Allineamento multiplo di sequenze*

Nicola Vitacolonna
vitacolo@dimi.uniud.it
http://www.dimi.uniud.it/~vitacolo
Università degli Studi di Udine

16 aprile 2002

Two homologous sequences whisper...
a full multiple alignment shouts out loud.
A. Lesk

1 Introduzione

L'allineamento di coppie di sequenze cattura le somiglianze fra *due* strutture. Tipicamente, tuttavia, le sequenze sono raggruppabili in *famiglie*: di solito si assume che le differenze all'interno di una famiglia siano conseguenza di mutazioni da un antenato comune avvenute nel corso dell'evoluzione. Sequenze appartenenti ad una famiglia solitamente sono simili dal punto di vista funzionale, anche se possono essere distanti dal punto di vista strutturale.

Sono desiderabili pertanto metodi di analisi che permettano di stabilire la relazione tra una data sequenza e una famiglia di sequenze. Ciò consente spesso di inferire alcune caratteristiche funzionali della sequenza in esame. A tale scopo il confronto a coppie non è adeguato, in quanto non consente di cogliere aspetti globali di una famiglia statisticamente rilevanti. Ad esempio, certe parti delle sequenze possono essere maggiormente conservate rispetto ad altre: è opportuno allora verificare fino a che grado queste parti siano presenti in una nuova sequenza di cui si vuole testare la relazione con la famiglia.

Un'indagine di questo tipo può essere compiuta mediante la ricerca dell'allineamento multiplo ottimo (*multiple sequence alignment problem*). Un esempio di allineamento multiplo è mostrato nella Figura 1. In generale, sono necessari un metodo statistico per valutare e confrontare gli allineamenti e un algoritmo per trovare la migliore valutazione. Nel seguito faremo riferimento soprattutto ad allineamenti di proteine, anche se quanto detto è applicabile a sequenze di DNA.

Diamo una definizione formale del problema. Sia Σ un alfabeto (per fissare le idee, formato dalle lettere che convenzionalmente denotano gli amminoacidi).

*Appunti basati sul contenuto del Capitolo 6 di [DEKM98].

```
1.  VTISCTGSSSNIGAG-NHVKWYQQLPG
2.  VTISCTGTSSNIGS--ITVNWYQQLPG
3.  LRLSCSSSGFIFSS--YAMYWVRQAPG
4.  LSLTCTVSGTSFDD--YYSTWVRQPPG
5.  PEVTCVVVDVSHEDPQVKFNWYVDG--
6.  ATLVCLISDFYPGA--VTVAWKADS--
7.  AALGCLVKDYFPEP--VTVSWNSG---
8.  VSLTCLVKGFYPSD--IAVEWESNG--
```

Figura 1: Un allineamento multiplo in cui sono mostrati insieme otto frammenti da sequenze di immunoglobulina. Il segno meno denota i *gap*. L'allineamento mette in luce i residui maggiormente conservati: esempi notevoli sono una delle cisteine (`C`) che formano ponti disolfuro (legami covalenti tra gli atomi di zolfo presenti nelle catene laterali delle cisteine) e il triptofano (`W`), rispettivamente in quinta e ventunesima posizione. Un altro esempio è la regione `Q.PG` alla fine delle prime quattro sequenze. È possibile inoltre osservare la dominanza di residui idrofobici (leucina `L`, isoleucina `I`, valina `V`, alanina `A`) in prima e terza posizione. L'allineamento permette di fare anche congetture sulla storia evolutiva delle sequenze: le prime quattro e le ultime quattro potrebbero derivare da due diversi antenati, a loro volta discendenti da un'unica sequenza. Ovviamente, frammenti piú lunghi permetterebbero un'analisi maggiormente dettagliata.

Per un intero $k > 2$ siano $x_1, \ldots, x_k$ k stringhe (*sequenze*) su Σ. Un *allineamento multiplo* di $x_1, \ldots, x_k$ è una tabella M di elementi di $\Sigma \cup \{-\}$ con le seguenti caratteristiche:

1. M ha k righe;
2. l'i-esima riga, se si ignorano i gap $-$, coincide con x_i;
3. ogni riga contiene almeno un carattere diverso da $-$.
4. ogni colonna contiene almeno un carattere diverso da $-$.

