Allineamento multiplo di sequenze

Bioinformatica a.a. 2008/2009

Letterio Galletta

Università di Pisa

22 Maggio 2009

Sommario

Allineamento multiplo

Naturale generalizzazione dell'allineamento tra due sequenze:
Date $s_1, \ldots, s_n$ su Σ, un allineamento $s'_1, \ldots, s'_n$ è ottenuto inserendo degli spazi in modo tale

- $| s'_1 | = \ldots = | s'_n |$
- nessuna colonna di soli spazi

Esempio

```
M Q P I L L L
M L R - L L -
M K - I L L L
M P P V L I L
```

Perché è utile?

Moltissimi campi di applicazione in bioinformatica

- ricostruzione filogenetica
- caratterizzazione delle proteine con funzioni sconosciute

Complessità computazionale

Il problema appartiene alla classe NP-Hard

- esistono algoritmi esatti (programmazione dinamica, ottimizzazione combinatoria), ma sono impraticabili
- si adottano metodi approssimati (algoritmi probabilistici, greedy)

Data l'importanza del problema si continua a cercare metodi sempre più veloci e precisi!

Allineamento progressivo

Cos'è?

- Il metodo euristico maggiormente utilizzato
- Consiste nella costruzione progressiva di allineamenti di coppie di sequenze

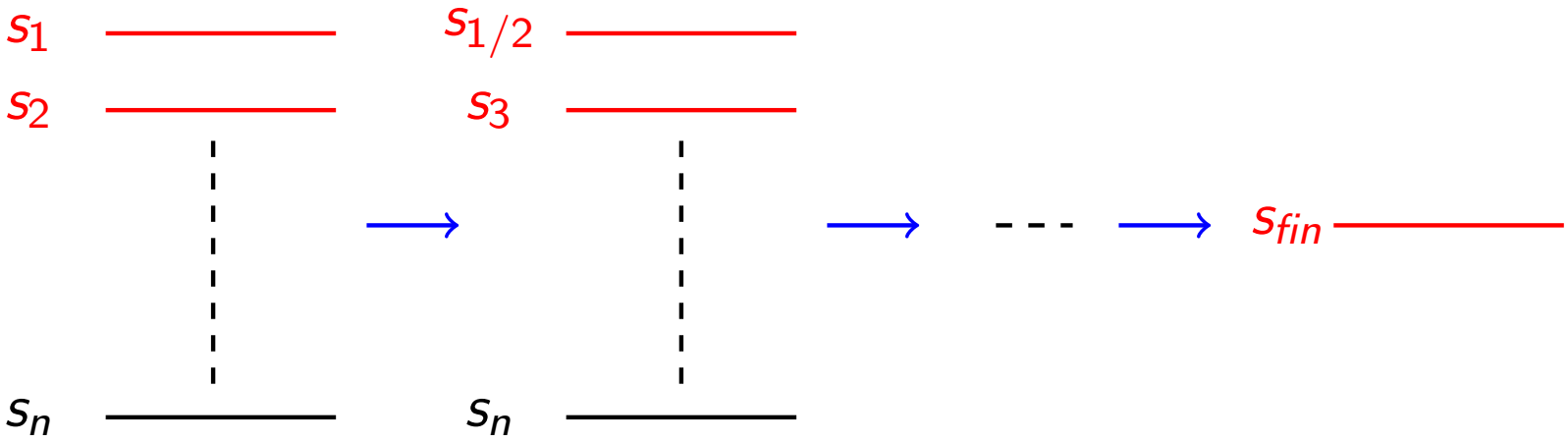
Schema generale

Dato un insieme di sequenze $S = \{s_1, \ldots, s_n\}$

1. Scegli due sequenze s_i e s_j
2. Allinea s_i e s_j ottenendo s_{ij}
3. Inserisci s_{ij} in S
4. Continua fino a quando non ci sono più sequenze da allineare

