequenze correlate filogeneticamente (cioè evolutivamente) si può calcolare la probabilità con cui ogni amminoacido subisce un evento di sostituzione, ovvero una point accepted mutation.

Furono calcolate le probabilità di sostituzione di ogni amminoacido con ciascun altro, prevedendo in media **una mutazione ogni 100** aminoacidi: questa venne chiamata **distanza evolutiva 1 PAM**.

La matrice **PAM 1** non ha applicazioni pratiche ma serve per calcolare le matrici PAM di ordine superiore, ottenute simulando successivi passi evolutivi, cioè applicando le probabilità di sostituzione definite in PAM 1.
Tipicamente si usano matrici comprese tra **PAM 10** (per sequenze filogeneticamente vicine) e **PAM 240** (per sequenze filogeneticamente molto lontane).

Bio

MATRICI DI SOSTITUZIONE

Stat

BLOSUM (Blocks Amino Acid Substitution Matrices):

[**Henikoff & Henikoff , 1992]** Utilizzano la banca dati **BLOCKS** che contiene una collezione di allineamenti multipli (senza gap) di segmenti di proteine suddivisi in famiglie. I blocchi corrispondono a regioni **strutturalmente conservate !**

Ogni blocco di allineamenti contiene sequenze con un numero di aminoacidi identici superiore ad una certa percentuale P, solitamente compresa tra il 30% e il 95%.

Da ogni blocco è possibile ricavare la frequenza relativa di sostituzione degli aminoacidi, che può essere utilizzata per calcolare la matrice di punteggi.

L'indice associato alla matrice BLOSUM indica la **percentuale di identità minima** all'interno del blocco.

Bio

MATRICI DI SOSTITUZIONE

Stat

PAM vs BLOSUM

BLOSUM 45 **BLOSUM 62** **BLOSUM 90**

> Divergenza < Divergenza

PAM 250 **PAM 160** **PAM 100**

Bio

MATRICI DI SOSTITUZIONE

Stat

PAM vs BLOSUM: quale delle due scegliere? (**I)**

I due tipi di matrici partono da **presupposti diversi**:

PAM: si assume un modello in cui le sostituzioni aminoacidiche osservate a grandi distanze evolutive derivino esclusivamente dalla somma di tante mutazioni indipendenti. I punteggi risultanti esprimono quanto sia più probabile che l'appaiamento di una coppia di aminoacidi sia dovuto a omologia piuttosto che al caso.

BLOSUM: non sono basate esplicitamente su un modello evolutivo delle mutazioni dei singoli aa. E' in grado di catturare meglio la sostenibilità di una certa mutazione **dato un contesto** (fornito dai blocchi di sequenze corrispondenti a regioni strutturalmente conservate).

Bio

MATRICI DI SOSTITUZIONE

Stat

PAM vs BLOSUM: quale delle due scegliere? **(II)**

Il confronto tra le matrici PAM e BLOSUM, a un livello di sostituzioni paragonabili, indica che i due tipi di matrice sono **effettivamente molto simili tra loro, ma non uguali**.

Le **PAM** tendono a premiare sostituzioni aminoacidiche che derivano da una singola base, penalizzando le sostituzioni che comportano cambi di codone più complessi.

Si ritiene che le matrici **BLOSUM** siano più adatte delle PAM per effettuare **ricerche di similarità** di sequenze.

Bio

BLOSUM 50

Stat

Sostituzioni a penalità lieve?

