BIOINFORMATICA

Introduzione e stringhe L1

La bioinformatica è una disciplina di frontiera, non canonizzata, molto recente nata grazie alle nuove tecnologie (intorno agli anni '80, grazie alla definizione delle sequenze di acidi nucleici e proteine).
La sua nascita è stata promossa da 3 eventi:
- le nuove tecnologie, il poter utilizzare testi nei computer oltre che numeri;
- le capacità di sequenziare acidi nucleici e proteine;
- la comparsa dei circuiti integrati programmabili tramite codice binario (come se fosse un miniprocessore), capaci di compiere molte funzioni (rivoluzione microelettronica).

E' la disciplina che applica alla biologia i principi della Scienza dell'informazione per rendere maggiormente comprensibile il complesso mondo delle Scienze della vita. Quindi in sintesi quando parliamo di bioinformatica ci riferiamo da un lato a tutto ciò che ci permette di conservare, organizzare e distribuire dati relativi agli esseri viventi che derivano per esempio dalla genomica e da discipline affini e dall'altro allo sviluppo di metodologie e all'implementazione di algoritmi per l'indagine in campo biomedico: tali metodologie sono utilizzate per l'estrapolazione dell'informazione biologica dai dati grezzi. La bioinformatica è quindi il rappresentare problemi biologici, sotto forma di sequenze, tramite strumenti informatici (il problema è biologico, l'informatica è solo lo strumento).

Oggetto di studio della bioinformatica sono il DNA, l'RNA e le proteine, i quali vengono codificate in seq. di caratteri (→ sequenziati in stringhe) tramite adeguati algoritmi. Le **stringhe** sono serie di bit, o di caratteri alfanumerici da quelli rappresentati, presente in una memoria o su un supporto.
- I *caratteri* che costituiscono le stringhe non sono infiniti, ma possono permettere "infinite" combinazioni (es. solo 4 nucleotidi per tutti i DNA).
- La *direzione*: andare dal 5'→3', quindi alla sua sx avrà un 5'-P ed alla sua destra 3'-OH; in una proteina la seq. Avrà alla sua sx un gruppo amminico e alla sua destra un gruppo COOH.

Le sequenze sono scritte da sx a dx, direzione coincidente alla direzione di trascrizione del DNA, senza interruzioni fra i caratteri: possono essere scritte tutte attaccate in minuscolo o maiuscolo oppure in maiuscolo tramite fosfati (p). Un'altra particolarità da rappresentare potrebbe essere la presenza, invece che del fosfato, di un'altra molecola, come gli atomi di S che indicano solfonazione. Un'altra rappresentazione è quella che prevede che non si debba rappresentare solo la sequenza ma qualche informazione in più.

>sequence of primer7
atcgtagtc

Questa è una rappresentazione di un primo file, *fast file*, che nasce dallo sviluppo di un programma detto FastA, con cui era necessario rappresentare in maniera abbastanza compatta una sequenza, avendo un minimo di informazione relativa alla sequenza e tutto il resto del file era per i nucleotidi.
E' un modo di rappresentazione molto intelligente perché permette di dedicare una stragrande maggioranza del file alla sequenza di DNA, però dà uno spazio pressoché infinito, nel quale conservare informazioni. L'unica regola è:
-Mettere il simbolo maggiore (>) all'inizio per indicare il testo completamente libero che può fornire tutta una serie di informazioni stabilendo per es. che il primo campo è l'organismo dal quale è stato derivato, il secondo la data in cui è stata fatta ecc…, il tutto separato da due punti.
Questo metodo è molto utile in quanto permette sia la lettura da parte di un computer, sia dall'uomo.

I file di sequenze possono essere letti da più programmi differenti in maniera simile. Di solito ogni programma ha un'unica funzione e la capacità di leggere il file allo stesso modo permette lo svolgimento delle differenti funzioni. Inoltre è possibile comprendere bene le sequenze leggendole e modificandole direttamente.

Le operazioni che si possono fare sulle sequenze (sequence manipulations) sono:
- Editing: aggiunta di caratteri, taglia e incolla, trova sequenze uguali (stesse operazioni sui testi)
- Complement (operazione di complemento), è un'operazione che permette di generare complementarietà nelle basi A-T, C-G.
-Composition
- Word Usage, Codon Usage
-Translation
-Plot ORF
-Pattern search, restriction enzymes
Per far sii che il programma funzioni è necessario che abbia differenti input in modo tale da generare un algoritmo che compia le azioni desiderate: leggi la sequenza (quindi importa la sequenza e leggila), determina le caselle delle seq. e le caselle della seq. corrispondente, se è una A scrivi T etc…
La sequenza generata dovrebbe essere la sequenza complementare, ma è letta e funzionale da destra verso sinistra, quindi deve essere riscritta al contrario. In realtà, non si tratta della sequenza complementare, perché alla macchina non è stata specificata la direzione di lettura, ma è una seq. Scritta al contrario, quindi quella non è la complementare, ma una inesistente. Quindi bisogna fare una seconda operazione: bisogna rappresentare questa seq in direzione 5'→3', prendendo l'ultimo dato inserito ed inserirlo in questo spazio come primo, il penultimo come secondo e così via per tutta la sequenza. In questo modo si ottiene la sequenza complementare.

Sequence Manipulation Tools → programmi per manipolare sequenze.
Packages, insieme di programmi in modo tale che un unico programma possa fare più operazioni:
- (UW) GCG → Genetics Computer Group
- Staden
- EMBOSS → The European Molecular Biology Open Software Suite
E' importante precisare che più programmi che complessivamente svolgono operazioni diverse ma finalizzate ad un obiettivo comune sono riuniti in ***package***. I programmi di uno stesso package rimangono separati, ma tipicamente utilizzano nei comandi una sintassi simile e permettono di scambiare dati con facilità. Un esempio di package è rappresentato da EMBOSS che è stato sviluppato per rispondere alle esigenze tipiche della comunità scientifica in ambito di biologia molecolare; è costituito da un gran numero di programmi originali sviluppati nell'ambito del progetto, ma integra anche diversi programmi generati separatamente di uso comune. Le aree di interesse sono piuttosto ampie e nell'ambito del package i vari programmi sono organizzati in gruppi omogenei per funzione:
- *Nucleic composition* è costituito da programmi che analizzano la composizione in basi di una sequenza e fanno predizioni su questa base,
-*Nucleic translation* permette la traduzione di seq. nucleotidiche in proteine con diverse modalità,
-*Nucleic restriction* contiene programmi utili alla ricerca di siti di restrizione.

Un'altra operazione da poter compiere è *calcolare la frequenza delle basi*. La frequenza è il numero di elementi favorevoli diviso il numero di eventi totali, permette di determinare la percentuale. In principio, l'attesa è che circa ¼ sia A, ¼ sia T, ¼ C, ¼ G; ma nella realtà, non tutte le sequenze di DNA hanno una frequenza per ogni nucleotide pari al 25%, ma molto spesso i numeri possono variare. E' interessante che anche solo il valore di contenuto, ovvero quanti nucleotidi le sequenze hanno, può avere una rilevanza biologica. Si utilizza il programma COMPSEQ (composition sequence) che conta il n di nucleotidi, potendo così confrontare la frequenza osservata con quella attesa (*expected frequency*) (Come funziona? Prendi la prima base se è una A, incrementa numero di A. Stampa il numero di A.)
Può capitare che la frequenza osservata sia maggiore di quella aspettata, bisogna valutare a seconda della sequenza se è solo una casualità o è presente un significato particolare.
Per confrontare le frequenze nucleotidiche di due sequenze differenti che non hanno la stessa lunghezza, bisogna calcolare il rapporto tra la frequenza obs/exp.
Il lavoro effettuato per calcolare il numero di ciascun nucleotide è abbastanza ovvio, ma non lo è quando si vuol contare il numero di coppie di nucleotide: *se ho una sequenza di n caratteri, le coppie saranno n-1.* Considerando tutte le possibili combinazioni di 2(n) che i 4(x) nucleotidi potevano assumere si hanno $n^x \rightarrow 2^4$ (AA, AT, AG, AC, CC, CA, CG, CT, GG, GC, GA, GT, TT, TA, TC, TG) .

Il gene utilizzato per la sequenza COMPSEQ è dell'mRNA. I *pattern*, ovvero le frequenze delle basi a coppie, ci può permettere di ipotizzare l'organismo a cui appartiene la sequenza.
Se predomina CC e CT quella sequenza probabilmente sarà un esone. Command line interface (EBI)

Compseq - Homo sapiens Bcl-2 mRNA

Word size 2
Total count 6029

# Word	Obs Count	Obs Frequency	Exp Frequency	Obs/Exp Frequency
AA	558	0.0925527	0.0625000	1.4808426
AC	295	0.0489302	0.0625000	0.7828827
AG	383	0.0635263	0.0625000	1.0164206
AT	432	0.0716537	0.0625000	1.1464588
CA	405	0.0671753	0.0625000	1.0748051
CC	339	0.0562282	0.0625000	0.8996517
CG	120	0.0199038	0.0625000	0.3184608
CT	383	0.0635263	0.0625000	1.0164206
GA	370	0.0613700	0.0625000	0.9819207
GC	289	0.0479350	0.0625000	0.7669597
GG	396	0.0656825	0.0625000	1.0509206
GT	330	0.0547354	0.0625000	0.8757671
TA	335	0.0555648	0.0625000	0.8890363
TC	324	0.0537403	0.0625000	0.8598441
TG	486	0.0806104	0.0625000	1.2897661

Compseq Homo sapiens type XV collagen (COL15A1) gene, exon 6

Word size 2
Total count 5063

# Word	Obs Count	Obs Frequency	Exp Frequency	Obs/Exp Frequency
AA	257	0.0507604	0.0625000	0.8121667
AC	240	0.0474027	0.0625000	0.7584436
AG	350	0.0691290	0.0625000	1.1060636
AT	244	0.0481928	0.0625000	0.7710843
CA	402	0.0793996	0.0625000	1.2703930
CC	479	0.0946079	0.0625000	1.5137270
CG	65	0.0128382	0.0625000	0.2054118
CT	456	0.0900652	0.0625000	1.4410429
GA	272	0.0537231	0.0625000	0.8595694
GC	313	0.0618211	0.0625000	0.9891369
GG	355	0.0701165	0.0625000	1.1218645
GT	264	0.0521430	0.0625000	0.8342880
TA	159	0.0314043	0.0625000	0.5024689
TC	370	0.0730792	0.0625000	1.1692672
TG	434	0.0857199	0.0625000	1.3715189
TT	403	0.0795971	0.0625000	1.2735532

