# ALGORITMO DI NEEDLEMAN-WUNSCH

Riferimenti:

- https://www.kaell.se/bibook/pairwise/needleman.html
- https://bioboot.github.io/bimm143_W20/class-material/nw/

“L'algoritmo di **Needleman-Wunsch** è stato uno dei primi approcci computazionali all'allineamento delle sequenze. La sua introduzione ha segnato un significativo progresso nella bioinformatica, consentendo il confronto sistematico e automatizzato delle sequenze biologiche” (come DNA RNA e proteine).

“L'algoritmo si basa sulla programmazione dinamica, un metodo che suddivide i problemi complessi in sottoproblemi più semplici e più piccoli, risolvendoli una sola volta e memorizzando le loro soluzioni. Nel contesto dell'allineamento di sequenza, costruisce un allineamento globale ottimale confrontando ogni carattere di una sequenza con ogni carattere di un'altra, considerando i costi di corrispondenze (matches), disallineamenti (mismatches) e lacune (gaps).”

Lo scopo è allineare quanto più possibile due sequenze di lunghezza simile, che si presume abbiano qualche correlazione, e calcolare il punteggio di allineamento, ovvero un numero che punta a calcolare la somiglianza tra le due sequenze.

/*Per capire cosa vuol dire allineare il più possibile due sequenze facciamo un esempio:

Seq X: AGGATTACAACT  Seq Y: CATGATTACTAA

Per farle combaciare possiamo aggiungere degli spazi, dei gap

Seq X: _ A G G A T T A C _ A A C T

Seq Y: C A T G A T T A C T A A _ _

In questo modo le nostre sequenze sono più allineate anche se non combaciano perfettamente.

Spoiler: questo è un esempio fatto a mente in base a quello che ho capito, non si se poi l’algoritmo le allinei proprio così */

Il nostro algoritmo dovrà quindi essere in grado di mandare in uscita le sequenze quanto più possibile allineate grazie all’aggiunta di gap oltre che il punteggio di allineamento.

Per calcolare il punteggio di allineamento si sfrutta una matrice 2D. Supponendo di avere una sequenza X e una Y lunghe rispettivamente LX e LY, la matrice avrà LY+1 righe e LX+1colonne, dove il +1 indica una prima cella di gap sia per le righe che per le colonne.

| | (GAP) 0 | G 1 | A 2 | C 3 | T 4 |
|---|---|---|---|---|---|
| (GAP) 0 | 0 | | | | |
| A 1 | | | | | |
| C 2 | | | | | |
| T 3 | | | | | |
| G 4 | | | | | |

In questo esempio abbiamo una sequenza X=GACT e una sequenza Y=ACTG

A questo punto bisogna assegnare i punteggi per match mismatch e gap. Assumiamo ad esempio di poter assegnare i seguenti punteggi:

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

(!! Nel codice che abbiamo vale -1 ma sto usando -2 per non avere un valore uguale al mismatch e magari confondere per i calcoli)

Una volta decisi bisogna iniziare a popolare la tabella. Per fare ciò si inizia dalla prima cella [0,0] e le si assegna il punteggio 0. A partire da questa cella si andrà a popolare tutta la prima riga e poi tutta la prima colonna. Una volta completate si più andare avanti con gli indici e quindi calcolare la cella [1,1] poi tutta la riga 2 e la colonna 2, e così a seguire finché non si è popolata tutta la tabella.

Per capire come calcolare i valori usiamo la tabella di sopra come esempio e la popoliamo.
La prima riga e la prima colonna sono le più facili e ci forniranno i valori di partenza per le altre celle. Siccome siamo della riga di GAP e abbiamo assegnato -2 come punteggio di gap partendo dallo 0 sommiamo -2 per ogni passaggio alla cella successiva. Quindi avremo:

| | (GAP) 0 | G 1 | A 2 | C 3 | T 4 |
|---|---|---|---|---|---|
| (GAP) 0 | 0 | -2 | -4 | -6 | -8 |
| A 1 | -2 | | | | |
| C 2 | -4 | | | | |
| T 3 | -6 | | | | |
| G 4 | -8 | | | | |

Dove ogni freccetta vale -2.

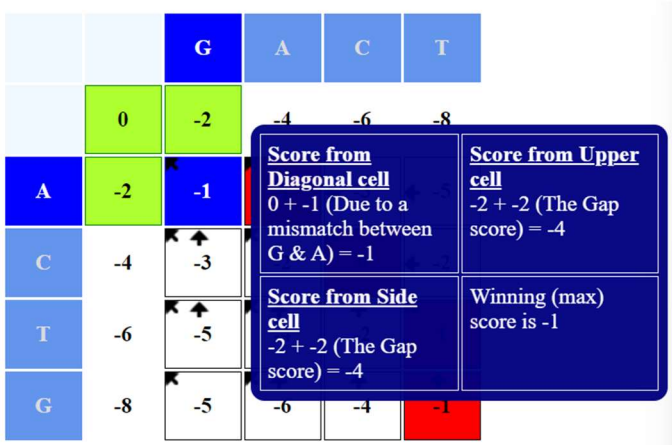
Il punteggio di ogni cella, infatti, è il valore massimo di tre diversi calcoli:

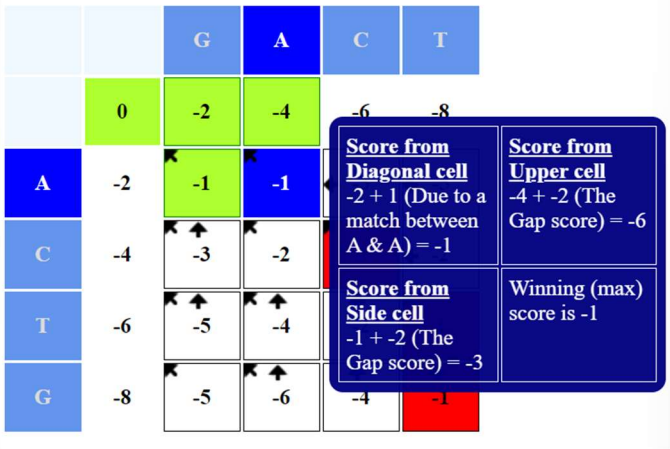
- Valore cella a sx + valore del gap (-2);
- Valore cella sopra + valore del gap (-2);
- Valore cella in alto a sx* + valore di match (+/- 1)

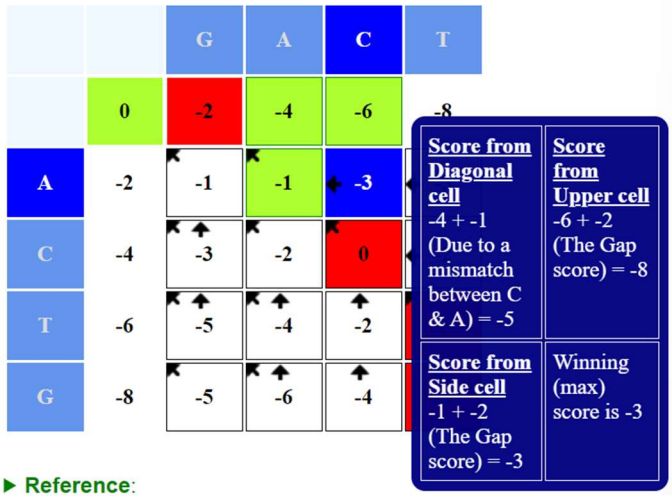
(* la cella sopra in diagonale)

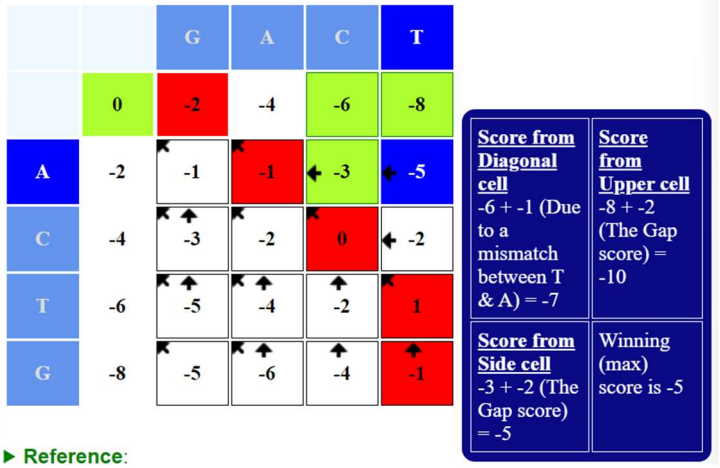
Le celle appena popolate (in verde) della riga non hanno valori sopra o precedenti sulla diagonale; quindi, l'unico valore disponibile è quello delle celle a sx + valore di gap (-2). Ragionamento simile per le celle sulla colonna.

Abbiamo ora ciò che ci serve per calcolare la cella [1,1] e da quella poi tutta la riga 1 e tutta la colonna 1

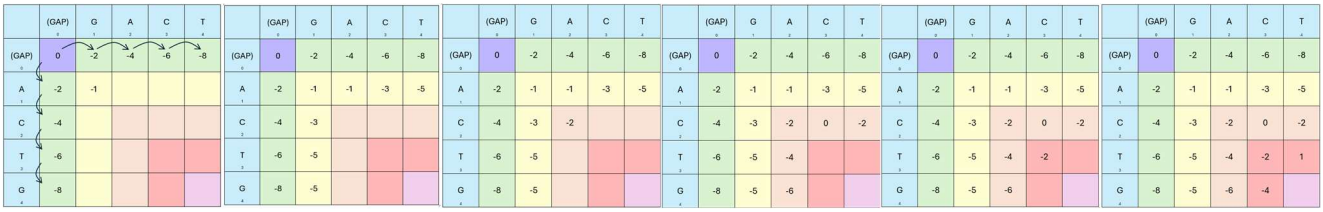



Poi si fa lo stesso ragionamento per la colonna e una volta completate le celle con indice 1 si passa alla cella [2,2] e si ripete lo stesso procedimento. E così per tutte le altre celle, popolando la matrice secondo le fasce di colori e iniziando da quella nell'angolo, sulla diagonale principale. Come mostrato di seguito