Denotiamo con ℓ il numero di colonne di M, con m_j la j-esima colonna e con m_{ij} il j-esimo simbolo dell'i-esima riga (che è un elemento di x_i o un gap). Inoltre, data una sequenza x, con $|x|$ si indica il numero di simboli di x e, per qualche $a, b \in \{1, \ldots, |x|\}$, con $x[a..b]$ si indica la sottosequenza di x individuata dalle posizioni a e b. Data una funzione S che associ un valore numerico a ciascun possibile allineamento M di $x_1, \ldots, x_k$, il *problema dell'allineamento multiplo ottimo* consiste nel minimizzare/massimizzare S al variare di M.

2 Un modello per la valutazione di un allineamento multiplo

È evidente che per ottenere un allineamento multiplo biologicamente significativo sia di fondamentale importanza la scelta della funzione S. Idealmente, S dovrebbe codificare tutta l'informazione nota sugli aspetti strutturali e funzionali e sulla storia evolutiva delle sequenze. In pratica, almeno le seguenti due caratteristiche sono prese in considerazione:

- si verifica che, statisticamente, residui in certe posizioni nelle sequenze di una famiglia sono maggiormente conservati rispetto a residui in altre posizioni. La funzione S dovrebbe allora tenere conto della posizione all'interno delle sequenze (*position-specific scoring*);
- le sequenze non sono indipendenti fra loro, ma appartenenti ad un *albero filogenetico* che ne descrive l'evoluzione.

Come ulteriore semplificazione, inizialmente considereremo modelli che ignorano entrambi questi aspetti. Cosí facendo è possibile calcolare un punteggio per ciascuna colonna dell'allineamento in modo indipendente. Il modo standard di farlo si basa sull'addizione di punteggi relativi a coppie di elementi ed è chiamato *costo SP* (*Sum of Pairs*). Data una funzione s che associa un punteggio ad una coppia di simboli (ad esempio, una matrice di sostituzione), il punteggio della j-esima colonna può essere calcolato come

$$S(m_j) = \sum_{i=1}^{k} \sum_{i<l\leq k} s(m_{ij}, m_{lj}) \tag{1}$$

e il punteggio di un allineamento M è[1]

$$S(M) = \sum_{j=1}^{\ell} S(m_j). \tag{2}$$

La funzione s deve essere definita anche quando uno o entrambi i suoi argomenti sono gap. Nel caso in cui la funzione di gap sia $\gamma(g) = -gd$ per qualche d, è sufficiente porre, per ogni $a \in \Sigma$, $s(-, a) = s(a, -) = -d$, e $s(-, -) = 0$. Se al contrario si usa una funzione di gap diversa, si pone $s(-, a) = s(a, -) = 0$ e si gestiscono i gap a parte mediante un termine addizionale nella (2), la cui forma dipenderà dalla particolare funzione utilizzata.

Si noti che, pur avendo il vantaggio della semplicità, questa formulazione non è esente da problemi. Oltre a non avere una giustificazione probabilistica (perché non si tiene conto delle relazioni filogenetiche tra le sequenze) ha anche un difetto che può essere illustrato dal seguente esempio. Con riferimento alla Figura 1, il peso associato alla quinta colonna, (quella contenente la cisteina in tutte le $k = 8$ sequenze), è dato da $s(\texttt{C}, \texttt{C})k(k-1)/2$, dove $k(k-1)/2$ è il numero di coppie distinte di `C` (`C` in sequenze diverse sono considerate diverse) nella colonna. Se si usa la matrice di sostituzione BLOSUM50, si ha $s(\texttt{C}, \texttt{C}) = 13$. Se nell'ultima sequenza vi fosse `G` al posto di `C`, il punteggio della colonna sarebbe decrementato di $(k-1)(13+3)$ essendo $s(\texttt{G}, \texttt{C}) = -3$. Il rapporto tra i due costi sarebbe:

$$\frac{16(k-1)}{13k(k-1)/2} = \frac{32}{13k}.$$

Si noti la dipendenza rispetto a k: al crescere del numero di sequenze la differenza relativa tra un allineamento corretto (tutte `C`) e uno scorretto (una `G` allineata con tutte `C`) *diminuisce*. Intuitivamente, il rapporto dovrebbe crescere dal momento che, all'aumentare di k, si rafforza la tesi della presenza di una cisteina conservata in quella posizione.