Compseq Homo sapiens gene for osteonidogen, intron 3

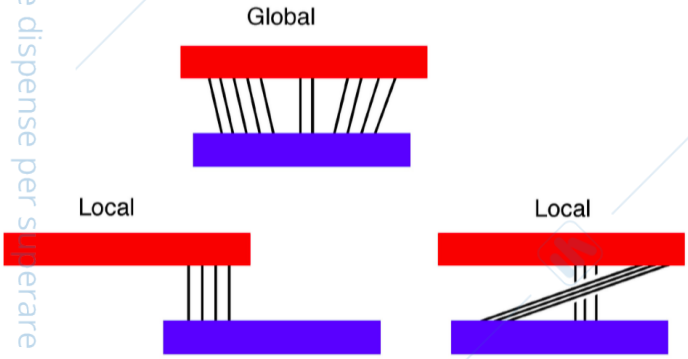
Word size 2
Total count 5812

# Word	Obs Count	Obs Frequency	Exp Frequency	Obs/Exp Frequency
AA	516	0.0887818	0.0625000	1.4205093
AC	306	0.0526497	0.0625000	0.8423950
AG	374	0.0643496	0.0625000	1.0295939
AT	480	0.0825877	0.0625000	1.3214040
CA	431	0.0741569	0.0625000	1.1865107
CC	267	0.0459394	0.0625000	0.7350310
CG	30	0.0051617	0.0625000	0.0825877
CT	392	0.0674467	0.0625000	1.0791466
GA	287	0.0493806	0.0625000	0.7900895
GC	210	0.0361321	0.0625000	0.5781142
GG	276	0.0474880	0.0625000	0.7598073
GT	319	0.0548864	0.0625000	0.8781831
TA	442	0.0760496	0.0625000	1.2167928
TC	337	0.0579835	0.0625000	0.9277357
TG	412	0.0708878	0.0625000	1.1342051
TT	733	0.1261184	0.0625000	2.0178940

Allineamento delle sequenze

L2

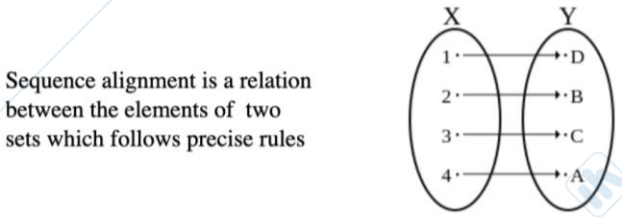
Qualora si abbiano a disposizione due sequenze di geni o proteine, il primo fondamentale passo per studiare l'evoluzione delle due sequenze e stabilire se tra di esse sussiste una *relazione di omologia*, cioè di discendenza da un comune antenato. Infatti qualsiasi evento spaziale o temporale che porti alla separazione fisica di due sequenze (speciazione, duplicazione genica) a partire da un comune antenato fa sì che le due sequenze smettano di scambiarsi il reciproco contenuto di informazione e che inizino a evolvere differentemente l'una dall'altra accumulando mutazioni in maniera indipendente. Alcune posizioni delle sequenze potranno conservare quindi il tratto caratteristico del comune antenato, altre potranno mutare conferendo possibilmente caratteristiche diverse alle 2 sequenze. Poiché naturalmente non è possibile seguire direttamente l'evoluzione di due o più sequenze l'unico metodo di cui si dispone per stabilire una relazione di omologia è il confronto delle sequenze attraverso un allineamento, ovviamente non tutti gli allineamenti di 2 sequenze hanno lo stesso valore nel darci informazioni sulle relazioni evolutive, infatti si cerca l'allineamento che identifica La corrispondenza biunivoca tra residui (nucleotidi o aa) che riflette nella maniera più accurata possibile l'accumulo di mutazioni e quindi la storia evolutiva delle sequenze; in molti casi si cerca l'allineamento che permette il minor numero di cambiamenti per passare da una sequenza ad un'altra (minor numero di indel e maggior numero di match) che rappresenta in pratica il percorso evolutivo più breve. Allineare due sequenze significa paragonare due sequenze e verificare se vi sono analogie (match) fra di esse.

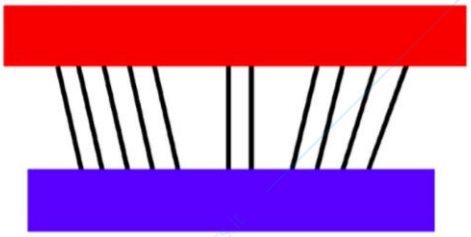


Due sequenze simili possono essere per esempio sequenze di uno stesso gene ma di specie diverse, quindi sequenze con analogie sono sequenze quasi identiche. Ovviamente le sequenze che si confrontano non sono coincidenti tramite la prima base con la prima base dell'altra sequenza, ma possono coincidere in punti diversi, quindi gli allineamenti possibili sono differenti. Ci sono differenti tipi di allineamento:
- globale, comprende grande parte di entrambe le sequenze;
- locale, solo alcuni pezzi di alcune sequenze sono simili, anche in maniera "mischiata".

L'allineamento è una relazione biunivoca, a un elemento di un insieme corrisponde uno ed un solo elemento dell'insieme corrispondente. Inoltre l'allineamento non può "incrociarsi", deve essere caratterizzato da linee "parallele", ciò significa che ci vuole un ordine sequenziale.

```
Seq 1  ATGCGTGTGTGCATGCAATGCGTGA
Seq 2  TGTGCATGCAAT

       ATGCGTGTGTGCATGCAATGCGTGA
       TGTGCATGCAAT

       ATGCGTGTGTGCATGCAATGCGTGA
         TGTGCATGCAAT

       ATGCGTGTGTGCATGCAATGCGTGA
           TGTGCATGCAAT

       .....

       ATGCGTGTGTGCATGCAATGCGTGA
                  ***********
                  TGTGCATGCAAT
```

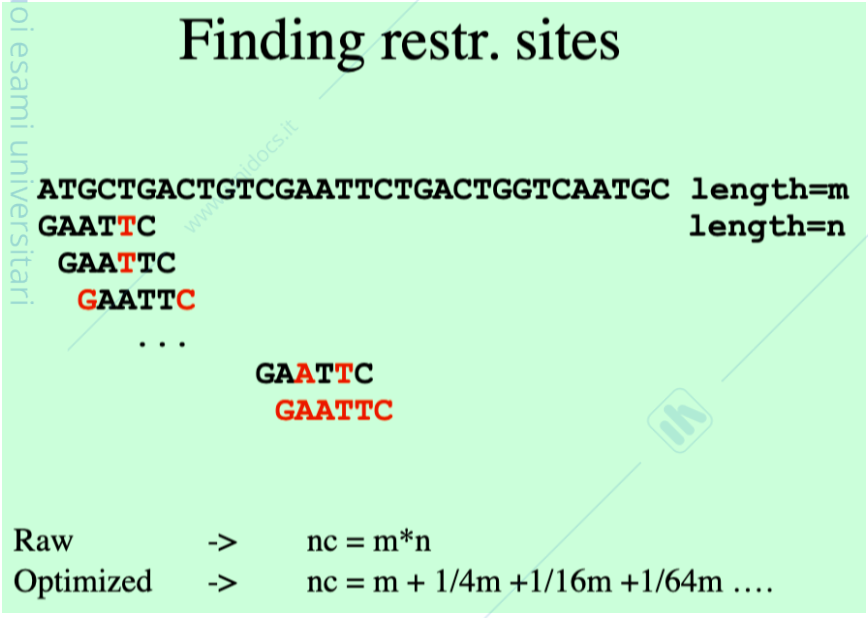




Esempio di allineamento

Approccio a forza bruta

Per cercare allineamento nelle sequenze si parte dalla prima posizione di entrambe. In questo caso si ha un evento inusuale ovvero ci sono zero appaiamenti. Tipicamente un allineamento cattivo non ha zero corrispondenze, un minimo ne ha sempre. E' difficile che si abbia un allineamento pari a zero, infatti prendendo in considerazione che tutte e due le sequenze sono composte da 4 basi è improbabile che neanche una volta ci sia la stessa base in una determinata posizione, soprattutto quando si parla di lunghe sequenze. *"Il nostro 0 non è mai 0"* ma è circa 10-20%.
Dopodiché si continua a cercare corrispondenze, la seconda sequenza shifterà di una base, quindi inizierà ad essere confrontata dalla seconda base della prima sequenza. Questo processo continuerà fino ad ottenere un allineamento caratterizzato dalla totalità di corrispondenze, o quasi.
Ovviamente il miglior appaiamento possibile, corrisponde ad un'analogia totale.
Gli appaiamenti sono buoni o cattivi a seconda di cosa stiamo cercando, bisogna valutare da caso a caso. Infatti a seconda di come si soddisfano i criteri stabiliti dalla domanda, da cosa si cerca, si può determinare quale sia il miglior appaiamento possibile.
Ad esempio, a volte si cerca un appaiamento perfetto, altre volte inserzioni e delezioni.
Con gli appaiamenti si misura la somiglianza, non la relazione evolutiva, quindi non si può stabilire tramite l'identità la specie a cui appartiene la sequenza. La similarità è un parametro quantitativo, cioè indica la percentuale di residui identici tra le due sequenze, mentre l'omologia indica l'esistenza di un antenato comune tra 2 geni o proteine. Quindi è un parametro qualitativo e se è vero che nella stragrande maggioranza dei casi l'omologia implica una similarità, non è detto il contrario.
Conoscere la *similarità* tra due sequenze comprende svolgere due azioni:
-Trovare l'allineamento ottimale
-Individuare il metodo numerico per ottenere la percentuale di similarità
Vengono così esaminate tutte le possibili soluzioni, esse saranno successivamente valutate ed infine confrontate per trovare quella con la risoluzione migliore. Questa metodica può riscontrare dei problemi ma in genere è molto attendibile; è un approccio tipicamente utilizzato per riconoscere i siti di restrizione all'interno di una sequenza.
Esempio. Cercare siti di restrizione GAATTC