Otteniamo così la matrice finale:

| | (GAP) 0 | G 1 | A 2 | C 3 | T 4 |
|---|---|---|---|---|---|
| (GAP) 0 | 0 | ← -2 | ← -4 | ← -6 | ← -8 |
| A 1 | ↑ -2 | ↖ -1 | ↖ -1 | ← -3 | ← -5 |
| C 2 | ↑ -4 | ↖ ↑ -3 | ↖ -2 | ↖ 0 | ← -2 |
| T 3 | ↑ -6 | ↖ ↑ -5 | ↖ ↑ -4 | ↑ -2 | ↖ 1 |
| G 4 | ↑ -8 | ↖ -5 | ↖ ↑ -6 | ↑ -4 | ↑ -1 |

Dove le frecce indicano da che cella precedente si è ottenuto il risultato migliore

Se ci sono più frecce il risultato si poteva ottenere a partire da entrambe.

Una volta ottenuta la matrice completa, bisogna effettuare il percorso inverso, partendo dall'angolo in basso a destra e tornando indietro passando per le celle che sono collegate con le freccette. In pratica noi partiamo dalla cella [4,4] e, per tornare alla cella [0,0], passiamo per quelle celle adiacenti che ci hanno portato al risultato sulla cella in analisi. Lo so che è contorto perché non so come spiegarlo, comunque facciamo l'esempio pratico: la cella [4,4] ha il valore -1, questo valore lo abbiamo ricavato dalla cella che si trova sopra, la [4(sulla x),3(sulla y)] che ha valore 1. Questo valore, che è il migliore ottenuto per questa cella, è quello proveniente dal calcolo a partire dalla cella [3,2] che ha valore 0. E così via il ragionamento a ritroso. Bisogna ricordarsi che, se il punteggio migliore di una cella si può ottenere a partire da due differenti celle precedenti, bisogna passare per la cella con punteggio migliore.

Questo passaggio, detto backtracking, è quello che permette la ricostruzione delle sequenze allineate. Partendo dalla fine andremo quindi a ricostruire e quindi riscrivere entrambe le sequenze, aggiungendo dei gap a una delle due sequenze in base a necessità, o riscrivendo il carattere analizzato, secondo i criteri esplicati di seguito.

Ogni passo che viene fatto indietro può avvenire in 3 direzioni:

- In diagonale: abbiamo quindi un confronto diretto tra i caratteri delle sequenze e, sia che otteniamo un match che un mismatch, aggiungiamo il carattere della sequenza X e il carattere della sequenza Y nelle rispettive sequenze di allineamento
- In verticale: Corrisponde ad un gap nella sequenza orizzontale (X), riscriviamo quindi il carattere della sequenza verticale nella sequenza allineata Y, mentre in quella allineata X metteremo un trattino (-)
- In orizzontale: corrisponde ad un gap nella sequenza verticale (Y), riscriviamo il carattere della sequenza orizzontale e aggiungiamo un trattino per quella verticale.

Facciamo un esempio per capirlo meglio:

Se prendiamo la tabella fatta precedentemente dobbiamo partire dal -1 della cella [4,4] che è stato ottenuto dalla cella sopra con valore 1. Abbiamo quindi uno spostamento verticale dobbiamo quindi riscrivere il carattere della sequenza Y quindi G e in corrispondenza mettere per la sequenza X un trattino (partendo dalla fine):

- Seq allineata X: … -
- Seq allineata Y: … G

Ci troviamo ora nella cella [4,3] che ha valore 1, ottenuto dalla cella diagonale [3,2] che ha valore 0, grazie al match di T e T. Dobbiamo quindi aggiungere entrambi i caratteri alle sequenze (alla sinistra di quelle già inserite). Otteniamo:

- Seq allineata X: … T -
- Seq allineata Y: … T G

Possiamo ora spostarci alla cella [3,2] che ha valore 0, valore ottenuto dalla cella diagonale [2,1], di valore -1, grazie ad un match tra C e C. Riscriviamo entrambi i caratteri come fatto prima:

- Seq allineata X: … C T -
- Seq allineata Y: … C T G

Stesso ragionamento ora per la cella [2,1] il cui valore è stato ricavato in diagonale dalla cella [1,0] di valore -2. Otteniamo quindi:

- Seq allineata X: … A C T -
- Seq allineata Y: … A C T G

Ora dalla cella [1,0] di valore -2 arriviamo alla cella [0,0] con valore 0 grazie ad uno spostamento orizzontale. Scriviamo quindi la lettera corrispondente alla sequenza X e mettiamo un – per la Y:

- Seq allineata X: G A C T -
- Seq allineata Y: - A C T G

Abbiamo quindi allineato le sequenze. E con questo finisce il processo dell'algoritmo in esame.