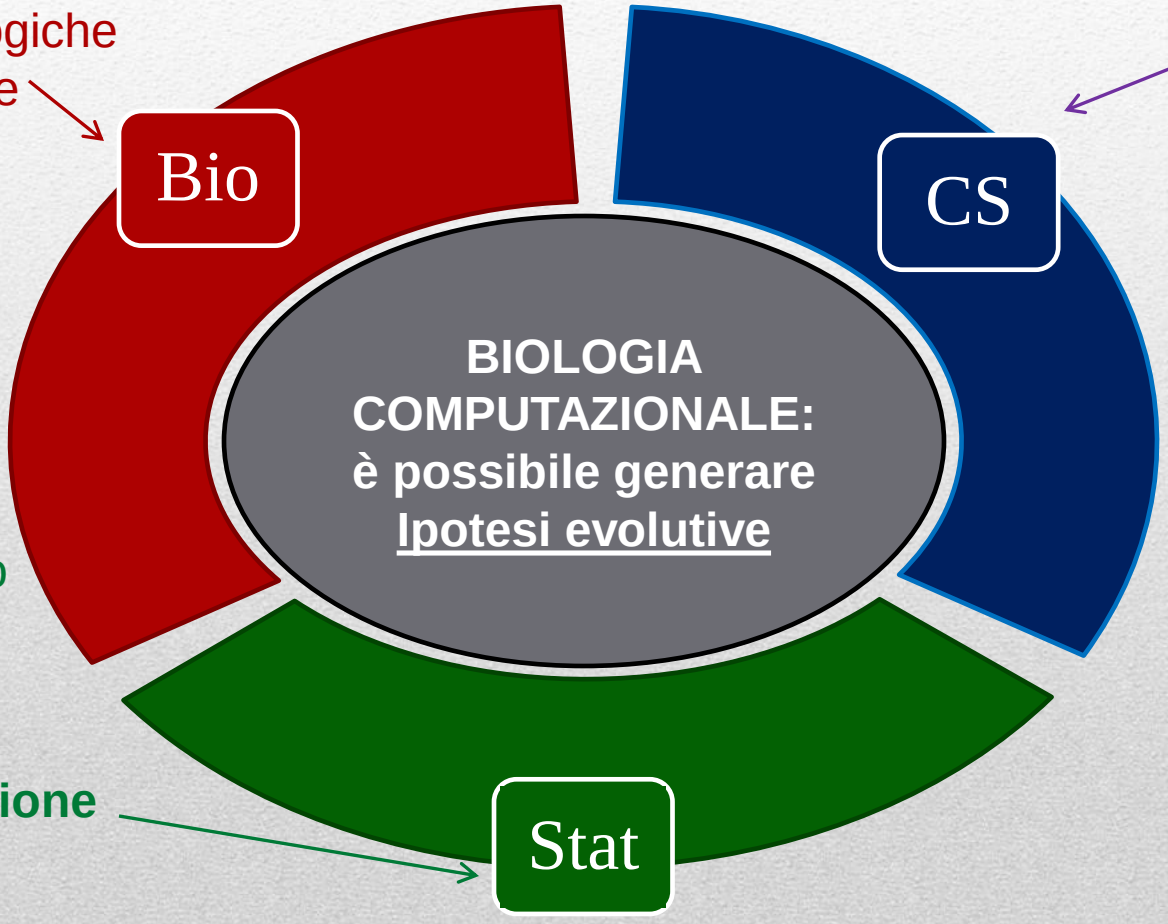
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**5**	−2	−1	−2	−1	−1	−1	0	−2	−1	−2	−1	−1	−3	−1	1	0	−3	−2	0
R	−2	**7**	−1	−2	−4	1	0	−3	0	−4	−3	3	−2	−3	−3	−1	−1	−3	−1	−3
N	−1	−1	**7**	2	−2	0	0	0	1	−3	−4	0	−2	−4	−2	1	0	−4	−2	−3
D	−2	−2	2	**8**	−4	0	2	−1	−1	−4	−4	−1	−4	−5	−1	0	−1	−5	−3	−4
C	−1	−4	−2	−4	**13**	−3	−3	−3	−3	−2	−2	−3	−2	−2	−4	−1	−1	−5	−3	−1
Q	−1	1	0	0	−3	**7**	2	−2	1	−3	−2	2	0	−4	−1	0	−1	−1	−1	−3
E	−1	0	0	2	−3	2	**6**	−3	0	−4	−3	1	−2	−3	−1	−1	−1	−3	−2	−3
G	0	−3	0	−1	−3	−2	−3	**8**	−2	−4	−4	−2	−3	−4	−2	0	−2	−3	−3	−4
H	−2	0	1	−1	−3	1	0	−2	**10**	−4	−3	0	−1	−1	−2	−1	−2	−3	2	−4
I	−1	−4	−3	−4	−2	−3	−4	−4	−4	**5**	2	−3	2	0	−3	−3	−1	−3	−1	4
L	−2	−3	−4	−4	−2	−2	−3	−4	−3	2	**5**	−3	3	1	−4	−3	−1	−2	−1	1
K	−1	3	0	−1	−3	2	1	−2	0	−3	−3	**6**	−2	−4	−1	0	−1	−3	−2	−3
M	−1	−2	−2	−4	−2	0	−2	−3	−1	2	3	−2	**7**	0	−3	−2	−1	−1	0	1
F	−3	−3	−4	−5	−2	−4	−3	−4	−1	0	1	−4	0	**8**	−4	−3	−2	1	4	−1
P	−1	−3	−2	−1	−4	−1	−1	−2	−2	−3	−4	−1	−3	−4	**10**	−1	−1	−4	−3	−3
S	1	−1	1	0	−1	0	−1	0	−1	−3	−3	0	−2	−3	−1	**5**	2	−4	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	2	**5**	−3	−2	0
W	−3	−3	−4	−5	−5	−1	−3	−3	−3	−3	−2	−3	−1	1	−4	−4	−3	**15**	2	−3
Y	−2	−1	−2	−3	−3	−1	−2	−3	2	−1	−1	−2	0	4	−3	−2	−2	2	**8**	−1
V	0	−3	−3	−4	−1	−3	−3	−4	−4	4	1	−3	1	−1	−3	−2	0	−3	−1	**5**

Isoleucina e Leucina: score di mutazione reciproci > 0

Triptofano: premio di “conservazione” estremamente elevato

ALLINEAMEMTO PAIRWISE

Sequenze biologiche **variano** durante l'evoluzione



E' **possibile** calcolare in modo **efficiente** un allineamento che richieda il minor numero variazioni per passare da una sequenza all'altra

Perché le ipotesi evolutive generate abbiano senso è necessario assegnare degli score di **sostituzione** che siano statisticamente frequenti nel corso dell'evoluzione