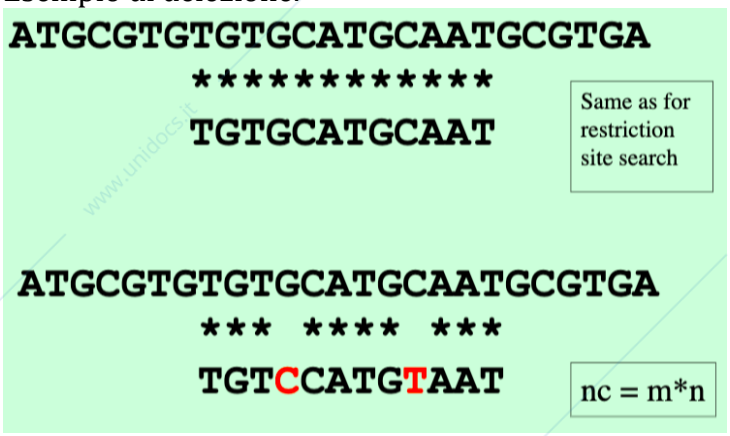


In una sequenza quanti siti di restrizione troviamo?
Bisogna tener conto di quanto sia fattibile la procedura a seconda della velocità che impiega a compierla. Se ci mette troppo tempo non ne vale la pena. La velocità dipende sia dal computer, sia dal programma, sia dall'esperimento. A livello dell'esperimento bisogna dunque valutare quanti confronti si devono compiere. L'operazione più impegnativa è il confronto.
Il numero di confronto è l'elemento che determina la rapidità a svolgere una determinata azione. *Il numero di confronto è pari al numero di basi della 1° sequenza moltiplicato per il numero di basi della 2° sequenza.* $\boldsymbol{N_c=MxN}$

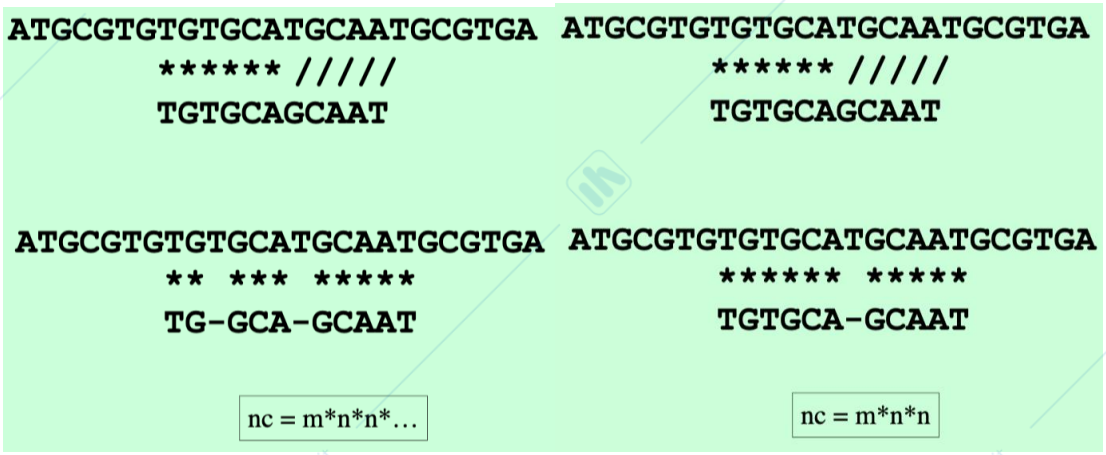
Il numero di allineamenti possibili è dato da: (M-N) +1.
Nel caso di GAATTC si devono compiere 6 confronti, o meglio 6 x n (regola m x n) dove n è la lunghezza della sequenza con la quale si confronta in questo caso GAATTC. (6 x n è un'approssimazione in quanto risulteranno sempre qualche confronto in più o in meno).
Quando si cerca identità perfetta si possono inserire ulteriori parametri per velocizzare il processo.

Per esempio inizio cercando la G, che possiamo trovare 1 volta su 4, dopodiché cerca la A, la quale è possibile trovare sempre 1 volta su4. Se A si trova si passa al terzo confronto, se no si ritorna al primo cercando di nuovo la G. Per ottimizzare i confronti, basta considerare che appunto nel DNA le basi sono 4, seguendo la regola N_c= m + ¼ m + 1/16 m+ 1/64 m
Dato che con GAATTC sono necessari 6 confronti si ha che ogni confronto è ¼, si ottiene che il confronto totale è 1/4096.
Quello descritto fin ora è un caso particolare.
In un caso generico non cerchiamo identità perfetta ma la maggior somiglianza per esempio 8 su 10. Si devono quindi valutare tutti i confronti e determinare il migliore a seconda dei criteri stabiliti.
E' necessario valutare sempre tutta la sequenza perché è sempre possibile trovare un appaiamento migliore, quindi è sempre necessaria la **regola m x n**.
3 miliardi di confronti li possiamo fare in 1 secondo.

Esempio di delezione.



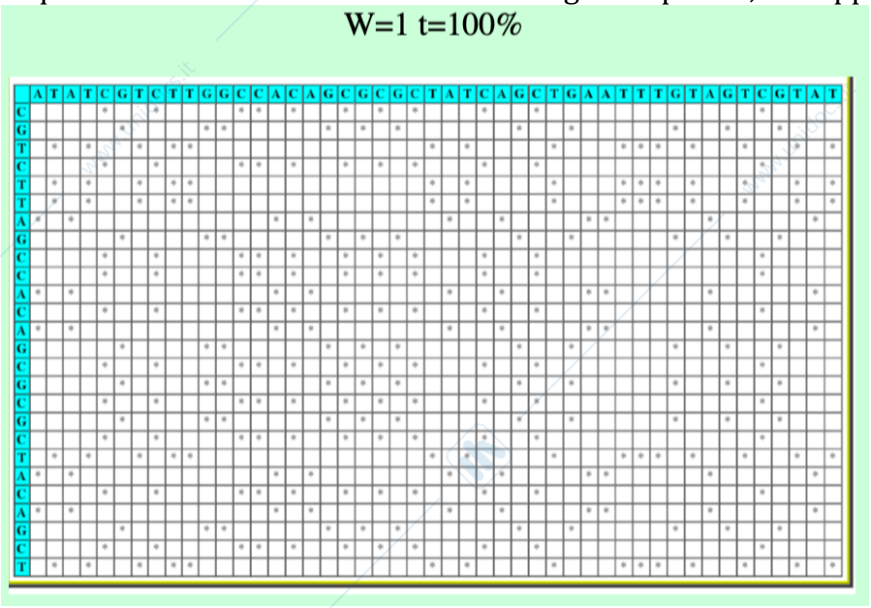
Se cerchiamo appaiamenti che presentano delezioni ci sarà un buco (gap) nella sequenza che stiamo confrontando con la principale, ma gli appaiamenti saranno tutti soddisfatti. Si avrà un aumento di similarità con l'introduzione del gap.
Ciò però comporta molte conseguenze.
match→ corrispondenza
mismatch → assenza di corrispondenza
Se si introduce il concetto di gap non varrà più la *regola m x n* ma è necessaria la **regola m x n x n**.
I gap si possono trovare in varie posizioni, per poterlo collocare in una posizione precisa, vuol dire moltiplicare x4. Per ogni allineamento è possibile trovare il numero di gap, ne consegue che il numero di confronto (N_c) sarà N_c= m x n x n; dove l'ultima n rappresenta il numero di gap presenti per allineamento.



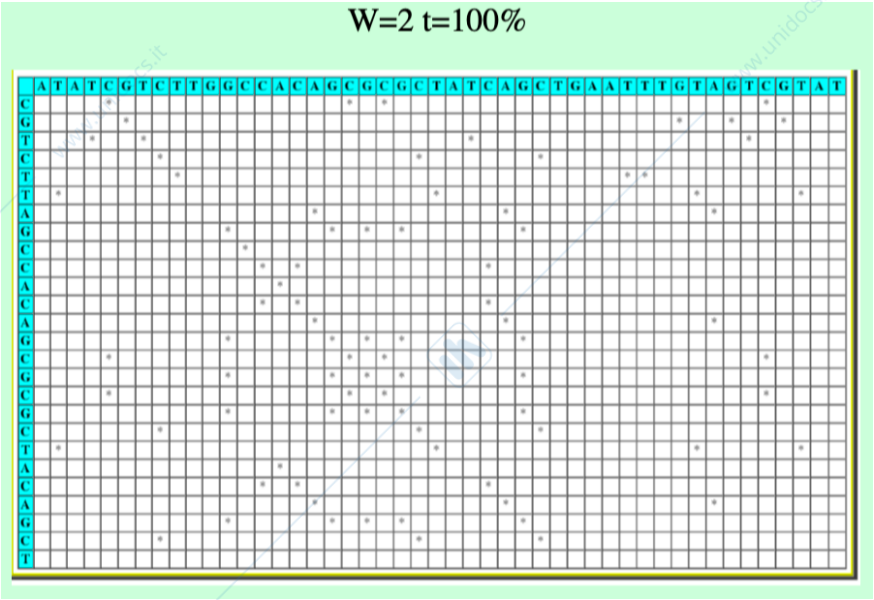
Quindi il tempo aumenta perché si cercano più appaiamenti.
Di solito gli appaiamenti sono indicati con l'asterisco, gli slash non vengono usati ma in questo caso ci aiutano a comprendere la corrispondenza, come se fossero frecce, e la presenza di gap.
Se vi sono due gap è necessario rimoltiplicare x n. Ciò ci porterà via via a una situazione non più

gestibile, in quanto le delezioni potrebbero anche essere 80. Si potrebbe fare, ma la soluzione non arriverebbe in tempi accettabili.