Allineamento progressivo

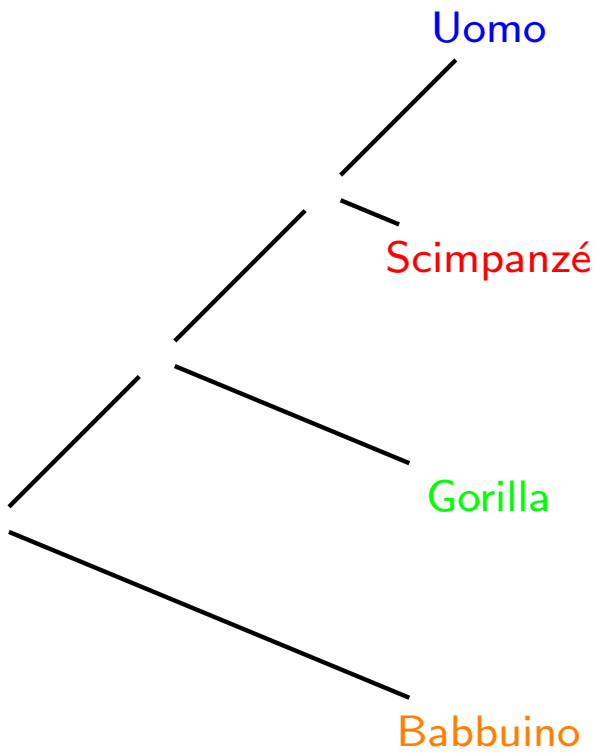
Esempio



- in quale ordine allineare le sequenze?
- è possibile allineare prima le sequenze più simili?

Albero filogenetico

- Per guidare le scelte delle sequenze è possibile fare uso di un albero filogenetico
- Un albero filogenetico mostra le relazioni di discendenza comune di gruppi di organismi



Allineamento progressivo

Conclusioni

L'allineamento progressivo è un metodo greedy

- non si ottimizza nessuna funzione di scoring globale
- non c'è nessuna garanzia sulla bontà del risultato

Esempio

T	*H*	*E*		*L*	*A*	*S*	*T*		*F*	*A*	–	*T*		*C*	*A*	*T*
T	*H*	*E*		*F*	*A*	*S*	*T*		*C*	*A*	–	*T*		–	–	–
T	*H*	*E*		*V*	*E*	*R*	*Y*		*F*	*A*	*S*	*T*		*C*	*A*	*T*
T	*H*	*E*		–	–	–	–		*F*	*A*	–	*T*		*C*	*A*	*T*

In pratica risulta

- abbastanza veloce
- ragionevolmente corretto se si fa uso degli alberi filogenetici

Sommario

Divide Progressive MSA [Lakshmi et al, 2008]

L'idea è di dividere il problema iniziale in sottoproblemi più facili da risolvere e combinare le soluzioni

Descrizione

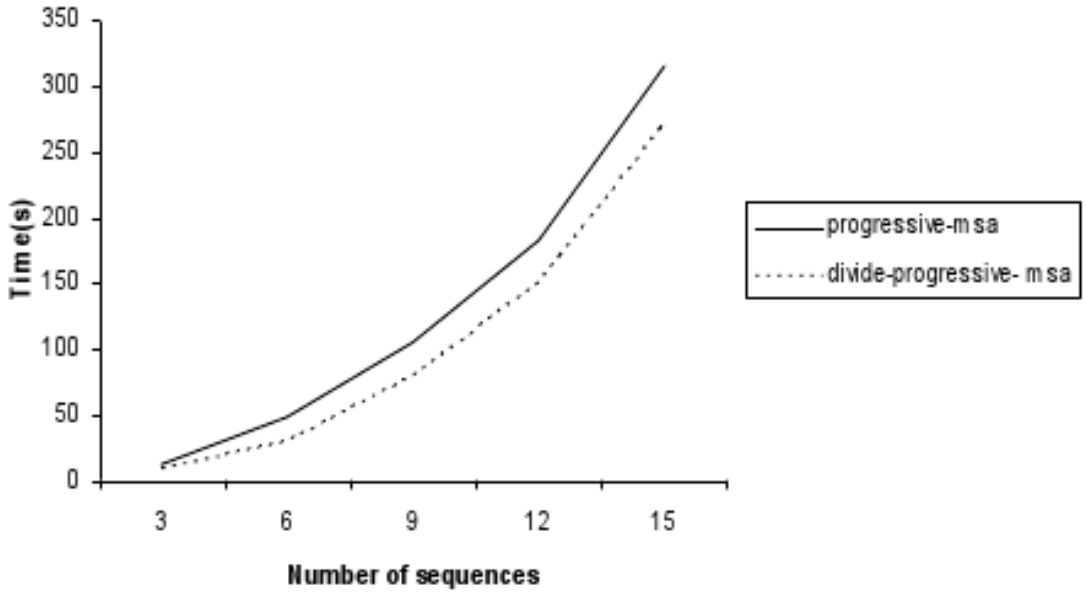
Sono previste tre fasi:

1. Dato $S = \{s_1, \ldots, s_n\}$ si divide ogni sequenza s_i nella posizione c_i ottenendo $S^P = \{s_1^P, \ldots, s_n^P\}$ e $S^S = \{s_1^S, \ldots, s_n^S\}$
   - il partizionamento può essere applicato ricorsivamente fino ad ottenere $S^X = \{s_1^X, \ldots, s_n^X\}$ con $\mid s_i^X \mid \leq L$
2. Eseguire un algoritmo di allineamento progressivo su ogni S^X
3. Combinare gli allineamenti ottenuti

Divide Progressive MSA [Lakshmi et al, 2008]

Conclusioni

Risultati sperimentali mostrano che si ottengono miglioramenti sia per quanto riguarda la precisione sia l'efficienza



Sommario

Segment-based multiple sequence alignment [Rausch et al, 2008]

Algoritmo che usa una strategia progressiva

L'idea

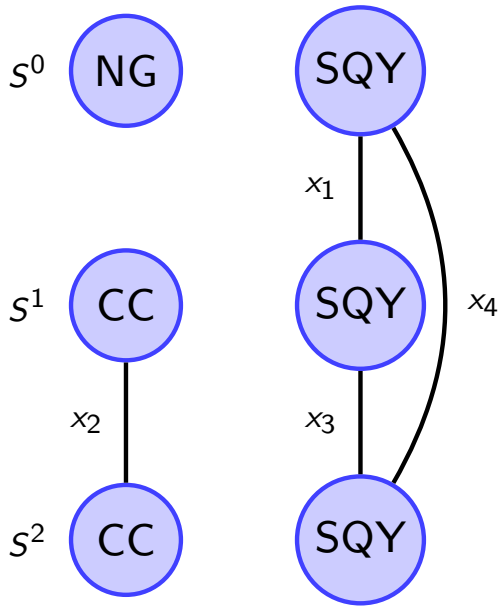
allineare segmenti (sotto stringhe) invece che singoli caratteri

Caratteristiche

- gli allineamenti sono grafi i cui vertici sono formati da segmenti
- prevede una fase che si occupa della consistenza

Grafi di allineamento

Un allineamento multiplo può essere rappresentato come un grafo pesato n-ripartito $G = (V, E)$



- V sono i segmenti
- E sono i match/mismatch
- il peso indica il beneficio ottenuto allineando i due segmenti

Grafi di allineamento

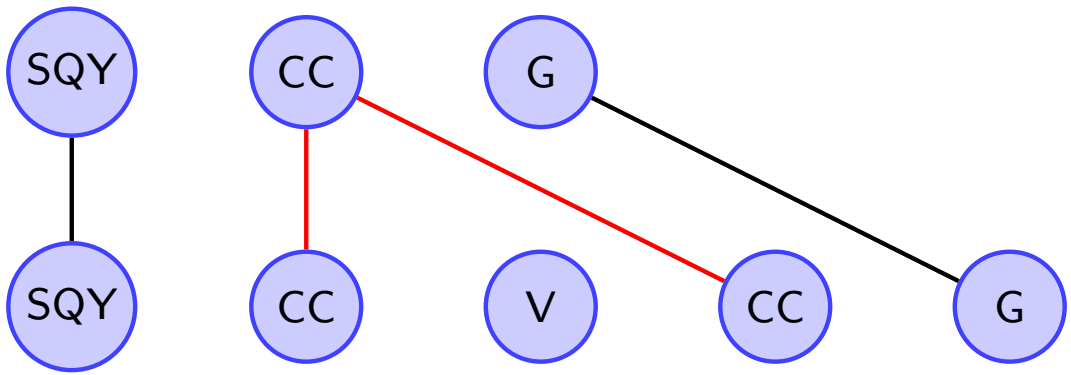
Osservazioni

- i gap sono ottenuti dalla topologia, sebbene il loro ordine non sia unico

Esempio

N	G	$-$	$-$	S	Q	Y	$-$	$-$	N	G	S	Q	Y
$-$	$-$	C	C	S	Q	Y	C	C	$-$	$-$	S	Q	Y
$-$	$-$	C	C	S	Q	Y	C	C	$-$	$-$	S	Q	Y

- non tutti gli i grafi formano un allineamento valido (traccia)



Allineare i grafi

Qual'è l'allineamento ottimo?