[1] Applichiamo la stessa funzione S sia ad un allineamento sia ad una singola colonna dell'allineamento. L'ambiguità è risolta dal contesto.

3 Algoritmo standard di programmazione dinamica

Un algoritmo naïve per la determinazione dell'allineamento multiplo ottimo consiste nella generalizzazione del caso relativo a due sequenze. Se si assume il modello di punteggi della Sezione 2, detto $\alpha(i_1, \ldots, i_k)$ il massimo punteggio di un allineamento delle sottosequenze iniziali $x_1[1..i_1], \ldots, x_k[1..i_k]$, il punteggio dell'allineamento ottimo può essere calcolato ricorsivamente come segue:

$$\alpha(i_1,\ldots,i_k) = \max \begin{cases} \alpha(i_1-1, i_2-1, \ldots, i_k-1) & + & S(m_{1i_1}, m_{2i_2}, \ldots, m_{ki_k}) \\ \alpha(i_1, i_2-1, \ldots, i_k-1) & + & S(-, m_{2i_2}, \ldots, m_{ki_k}) \\ \alpha(i_1-1, i_2, \ldots, i_k-1) & + & S(m_{1i_1}, -, \ldots, m_{ki_k}) \\ & \vdots & \\ \alpha(i_1-1, i_2-1, \ldots, i_k) & + & S(m_{1i_1}, m_{2i_2}, \ldots, -) \\ \alpha(i_1, i_2, i_3-1 \ldots, i_k-1) & + & S(-, -, m_{3i_3}, \ldots, m_{ki_k}) \\ & \vdots & \\ \alpha(i_1, i_2-1, \ldots, i_{k-1}-1, i_k) & + & S(-, m_{2i_2}, \ldots, -) \\ & \vdots & \end{cases}$$

Nella precedente equazione compaiono tutte le 2^k-1 combinazioni di gap eccetto quella contenente soltanto gap. Procedendo in modo analogo al caso di coppie di sequenze, usando tale equazione si costruisce una matrice k-dimensionale con $|x_1||x_2|\cdots|x_k|$ elementi. Ciascun elemento richiede il calcolo di $2^k - 1$ valori. Il tempo totale è dunque $O(2^k \prod_{i=1}^{k} |x_i|) \cdot O(\text{calcolo di } S)$, mentre lo spazio richiesto è quello per la memorizzazione della matrice k-dimensionale, vale a dire $O(\prod_{i=1}^{k} |x_i|)$.

È stato dimostrato ([WJ94]) che il problema dell'allineamento multiplo con costo SP è NP-completo.

4 Algoritmo di Carrillo-Lipman

L'esplorazione di tutto lo spazio della matrice k-dimensionale è decisamente troppo costoso. In questa sezione vedremo una tecnica per ridurre il volume degli elementi da calcolare. Si osservi innanzitutto che la (2) può essere riscritta come segue:

$$S(M) = \sum_{i=1}^{k} \sum_{i<l\leq k} \sum_{j=1}^{\ell} s(m_{ij}, m_{lj}). \tag{3}$$

La (3) può essere interpretata come la somma dei punteggi di tutti gli allineamenti delle coppie di sequenze, definiti a partire dall'allineamento multiplo mediante la restrizione, o *proiezione*, su due sequenze. Dato un allineamento M di $x_1, \ldots, x_k$, denotiamo con $M|_{i,l}$ la proiezione di M sulla coppia di sequenze x_i, x_l. Allora la (3) diventa

$$S(M) = \sum_{i=1}^{k} \sum_{i<l\leq k} S(M|_{i,l}). \tag{4}$$

Il punteggio $S(M|_{i,l})$ può essere calcolato nello stesso modo in cui si calcola il punteggio nell'allineamento di coppie di sequenze.