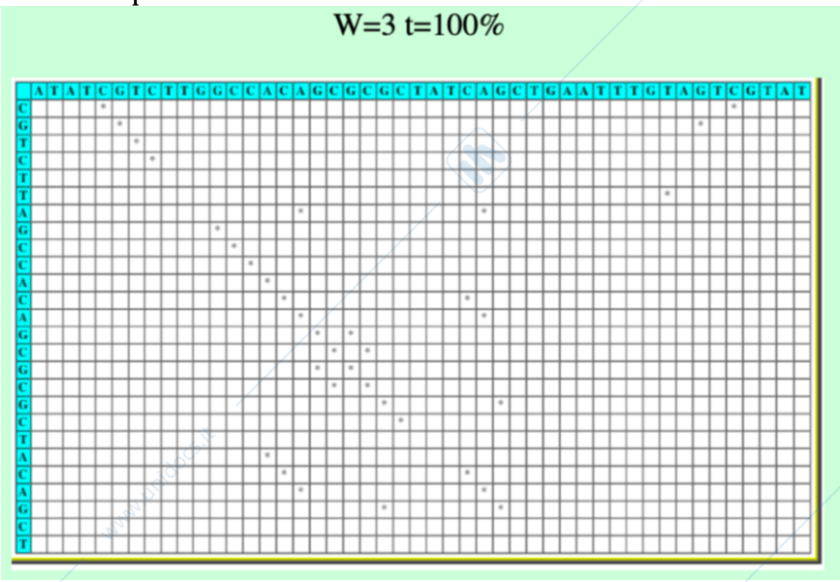
Per evitare ciò si valuta i confronti effettivamente necessari, ovvero il numero minore possibile.
Si confrontano quindi ogni base della prima sequenza con ognuna delle basi della seconda sequenza, generando un grafico tipo battaglia navale (in questo modo non si introducono gap).
Si può notare che quindi ci saranno solo $N_c = m \times n$. In sostanza MxN geometricamente è l'area di un rettangolo nel quale M e N sono due lati. All'interno di questo rettangolo si avranno tante caselle e ce ne sarà una per ogni coppia di basi. Con * si rappresentano le coppie di basi uguali. In questo modo si ha un insieme di micro allineamenti ma nessuno di essi esprimerà l'allineamento ottimale.
Graficamente è possibile verificare l'appaiamento ottimale, ovvero la diagonale. Con questo metodo il gap è espresso come un salto di diagonale. Le **dot matrix** sono molto convenienti per l'immediatezza e l'intuitività con cui spesso riescono a fornire indicazioni utili, tuttavia non esprimono un valore dell'allineamento, ma solo una rappresentazione grafica. Esse consentono di individuare similarità di sequenza e un facile riconoscimento di regioni ripetute, che appaiono come segmenti paralleli.



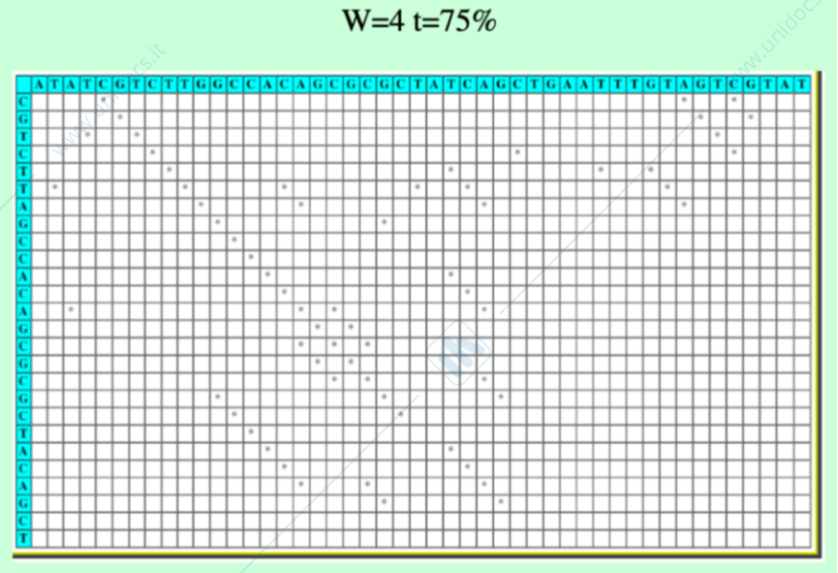
Nonostante ci siano differenti appaiamenti, tutte le diagonali individuabili sono allineamenti. Tuttavia è possibile anche avere la necessità di introdurre un gap, quando una diagonale sarebbe completata da una successiva, ciò lo si fa tramite un gap.
Per evitare di analizzare tutto il grafico e anche gli appaiamenti che formano il rumore, ovvero non sono utili al nostro fine, si può rendere la ricerca più selettiva. Per esempio non si valuta più gli appaiamenti uno per uno ma a due, sarà sempre breve il tempo con cui lo si fa, ma la diagonale sarà nettamente più evidente. Facendo così i mismatch potrebbero essere ingranditi, ma ne vale la pena.



La valutazione da due può passare anche a tre ma man mano il segnale si inizierà a perdere, e non conviene più aumentare la ricerca.



Per evitare che il segnale si perda si possono inserire altre differenti variabili, come per esempio inserendo un * ogni 3 su 4, 4 su 5 etc. In questo modo il segnale sarà nettamente più evidente, mentre il rumore di fondo sarà ridotto. Inoltre si potrà così anche diminuire il mismatch.

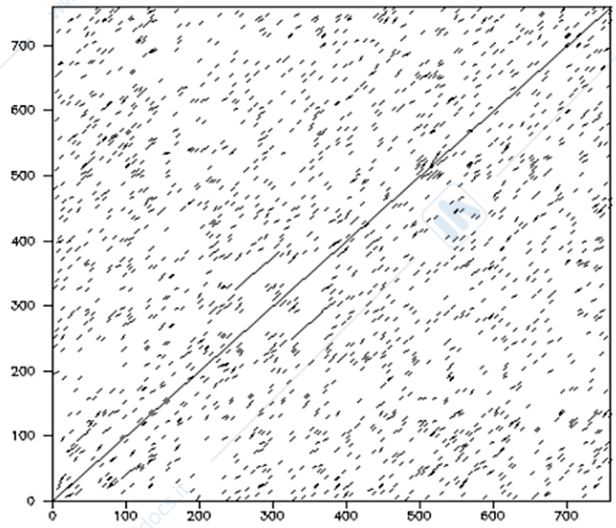


Per vedere meglio l'allineamento le azioni da svolgere sono una delle due:
-Aumentare il segnale
-Ridurre il rumore di fondo
Non è importante quanto sia alto il segnale o quanto sia basso il rumore di fondo, ma qual è il rapporto tra segnale e rumore. Quanto più alto è il segnale e basso il rumore tanto è migliore il risultato. Per *rumore di fondo* si intende il fenomeno generato da match distribuiti sul piano della matrice al di fuori della diagonale; per *segnale* vengono indicate le zone di somiglianza significative. Il difetto di questo metodo è che non si possono notare le differenze tra una sequenza allineata al 100% e una sequenza che ha uno o più mismatch. Quindi con questo metodo è stato rafforzato il rapporto segnale/rumore, si è riusciti notevolmente ad identificare l'appaiamento tra sequenze ma sono state perse le info circa la qualità dell'appaiamento, e risulta difficile distinguere cose perfettamente identiche e quelle che contengono un mismatch.
Le sequenze si definiscono collineari se l'appaiamento inizia dalla prima base di ognuna.

Dot Matrix L3

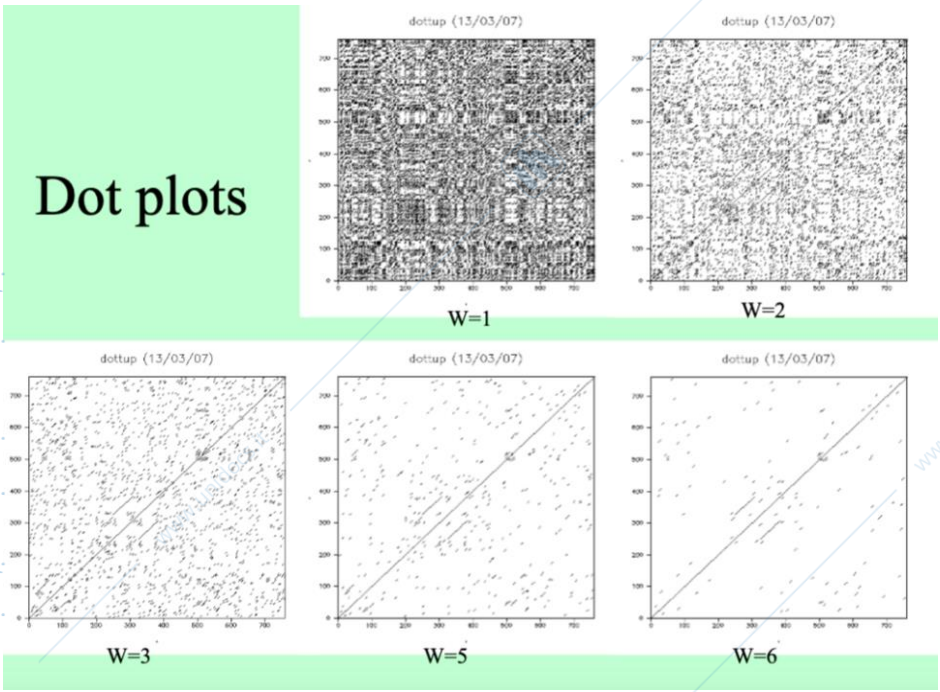
Una **dot-matrix** è una matrice di punti usata per visualizzare caratteri grafici, simboli ed immagini. La dot matrix (matrice a punti o dot plot) è stato il primo semplice sistema di visualizzazione di allineamenti(1970) ed è un sistema relativamente semplice capace di identificare le zone di somiglianza locale tra due sequenze e di visualizzare una mappa grafica della loro localizzazione. Questo metodo non produce direttamente un allineamento tra le 2 sequenze, bensì una mappa delle loro zone di somiglianza e della localizzazione di duplicazioni, inserzioni, delezioni, inversioni ecc. Le 2 sequenze da confrontare sono ai margini di una matrice bidimensionale in cui le 2 sequenze sono scritte una in alto da sinistra a destra e l'altra a sinistra dall'alto in basso. Se 2 residui (aa o nucleotidi), di sequenze diverse, corrispondenti alla stessa casella della matrice sono diversi questa differenza (**mismatch**) sarà visualizzata da un interruzione della diagonale. I **gap** (ad es delezione: assenza di una parola in una stringa rispetto all'altra) appaiono come salti in diagonale (diagonale spezzata). Le sequenze ripetute appaiono come segmenti diagonali paralleli.
Alcuni aspetti dell'analisi visuale di una dot matrix: se analizzassimo due sequenze identiche (cioè usiamo la stessa sequenza sia come sequenza orizzontale che come sequenza verticale) allora otterremmo una diagonale continua che parte dall'angolo in alto a sinistra per arrivare a quello in basso a destra. Ovviamente oltre alla diagonale troveremmo molti altri puntini, sparsi nell'area della matrice al di fuori della diagonale principale. Si consideri che ci sono 20 aa diversi, quindi in una sequenza casuale ci dovremmo aspettare una casella positiva ogni 20. Similmente, con acidi nucleici dovremmo aspettarci una casella positiva ogni 4, con un notevole rumore di fondo. Quindi con il termine di "*rumore di fondo*" indichiamo quel fenomeno generato da match distribuiti sul piano della matrice al di fuori delle diagonali e che deriva dal fatto che, dato un insieme finito di caratteri componenti le stringhe, è inevitabile osservare corrispondenze casuali di simboli.