La traccia dove la somma di tutti i pesi meno il costo di tutti i possibili gap è massima

Come trovarlo?

- i grafi possono essere considerati sequenze di vertici invece che di caratteri
- dato un albero guida la procedura di allineamento opera come se i vertici fossero singoli caratteri

Affinché funzioni è necessario che i segmenti siano tutti distinti e non abbiano sovrapposizioni

Procedura di Segment-match refinement

Obiettivo

- calcola una suddivisione dei segmenti in modo tale che nessun segmento si sovrapponga parzialmente
- una soluzione banale è la suddivisione in singoli caratteri
  - non va bene perché si vuole la suddivisione che genera un grafo con il minor numero di nodi

Consistenza

In un allineamento progressivo la bontà del risultato finale è influenzata

- l'ordine con le quali si allineano le sequenze
- gli errori fatti nelle prime fasi

Esempio

T	H	E		L	A	S	T		F	A	–	T		C	A	T
T	H	E		F	A	S	T		C	A	–	T		–	–	–
T	H	E		V	E	R	Y		F	A	S	T		C	A	T
T	H	E		–	–	–	–		F	A	–	T		C	A	T

Soluzione

Non basarsi solo su scelte locali ma stimare se l'allineamento tra due sequenze è consistente con il quello finale

- il peso dato ad ogni coppia di residui tiene conto anche delle informazioni derivate dall'analisi delle altre sequenze

Consistenza

Il concetto di consistenza

- è stato introdotto in [Gotoh, 1990]
- è stato implementato per la prima volta nel tool T-Coffee [Notredame et al, 2000]

L'idea

Dato un allineamento $A \leftrightarrows B$ si considerano gli allineamenti $A \leftrightarrows C$ e $C \leftrightarrows B$

- $A[X]$ il residuo X di A
- $W(A[X], B[X])$ peso dei residui
- $W(A[X], B[X]) += min\{ W(A[X], C[X]), W(C[X], B[X])\}$

Consistenza

L'algoritmo di consistenza è derivato da quello di T-Coffee ma opera su un grafo

Pseudocodice

Input: un grafo di allineamento $G = (V, E)$

foreach $v \in V$ **do**
    foreach x, y *adiacenti a* v **do**
        if *esiste* $z = (x, y) \in E$ **then**
            $peso(z) += min\{peso(v, x), peso(v, y)\}$;
        end
        else
            crea $z = (x, y)$;
            $peso(z) = Peso_{min}$;
        end
    end
end

Segment-based multiple sequence alignment [Rausch et al, 2008]

Conclusioni

L'algoritmo visto

- effettua l'allineamento su segmenti invece che caratteri
- utilizza una procedura per garantire la consistenza del risultato
- è usato in un tool (`http://www.seqan.de/projects/msa.html`)
- funziona abbastanza bene in pratica

Conclusioni

Cosa abbiamo visto in questo seminario?

- allineamento multiplo è un problema fondamentale in bioinformatica ma difficile (NP-Hard)
- nella pratica si utilizzano metodi approssimati (allineamento progressivo)
- si cerca di trovare sempre algoritmi più efficienti e precisi
  - Divide Progressive MSA
  - Segment-based multiple sequence alignment

Riferimenti


T. Rausch, A. Emde, D. Weese, A. Döring, C. Notredame, K. Reinert.
Segment-based multiple sequence alignment.
Bioinformatics 24(16): i187–192.


C. Notredame, D. G. Higgins, J. Heringa.
T-Coffee: a novel method for fast and accurate multiple sequence alignment.
J. Mol. Bio. 302: 205-217.


P. V. Lakshmi, Allam Appa Rao, GR Sridhar.
An efficient progressive alignment algorithm for multiple sequence alignment.
IJCSNS 8(10): 301-305.


O. Gotoh.
Consistency of optimal sequence alignments.
BMB 52.