Sia M^{opt} l'allineamento ottimo di $x_1, \ldots, x_k$. Vogliamo stabilire, per ogni coppia di sequenze x_p, x_q, un limite inferiore al punteggio della proiezione $M^{\text{opt}}|_{p,q}$ su x_p e x_q. Se siamo in grado di fare ciò, è possibile escludere, in virtú dell'interpretazione data alla (3), tutti gli allineamenti multipli tali che almeno una delle proiezioni abbia un punteggio inferiore a tale limite.

Supponiamo di conoscere un allineamento euristico $\hat{M}$ di $x_1, \ldots, x_k$ "prossimo" a quello ottimo. Indichiamo inoltre con $d(x_i, x_l)$ il punteggio dell'allineamento ottimo della *coppia* di sequenze x_i, x_l. Per definizione, si ha che $S(M^{\text{opt}}) \geq S(\hat{M})$. Usando la (4), ciò equivale a

$$\sum_{i=1}^{k} \sum_{i<l\leq k} \left(S(M^{\text{opt}}|_{i,l}) - S(\hat{M}|_{i,l})\right) \geq 0. \tag{5}$$

Si noti che, in generale, nulla si può dire sui punteggi delle proiezioni $M^{\text{opt}}|_{i,l}$ e $\hat{M}|_{i,l}$, se non che entrambi devono essere al piú pari a $d(x_i, x_l)$. Fissate due particolari sequenze x_p e x_q, la (5) può essere riscritta nella seguente forma:

$$\Big(\sum_{\substack{i=1\\ i\neq p}}^{k} \sum_{\substack{i<l\leq k\\ l\neq q}} \left(S(M^{\text{opt}}|_{i,l}) - S(\hat{M}|_{i,l})\right)\Big) + S(M^{\text{opt}}|_{p,q}) - S(\hat{M}|_{p,q}) \geq 0, \tag{6}$$

che riordinando i termini diventa

$$S(\hat{M}|_{p,q}) - \Big(\sum_{\substack{i=1\\ i\neq p}}^{k} \sum_{\substack{i<l\leq k\\ l\neq q}} \left(S(M^{\text{opt}}|_{i,l}) - S(\hat{M}|_{i,l})\right)\Big) \leq S(M^{\text{opt}}|_{p,q}). \tag{7}$$

Poiché $S(M^{\text{opt}}|_{i,l}) \leq d(x_i, x_l)$, sostituendo nella sommatoria si ottiene

$$S(\hat{M}|_{p,q}) - \Big(\sum_{\substack{i=1\\ i\neq p}}^{k} \sum_{\substack{i<l\leq k\\ l\neq q}} \left(d(x_i, x_l) - S(\hat{M}|_{i,l})\right)\Big) \leq S(M^{\text{opt}}|_{p,q}). \tag{8}$$

Il termine sinistro della disuguaglianza, detto *limite di Carrillo-Lipman*, rappresenta il limite cercato al punteggio della proiezione $M^{\text{opt}}|_{p,q}$ su x_p, x_q.

Come si vede, per calcolare i limiti inferiori è necessario conoscere soltanto un allineamento euristico (come questo venga ottenuto è argomento delle prossime sezioni) e i $k(k-1)/2$ allineamenti ottimi di coppie $d(x_i, x_l)$: questi ultimi si calcolano con algoritmi standard per l'allineamento di coppie di sequenze. Una volta fatto ciò, per ogni x_p, x_q si individua un insieme di coppie di indici (i_p, i_q) tali che il punteggio del miglior allineamento di x_p e x_q che passi attraverso (i_p, i_q) sia almeno pari al limite inferiore. Questo si può fare in tempo quadratico usando i punteggi di Viterbi in avanti e all'indietro per ogni elemento della matrice generata dall'algoritmo per l'allineamento a coppie. Infine, l'algoritmo della Sezione 3 è applicato soltanto ai valori ottenuti mediante l'intersezione di tutti gli insiemi precedentemente calcolati. Omettiamo i dettagli con cui quest'ultimo passo viene portato a termine. L'algoritmo di Carrillo-Lipman è descritto in [CL88] ed è stato implementato nel programma MSA.