Esempio Matrice nella quale sono state confrontate due sequenze di 700 basi (assenza del problema di overbooking, sovrapposizione del singolo punto, ma è impossibile un riconoscimento base per base). Confrontando in tal modo le due sequenze, si è effettuato un ***dotuup*** (matrice a soglia 100% e w=3)



La prima differenza con i grafici precedenti: vengono confrontate proteine vere infatti ci sono ordini di grandezza molto più grandi (700).A livello grafico non sono nettamente differenziati, non si può dedicare un punto a ciascun amminoacido in quando sono troppi. Non ci sono più asterischi ma punti. Inoltre la diagonale presenta degli indel che sembrerebbero dei salti ma è solo un errore a livello grafico in quanto le diagonali in bioinformatica non sono mai precise a 45°. Data l'elevata quantità ci sarebbero eccessive somiglianze e molto rumore quindi è meglio confrontare sezioni ristrette per volta.

Le sequenze sono confrontabili e molto simili.
Possiamo avere sia grafici chiari che scuri, modificando il numero di basi da confrontare (window) o la soglia (Threshold) per poter stabilire quanto il background sia presente o meno.

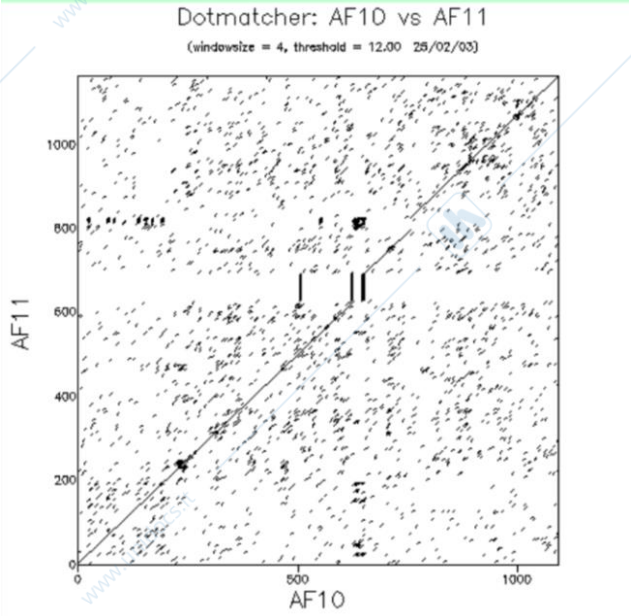


W=Window , T=Threshold (soglia).

- Da un lato si aumenta W finché non si perde troppo il segnale e quindi, il background comincia a non essere più molto significativo, perché è tutto bianco.
- Modificando T, si fondono tutti i punti e si eliminano addirittura i singoli punti di mismatch: è utili per garantire visibilità.

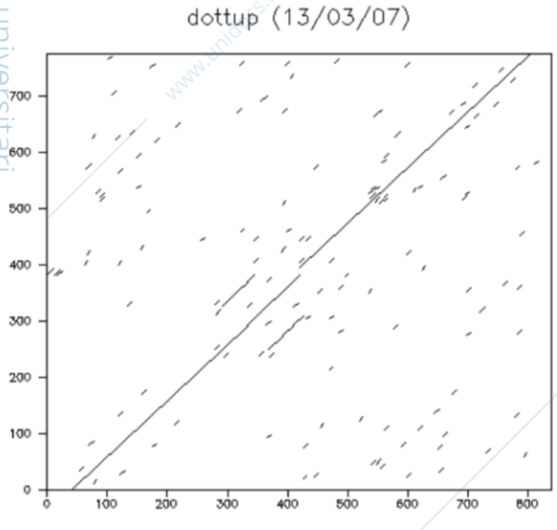
Per esempio, una sequenza difficilmente identificabile in mezzo a tante altre simili con uno o più punti di interruzione, potrebbe essere recuperata continuando a "buttare giù" il background.

Confronto AF11 e AF10



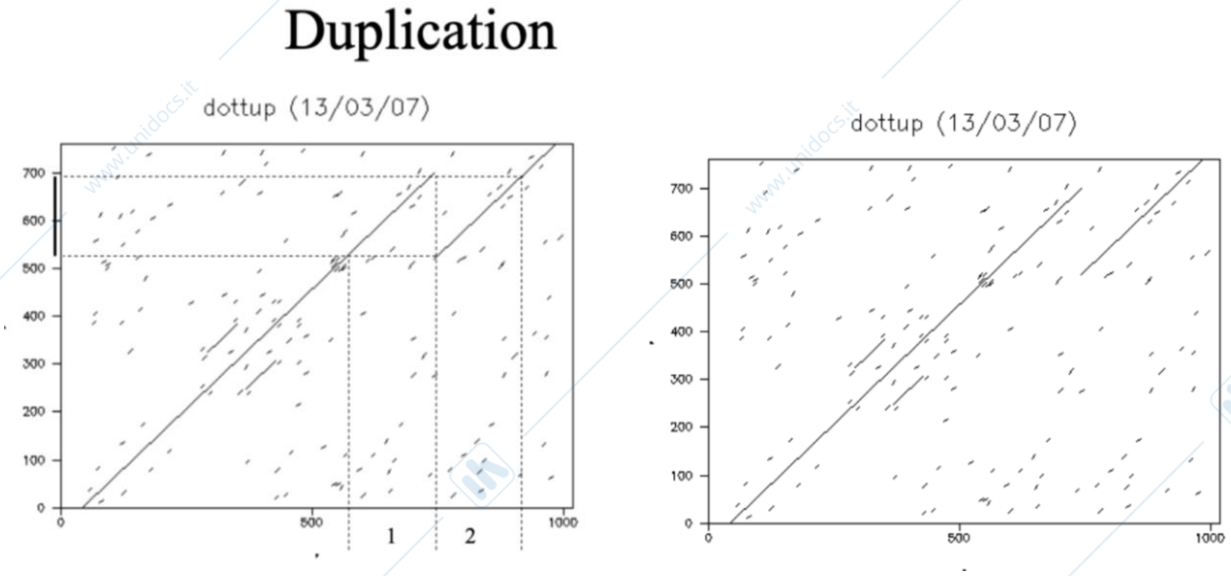
Bisogna notare prima di tutto wordsize = 4, ovvero che si stanno confrontando parole da 4; mentre t=12.
→Non è una soglia del 100%, ma del 50-60%
Le due sequenze sono diverse, perché non c'è una diagonale continua ma, sono collineari (esistono tratti di non similarità).
Non ci sono mismatch, ci sono gap (tra 600 e 800).
La AF11 ha un'inserzione tra 500 e 600, assente in AF10: esiste un pattern di 4 basi che si ripete e ciò è rappresentato dalla fascia bianca.
La fascia vuota rappresenta che non c'è nessun caso in cui il grado di somiglianza è tale da superare la soglia.
Ciò è attribuibile al fatto che dato che si confrontano parole di 4, c'è 1/160 000 un appaiamento "perfetto", questo evento spiega anche le tre linee tra 600 e 800.

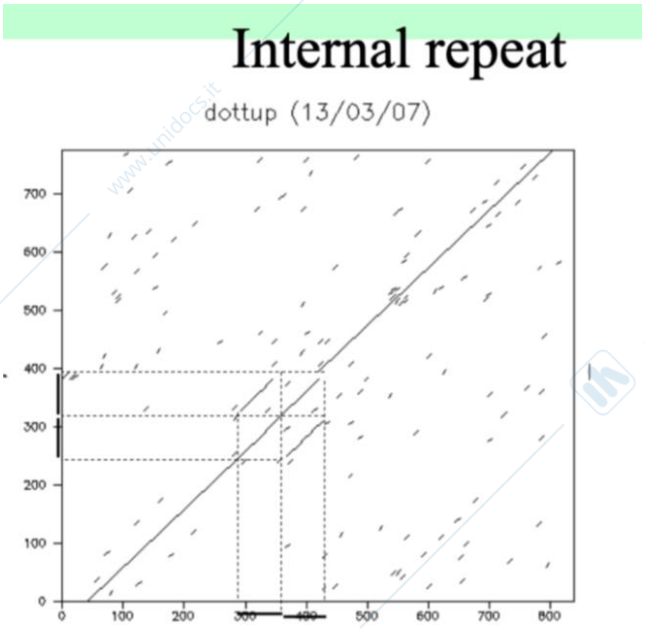
Le sequenze ripetute (a bassa complessità) "disturbano" molto il processo di ricerca di similarità. Esse danno infatti luogo a moltissimi match tra di esse, visto che nei genomi specialmente quelli degli organismi superiori (vertebrati, piante, ecc.) le sequenze ripetute occupano una grandissima percentuale. In genere nei programmi per la ricerca di similarità in database viene effettuato un mascheramento delle regioni ripetute (low complexity regions) proprio per evitare che la presenza di eventuali repeat nella query catturi come prime hit più simili tutte le sequenze ripetute simili ad essa contenute nei database. (Nel genoma umano: le seq. ripetute costituiscono più del 50% del genoma).



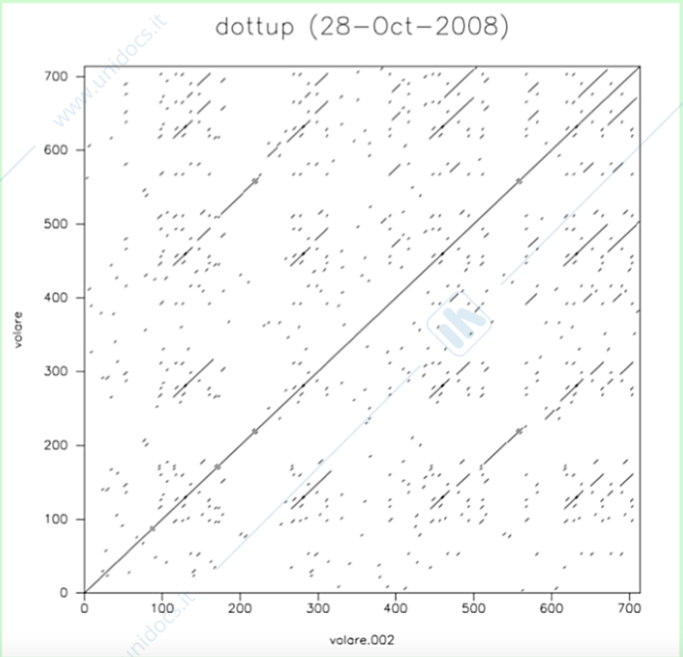
In questo caso non sono collineari all'estremità sinistra, quella orizzontale ha un tratto in più a sinistra intorno a 400.
Lo stesso vale per destra.
E' sempre utilizzata una soglia, infatti c'è poco rumore.
A livello 400 è presente un'inserzione che coincide con uno spazio bianco nella diagonale.

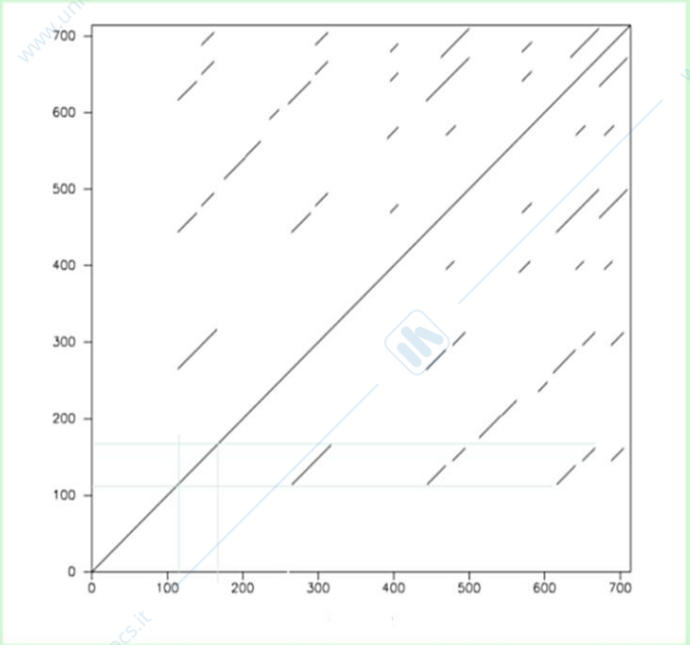
In quest'altro esempio c'è **inserzione** perché c'è un salto di diagonale. L'inserzione è nella sequenza "x" perché c'è un tratto in più. Ciò lo si capisce dallo shift a destra della diagonale.
In quale sequenza c'è stata un'inserzione?
1. Per capire quale delle due sequenze contiene basi che non si appaiano all'altra, bisogna analizzare le controparti (tracciando linee verticali e orizzontali per vedere se confluiscono sulla diagonale). Alla fine dell'analisi si nota che c'è un pezzo in più proprio sulla sequenza in "x".
2. Vedo la diagonale più lunga e per completare l'allineamento, devo spostarmi dall'una all'altra diagonale che si trova un certo numero di basi più avanti sulla sequenza "x" che avrà quindi un pezzo in più che corrisponde al tratto di differenza tra le due diagonali.
Es. il punto 600 su "y" su "x" corrisponde a due punti nei tratti 1 e 2, quella è sicuramente un'inserzione, in particolare una **duplicazione**.
Si ha 100% di identità, poiché saltando da una diagonale all'altra, si ha comunque 100% di identità.
Le 2 sequenze contengono le stesse basi, c'è un'unica inserzione, altrimenti ci sarebbe un'unica linea.
Diagonali parallele → DUPLICAZIONE





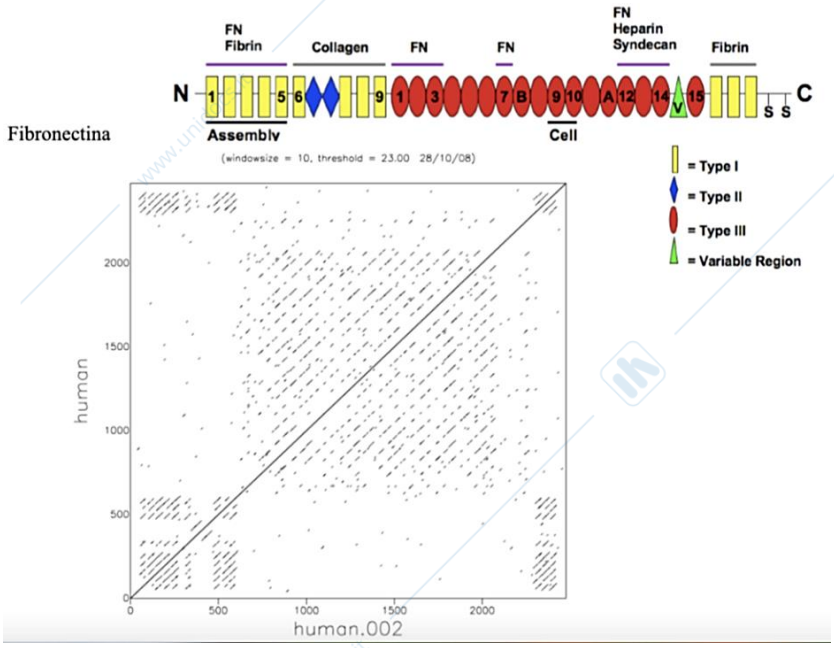
E' un pattern caratteristico per avere un tratto ripetuto due volte. Le due sequenze sono uguali ma sono presenti anche piccole diagonali che indicano punti di ripetizione. Se confrontate una sequenza con sé stessa, avremmo la diagonale senza salto e potremmo osservare i domini ripetuti.





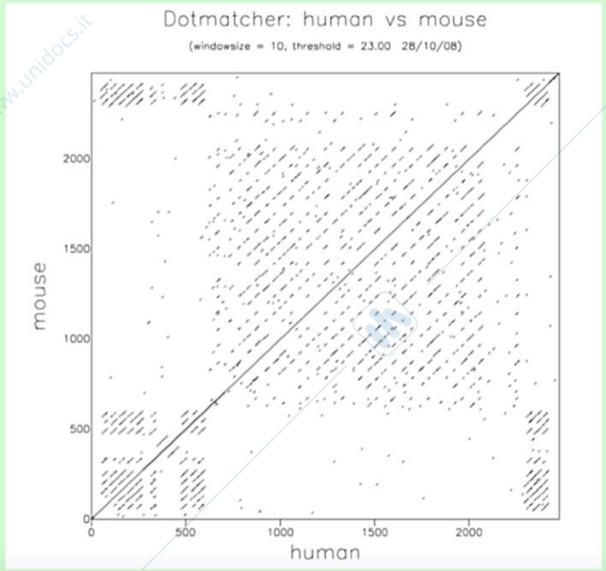
Le due sequenze sono sicuramente collineari, perché iniziano in un punto e finiscono nello stesso punto. Ci sono delle duplicazioni: linee parallele alla diagonale principale, disposte l'una a destra e l'altra a sinistra. Ad es. , da 100 a 200 uguale al tratto su "x" tra 250 e 350. *Quante volte è duplicato?* Sulla stessa linea ci sono anche altri pezzetti che stabiliscono che ci sono 4 copie e mezzo dello stesso tratto, dove nella terza e nella quarta ci sono dei salti di diagonale e mismatch, mentre le loro estremità sono uguali. Ciò accade in maniera speculare anche dall'altro lato della diagonale con il pezzetto all'altezza di 300 in verticale, e si possono notare anche le altre 4 e mezzo corrispondenze. Ci sono tante tante tante ripetizioni.

Human vs Human

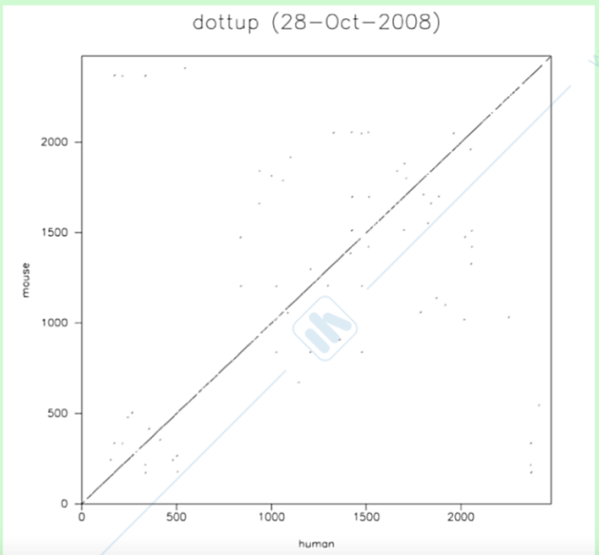


Proteina umana (fibronectina) rispetto a sé stessa. Il repeat più evidente è il blocco centrale: sono tante duplicazioni. Per sapere i repeat totali si contano le diagonali qui sono circa 15-16. Altro repeat lo abbiamo nella zona presso i 500 e una oltre i 200.
Esistono dei type one repeat, dei types 2, dei types 3 repeat. Quindi il blocco centrale definisce un tipo di sequenza ripetuta, l'altro blocco in basso a sx, più articolato, più quello all'estremità definisce un altro blocco di sequenza ripetuta. *Il pattern è sempre simmetrico alla linea centrale, ma un pattern in cui si ha una linea centrale e una linea a dx e a sx, corrisponde a 2 duplicazioni.*