5 Metodi di allineamento progressivo

Il metodo piú comune di eseguire un allineamento multiplo è il cosiddetto *allineamento progressivo* (*progressive alignment*), basato sulla costruzione di una successione di allineamenti a coppie. Si scelgono due sequenze e si allineano: questo allineamento rimane fissato nei passi successivi. Dopodiché si sceglie una terza sequenza e si allinea al precedente allineamento, e cosí via. Questo approccio è euristico e non garantisce di trovare l'allineamento ottimo: per contro, è efficiente e spesso dà risultati ragionevoli.

L'euristica piú importante utilizzata negli algoritmi di allineamento progressivo prevede che le coppie di sequenze con maggiore grado di somiglianza o, equivalentemente, la cui "distanza genetica" sia minore, siano allineate per prime. Questo modo di procedere è giustificato dal fatto che coppie di sequenze maggiormente somiglianti hanno maggiore probabilità di essere derivate piú recentemente da un antenato comune, e quindi il loro allineamento fornisce l'informazione piú "affidabile" che è possibile ricavare dalle sequenze. In particolare, le posizioni dei gap in sequenze maggiormente correlate sono tipicamente piú accurate rispetto a quelle relative a sequenze meno simili. Ciò porta a formulare la regola euristica per cui i gap degli allineamenti iniziali vadano preservati quando si allineano nuove sequenze (*once a gap, always a gap*).

Molti algoritmi di questo tipo utilizzano i cosiddetti *alberi guida* (*guide tree*), alberi binari le cui foglie sono etichettate con sequenze e i cui nodi interni rappresentano gruppi (*cluster*) di sequenze. Gli alberi guida sono simili agli alberi filogenetici, ma poiché il loro scopo è solo quello di determinare l'ordine in cui effettuare un allineamento progressivo, la loro costruzione è meno accurata e l'informazione che forniscono è meno precisa rispetto ad un vero e proprio albero filogenetico.

5.1 Algoritmo di Feng-Doolittle

L'algoritmo di Feng-Doolittle ([FD87]) è uno dei primi algoritmi per l'allineamento progressivo, in cui trovano applicazione le idee appena descritte. Delineamo i passi dell'algoritmo:

1. calcola i $k(k-1)/2$ allineamenti a coppie delle k sequenze e converti i punteggi in *distanze*;
2. usa un algoritmo di *clustering* incrementale, come quello di Fitch e Margoliash[2] ([FM67]), per costruire un albero guida a partire dalle distanze calcolate al passo precedente;
3. visita i nodi nell'ordine in cui sono stati aggiunti all'albero e allinea i figli (che possono essere sequenze o *cluster*) finché non sono state allineate tutte le sequenze. Una sequenza è aggiunta ad un esistente gruppo di sequenze allineandola con ciascuna di esse e scegliendo l'allineamento col

[2] L'obiettivo del metodo di Fitch-Margoliash è di trovare l'albero che minimizza la quantità $\sum_{i,j} \frac{(D_{ij}-d_{ij})^2}{D_{ij}^2}$, dove D_{ij} è la distanza tra l'i-esima e la j-esima sequenza e d_{ij} è una distanza media calcolata come la somma delle lunghezze dei rami nel cammino da i a j.

	1	2	3	4	5	6	7
A	.25	.125					
R		.125					
C					1		
E		.125					
G				.125			.25
H							
I			.25				.125
L	.25		.625			.375	
P	.125						
S		.25		.375		.125	.125
T		.375		.375		.375	
V	.375		.125	.125		.125	.5

Figura 2: Il profilo corrispondente alle prime sette posizioni dell'allineamento di Figura 1. Le righe omesse e le posizioni senza numeri contengono zero.

punteggio massimo. Due gruppi di sequenze sono fusi insieme allineandoli sulla base dell'allineamento della coppia con il punteggio migliore.