Fibronectina umana vs murina

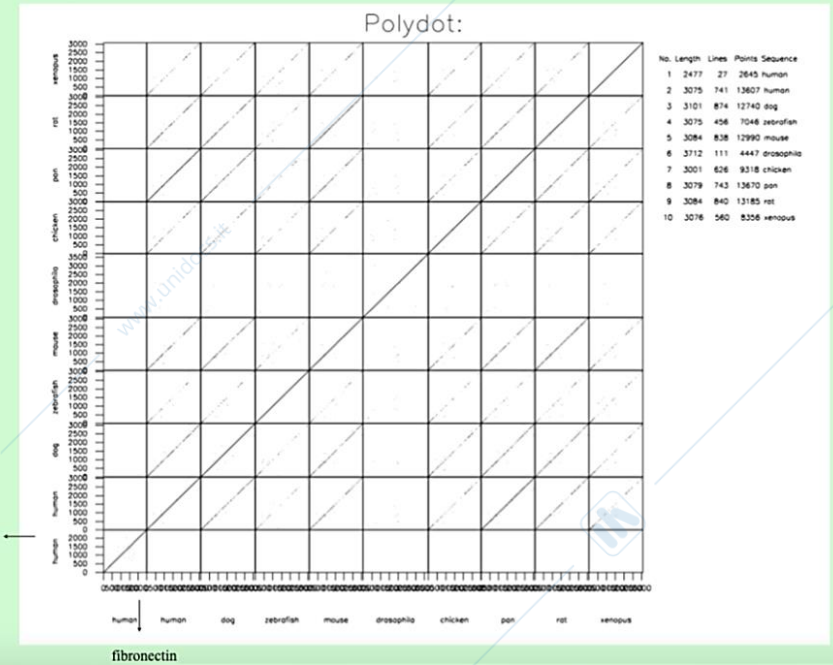


Fibronectina umana vs murina; window maggiore



E' possibile che la fibronectina dell'uomo sia al 100% uguale a quella del topo?
Se si fa lo stesso ma uomo vs topo, sarà molto simile ma ci saranno anche differenze.
Diagonale spezzettata, interrotta→ differenze.
La proteina umana e quella di topo hanno diversi punti di differenza e l'evento di replicazione che crea tratti di omologia interni, probabilmente è dovuto ad una duplicazione genica che è avvenuta molto tempo prima che il topo e l'uomo si dividessero perché è comune a tutti gli organismi che possiedono tale proteina (antenato comune).
(L'insulina di maiale e quella dell'uomo sono uguali, infatti in passato era utilizzata nei diabetici senza generare reazioni collaterali.)

Laminine di specie diverse

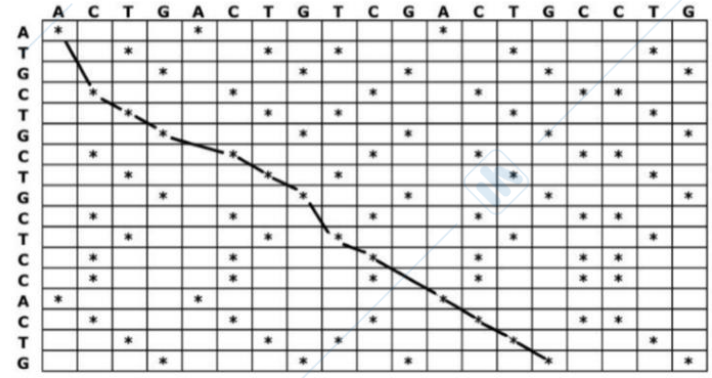


Questa è una dot matrix che confronta la fibronectina umana e laminine di tutte le specie messe una dopo l'altra. La diagonale principale è costituita dal confronto della proteina di uno specifico organismo con un altro della stessa specie (umano-umano, topo-topo, scimpanzé-scimpanzé ecc…).
Quali proteine sono più simili tra loro?
Quella di uomo e pan (scimpanzé), topo e ratto. La zebra è uguale a sé stessa ma è diversa da tutte le altre. Aumentando il grado di sensibilità, si noteranno gradi di similarità che ora non sono visibili, più linee continue. Aumentando la W e la T, si vedrà meno similarità.

Scoring Matrix

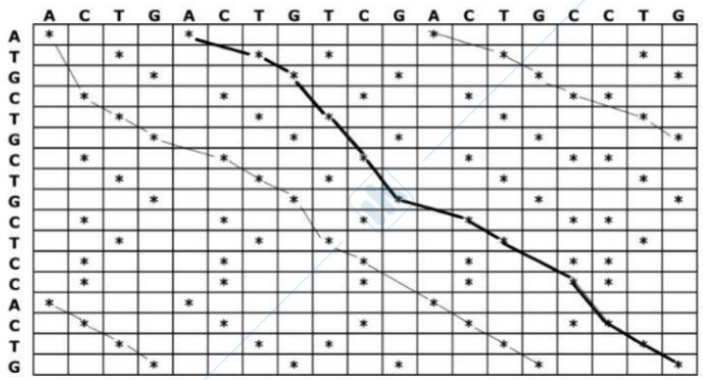

```
A--CTGACTG-TCGACTGCCTG
*  **  *** ** ****
ATGCTG-CTGCTCCACTG
```

Dot plot - one path




```
ACTGACTG-T-C-GACTGC-CTG
 * ** * * * ** * ***
A-TGCTGCTG-CTCCACTG
```

Dot plot - one path



Lo scopo dell'allineamento di sequenze è di individuare gli elementi più simili tra di loro di due sequenze. Questa similarità deve essere valutata in qualche modo. Per fare ciò, vengono utilizzate le matrici di punteggio che contengono un valore per ogni possibile sostituzione. Il punteggio di allineamento è la somma dei valori della matrice per ogni coppia amminoacidica allineate. Per i gaps, è necessaria una speciale gap score (di cui la più semplice è quella di assegnare una valore costante di penalità per ogni indel). L'allineamento ottimale è l'unico che massimizza il punteggio di allineamento.
PAM matrices sono la famiglia più comune delle score matrices. PAM sta per Percent Accepted Mutations, dove "accepted" significa che la mutazione è inserita all'interno della sequenza in esame.
Le matrici PAM possono contenere valori positivi e negativi: se il punteggio di allineamento è maggiore di 0, significa che le due sequenze sono correlate (considerando quella matrice di punteggio); se è negativo, le due sequenze non sono in relazione.
La score matrix non viene utilizzata per trovare la similarità tra le sequenze, non è nemmeno un metodo di confronto tra due proteine, è semplicemente un file, una tabella nella quale sono conservati tutti i punteggi che sono pianificati per effettuare un determinato confronto tra due sequenze.

Esempio: In questa score matrix, cambiando una A→A, si ha 1; il cambio tra una A→G è meno traumatico rispetto A→C, dato che è avvenuta una sostituzione di una base pirimidinica con una base della stessa natura chimica, per cui varrà 0.5. Questa metodica non è sempre vera, dipende dalle sequenze da confrontare e a seconda del criterio con cui devono essere discriminate l'una dall'altra. Un'analisi del DNA dal punto di vista dell'ingombro sterico può utilizzare questa scoring matrix; nel caso bisognasse valutare la capacità di traduzione di una proteina, non andrebbe bene.

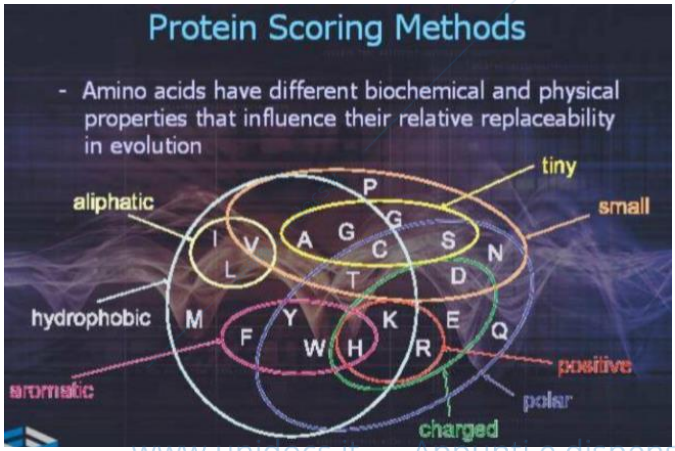
Scoring matrix

	A	C	G	T
A	1	0	0	0
C	0	1	0	0
G	0	0	1	0
T	0	0	0	1

→

	A	C	G	T
A	1	0	.5	0
C	0	1	0	.5
G	.5	0	1	0
T	0	.5	0	1

Applicando regole diverse, è possibile produrre matrici che rispondo a **criteri differenti** *per esempio:*
-Matrici per similarità di codon, che riflettono il numero di eventi mutazionali necessari per trasformare il codon per un aa in quello per un altro.
-Matrici che riflettono la similarità biochimica tra aa; è possibile tener conto anche della similarità tra aa basici, aromatici, o di altri criteri.

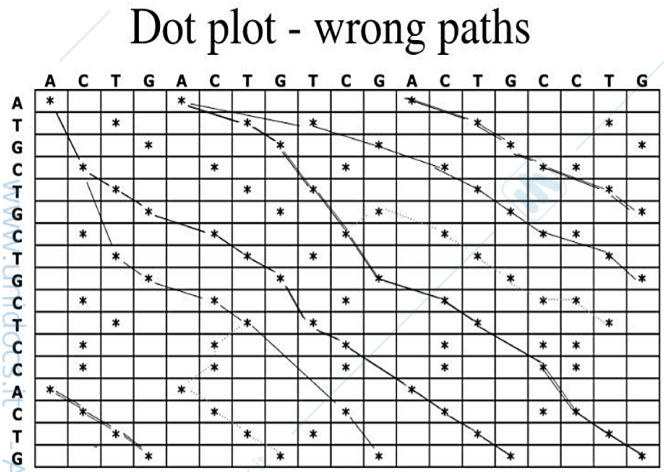


Hydrophobicity matrix

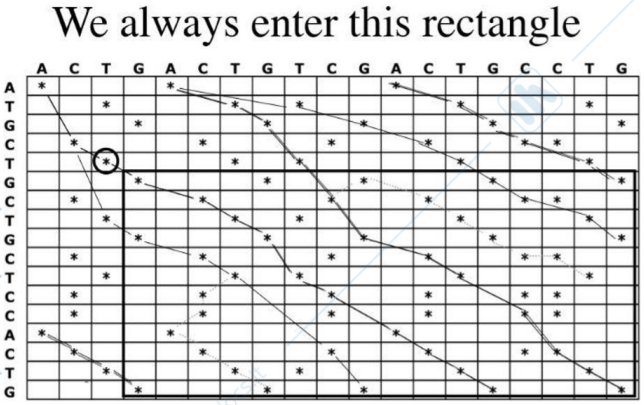
	R	K	D	E	B	Z	S	N	Q	G	X	T	H	A	C	M	P	V	L	I	Y	F	W
Arg = R	10	10	9	9	8	8	6	6	6	5	5	5	5	5	4	3	3	3	3	3	2	1	0
Lys = K	10	10	9	9	8	8	6	6	6	5	5	5	5	5	4	3	3	3	3	3	2	1	0
Asp = D	9	9	10	10	8	8	7	6	6	6	5	5	5	5	5	4	4	4	3	3	3	2	1
Glu = E	9	9	10	10	8	8	7	6	6	6	5	5	5	5	5	4	4	4	3	3	3	2	1
Asx = B	8	8	8	8	10	10	8	8	8	8	7	7	7	7	6	6	6	5	5	5	4	4	3
Glx = Z	8	8	8	8	10	10	8	8	8	8	7	7	7	7	6	6	6	5	5	5	4	4	3
Ser = S	6	6	7	7	8	8	10	10	10	10	9	9	9	9	8	8	7	7	7	7	6	6	4
Asn = N	6	6	6	6	8	8	10	10	10	10	9	9	9	9	8	8	8	7	7	7	6	6	4
Gln = Q	6	6	6	6	8	8	10	10	10	10	9	9	9	9	8	8	8	7	7	7	6	6	4
Gly = G	5	5	6	6	8	8	10	10	10	10	9	9	9	9	8	8	8	8	7	7	6	6	5
??? = X	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	9	8	8	8	7	7	5
Thr = T	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	9	8	8	8	7	7	5
His = H	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	9	8	8	8	7	7	5
Ala = A	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	9	8	8	8	7	7	5
Cys = C	4	4	5	5	6	6	8	8	8	8	9	9	9	9	10	10	9	9	9	9	8	8	5
Met = M	3	3	4	4	6	6	8	8	8	8	9	9	9	9	10	10	10	10	9	9	8	8	7
Pro = P	3	3	4	4	6	6	7	8	8	8	8	8	9	9	9	10	10	10	9	9	9	8	7
Val = V	3	3	4	4	5	5	7	7	7	8	8	8	8	8	9	10	10	10	10	10	9	8	7
Leu = L	3	3	3	3	5	5	7	7	7	7	8	8	8	8	9	9	9	10	10	10	9	9	8
Ile = I	3	3	3	3	5	5	7	7	7	7	8	8	8	8	9	9	9	10	10	10	9	9	8
Tyr = Y	2	2	3	3	4	4	6	6	6	6	7	7	7	7	8	8	9	9	9	9	10	10	8
Phe = F	1	1	2	2	4	4	6	6	6	6	7	7	7	7	8	8	8	8	9	9	10	10	9
Trp = W	0	0	1	1	3	3	4	4	4	5	5	5	5	5	6	7	7	7	8	8	8	9	10

Needleman e Wunsch

Il modo più semplice di trovare l'allineamento ottimale consiste nel calcolare un punteggio per tutti gli allineamenti possibil e scegliere quello che ha il punteggio più alto.
Nella matrice riportata in figura esistono molti allineamenti possibili e non è immediatamente evidente quale di questi sia il migliore. Il primo problema è infatti quello di trovare un modo per calcolare il punteggio; applicando questo metodo, l'allineamento in al a destra (p=6) in figura risulta migliore di quello in basso a sinistra (p=4), ma, se sottraiamo 1 punto per ogni gap (-1), i due punteggi diventano uguali, e l'allineamento di sinistra diventa addirittura meglio se si stabilisce che un gap vale -2. Quale sia il migliore allineamento dipende quindi fortemente dai criteri utilizzati per calcolare il punteggio, ed è quindi molto importante fare attenzione nella scelta di questi criteri.



In maniera indipendente dalla scelta dei criteri utilizzati per il calcolo del punteggio, è però necessario **definire la lista degli allineamenti possibili**. Uno stesso residuo può essere parte di allineamenti alternativi. Alla luce di queste considerazioni, appare chiaro che il numero di allineamenti possibili è molto più alto del numero di diagonali, e che cresce molto velocemente al crescere delle dimensioni della matrice. Non tutti i path però costituiscono allineamenti possibili: gli allineamenti che prevedono il riutilizzo degli stessi residui in posizioni diverse, non vanno considerati.



Quindi, per una qualisiasi casella della matrice, passano molti path; tuttavia, si è limitati a pensare a quelli che proseguono dalla casella cerchiata, ci si rende conto che essi devono tutti proseguire entrando all'interno del triangolo indicato. In pratica, il path può continuare nella casella immediatamente in basso a destra (senza introdurre gap), oppure in una di quelle alla destra di quest'ultima, introducendo così uno o più gap nella seq posta in verticale, oppure ancora in una di quelle sotto di essa introducendo gap nella seq in orizzontale.

Per calcolare il punteggio è utile sostituire gli asterischi con dei valori numeri (ad esempio, inserendo 1 al loro posto e 0 o nulla negli altri).

	A	C	T	G	A	C	T	G	T	C	G	A	C	T	G	C	C	T	G
A	14	13	12	11	12	11	10	10	9	8	7	8	6	5	4	3	2	1	0
T	12	12	13	11	11	10	11	9	10	8	7	7	6	6	4	3	2	2	0
G	12	11	11	12	11	10	9	10	9	8	8	7	6	5	5	3	2	1	1
C	11	12	10	10	10	11	9	9	8	9	7	7	7	5	4	4	3	1	0
T	10	10	11	9	9	9	10	8	9	8	7	7	6	6	4	3	2	2	0
G	10	9	9	10	9	8	8	9	8	7	8	7	6	5	5	3	2	1	1
C	9	10	8	8	8	9	8	8	7	8	6	6	7	5	4	4	3	1	0
T	8	8	9	7	7	7	8	7	8	7	6	6	5	6	4	3	2	2	0
G	8	7	7	8	7	6	6	7	6	6	7	6	5	4	5	3	2	1	1
C	7	8	6	6	6	7	6	6	5	6	5	5	6	4	4	4	3	1	0
T	6	6	7	6	6	5	6	5	6	5	5	5	4	5	4	3	2	2	0
C	5	6	5	5	5	6	5	5	5	5	4	4	5	4	4	4	3	1	0
C	4	5	4	4	4	5	4	4	4	5	4	3	4	3	3	4	3	1	0
A	4	3	3	3	4	3	3	3	3	3	3	4	3	3	3	3	2	1	0
C	2	3	2	2	2	3	2	2	2	3	2	2	3	2	2	3	3	1	0
T	1	1	2	1	1	1	2	1	2	1	1	1	1	2	1	1	1	2	0
G	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1

	A	C	T	G	A	C	T	G	T	C	G	A	C	T	G	C	C	T	G
A	14	13	12	11	12	11	10	10	9	8	7	8	6	5	4	3	2	1	0
T	12	12	13	11	11	10	11	9	10	8	7	7	6	6	4	3	2	2	0
G	12	11	11	12	11	10	9	10	9	8	8	7	6	5	5	3	2	1	1
C	11	12	10	10	10	11	9	9	8	9	7	7	7	5	4	4	3	1	0
T	10	10	11	9	9	9	10	8	9	8	7	7	6	6	4	3	2	2	0
G	10	9	9	10	9	8	8	9	8	7	8	7	6	5	5	3	2	1	1
C	9	10	8	8	8	9	8	8	7	8	6	6	7	5	4	4	3	1	0
T	8	8	9	7	7	7	8	7	8	7	6	6	5	6	4	3	2	2	0
G	8	7	7	8	7	6	6	7	6	6	7	6	5	4	5	3	2	1	1
C	7	8	6	6	6	7	6	6	5	6	5	5	6	4	4	4	3	1	0
T	6	6	7	6	6	5	6	5	6	5	5	5	4	5	4	3	2	2	0
C	5	6	5	5	5	6	5	5	5	5	4	4	5	4	4	4	3	1	0
C	4	5	4	4	4	5	4	4	4	5	4	3	4	3	3	4	3	1	0
A	4	3	3	3	4	3	3	3	3	3	3	4	3	3	3	3	2	1	0
C	2	3	2	2	2	3	2	2	2	3	2	2	3	2	2	3	3	1	0
T	1	1	2	1	1	1	2	1	2	1	1	1	1	2	1	1	1	2	0
G	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1

Nell'**algoritmo di Needlemann e Wunsch,** in ciascuna casella, questo numero viene sostituito da valori che corrispondono al punteggio del miglior path, tra quelli che, passando per quella casella, proseguono fino alla fine della matrice concludendosi sul suo margine destro o inferiore. Ricordando che i path possibili continuano solo all'interno del rettangolo posto in basso e a destra della casella stessa, basta trovare, nel rettangolo, il valore più alto, che si troverà lungo i margini alto e sx, e sommarlo al valore contenuto nella casella in esame. In questo modo questi valori possono essere calcolati per tutte le caselle della matrice, partendo dall'angolo in basso a dx e proseguendo fino ai margini superiore e sx.
Nella matrice utilizzata come esempio, ci sono due percorsi differenti dovuti al fatto che nel punto di diramazione c'è un'ambiguità: ci sono due valori equivalenti che posso considerare (due 8).
Non sono ammessi spostamenti diretti verso destra della stessa riga o verso il basso della stessa colonna della casella considerata.
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. "
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) Io 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 3°seq,
(seq C): S(A,B) ≤ S(A,C) + S(B,C).

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	

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.

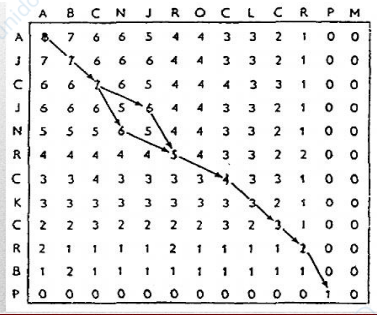
		A	B	C	N	Y	R	Q	C	L	C	R	P	M
	A	1												
	Y					1								
	C			1					1		1			
	Y					1								
	N				1									
	R						1					1		
	C			1					1		1			
	K													
	C			1					1		1			
	R						1					**2**	**0**	**0**
	B	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
	P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

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

NB:
Non sono ammessi spostamenti **diretti** verso destra o verso il basso.



La soluzione non è **unica** ... (+ **percorsi** con lo score massimo)