L'algoritmo di Feng-Doolittle è implementato ad esempio nei programmi FITCH e KITSCH e nel programma PileUp (che usa UPGMA come algoritmo di clustering).

5.2 Algoritmo di Thompson-Higgins-Gibson

Un problema legato all'algoritmo di Feng-Doolittle è che tutti gli allineamenti sono determinati sulla base di confronti a coppie. Durante un allineamento progressivo è però vantaggioso usare l'informazione, dipendente dalla posizione, che si acquisisce quando si è allineato un gruppo di sequenze. Quest'informazione include il grado in cui certe parti delle sequenze sono conservate: le incompatibilità in posizioni altamente conservate possono essere penalizzate in modo diverso rispetto a differenze in posizioni ad alta variabilità. Ad esempio, nella Figura 1, una `G` nella quinta colonna, in cui vi è un'alta probabilità di una cisteina conservata, dovrebbe fornire un punteggio inferiore rispetto ad una `G`, diciamo, in quarta posizione. Anche i gap possono essere pesati in modo differente, proporzionalmente alla loro frequenza in una certa posizione. In questa sezione vedremo un algoritmo che usa uno schema di valutazione dipendente dalla posizione, chiamato anche *profilo*.

Dato un allineamento multiplo M con ℓ colonne, un *profilo* per M è una tabella $|\Sigma \cup \{-\}| \times \ell$ in cui ciascuna colonna contiene numeri che indicano la frequenza con cui ciascun simbolo compare in quella posizione. La Figura 2 mostra un esempio di profilo.

Allineare una sequenza ad un profilo significa allinearla alle colonne del profilo, permettendo eventualmente gap. La Figura 3 mostra un possibile allineamento della stringa `VTICL` al profilo di Figura 2.

Il punteggio associato all'allineamento di un carattere ad una colonna del

profilo si può calcolare come la somma, pesata con i valori del profilo, di tutti i possibili accoppiamenti di simboli. Con riferimento alla Figura 3, assumendo di usare la matrice BLOSUM50 per valutare gli accoppiamenti, il punteggio della `V` in prima posizione è dato da $.25 \cdot 0 + .25 \cdot 1 - .125 \cdot 3 + .375 \cdot 5 = 1.75$. Il punteggio dell'intero allineamento è la somma dei valori per le singole colonne. Formalmente, siano $P = (p_{ij})$ un profilo, a_i il simbolo che etichetta l'i-esima riga del profilo, e $\alpha \in \Sigma \cup \{-\}$ un carattere. Il punteggio dell'allineamento di α alla j-esima colonna di P è allora

$$v(\alpha, j) = \sum_{i=1}^{|\Sigma \cup \{-\}|} p_{ij} s(a_i, \alpha). \tag{9}$$

Il punteggio dell'allineamento di una sequenza (con gap) $x = \alpha_1 \cdots \alpha_\ell$ ad un profilo P con ℓ colonne è

$$S(x, P) = \sum_{j=1}^{\ell} v(\alpha_j, j). \tag{10}$$

L'allineamento ottimo di una sequenza x ad un profilo P si calcola usando un algoritmo di programmazione dinamica simile a quelli per gli allineamenti a coppie. Sia $F(i, j)$ il punteggio del miglior allineamento di $x[1..i]$ alle prime j colonne di P. Le equazioni per il calcolo dell'allineamento ottimo sequenza-profilo sono le seguenti:

$$F(0, j) = \sum_{k=1}^{j} v(-, k), \tag{11}$$

$$F(i, 0) = \sum_{k=1}^{i} s(x[k..k], -), \tag{12}$$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + v(x[i..i], j) \\ F(i-1, j) + s(x[i..i], -) \\ F(i, j-1) + v(-, j) \end{cases} \quad \text{per } i, j > 0. \tag{13}$$

È possibile anche *allineare un profilo ad un profilo*, estendendo in modo opportuno il calcolo dei punteggi e le equazioni ricorsive per la determinazione dell'ottimo. Nel caso di due profili, eventuali gap vanno inseriti nell'intera colonna di un profilo, al fine di preservare l'allineamento multiplo già esistente all'interno di ciascuno dei due gruppi di sequenze.

Una nota implementazione dell'allineamento progressivo mediante profili è il programma CLUSTAL W ([THG94]). I passi dell'algoritmo, simile a quello di Feng-Doolittle, sono a grandi linee i seguenti:

1	2	3	4	5	6	7
V	T	I	–	C	L	–

Figura 3: Un possibile allineamento della sequenza `VTICL` al profilo di Figura 2.

1. calcola i $k(k-1)/2$ allineamenti a coppie delle k sequenze e converti i punteggi in *distanze*;

2. usa un algoritmo di *neighbour-joining clustering* per costruire un albero guida a partire dalle distanze calcolate al passo precedente;

3. allinea progressivamente i nodi in ordine decrescente di similarità usando allineamenti sequenza-sequenza, sequenza-profilo, profilo-profilo.

CLUSTAL W fa uso inoltre di molte regole ad hoc. Ad esempio, le sequenze di una famiglia hanno associato un *peso* (che serve per compensare un eventuale sbilanciamento nella distribuzione statistica delle sequenze); si usano matrici di sostituzione diverse a seconda del grado di similarità fra le sequenze da confrontare; i punteggi di gap variano in relazione alla frequenza dei residui allineati con i gap.

5.3 Metodi di raffinamento iterativo: algoritmo di Barton-Sternberg

I metodi di raffinamento iterativo hanno lo scopo di superare la limitazione degli algoritmi precedenti dovuta al fatto che, una volta che un gruppo di sequenze è stato allineato, tale configurazione non è piú modificabile ai passi successivi (tranne per il fatto che possono essere inseriti gap in intere colonne). In un metodo iterativo, una volta generato un allineamento iniziale (impiegando ad esempio uno dei metodi precedentemente descritti), una sequenza o un insieme di sequenze è rimosso dall'allineamento e riallineato al profilo relativo alle rimanenti sequenze. Il procedimento è iterato finché non si riscontra piú alcun miglioramento nel punteggio. Si può dimostrare che questa procedura, se si usano tutte le sequenze, converge ad un massimo *locale*.

L'algoritmo seguente ([BS87]) è un esempio di combinazione dei metodi fin qui proposti:

1. date k sequenze, calcola i punteggi di tutti gli allineamenti a coppie. Allinea le sequenze x_1 e x_2 con il massimo grado di somiglianza.

2. per ogni $i = 3, \ldots, k$ sia x_i la sequenza che ha il maggior punteggio nell'allineamento sequenza-profilo rispetto al profilo di $x_1, \ldots, x_{i-1}$, e allineala con tale profilo;

3. per ogni $i = 1, \ldots, k-1$ rimuovi x_i dall'allineamento e riallinea x_i al profilo delle restanti sequenze.

4. ripeti il passo precedente finché il punteggio non converge oppure fino a quando si raggiunge un numero massimo di iterazioni.

Le idee di allineamento mediante profili e di raffinamento iterativo sono strettamente legate ad un altro tipo di approccio al problema dell'allineamento multiplo, un approccio probabilistico basato sui modelli di Markov nascosti.

Riferimenti bibliografici

[BS87] G. J. Barton e M. J. E. Sternberg. A strategy for the rapid multiple alignment of protein sequences. *Journal of Molecular Biology*, 198:327–337, 1987.

[CL88] H. Carrillo e D. Lipman. The multiple sequence alignment problem in biology. *SIAM Journal of Applied Mathematics*, 48:1073–1082, 1988.

[DEKM98] R. Durbin, S. Eddy, A. Krogh e G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[FD87] D. F. Feng e R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.

[FM67] W. M. Fitch e E. Margoliash. Construction of phylogenetic trees. *Science*, 155:279–284, 1967.

[THG94] J. D. Thompson, D. G. Higgins e T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acid Research*, 22:4673–4680, 1994.

[WJ94] L. Wang e T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.