Curs 8.
Alinierea secvențelor:
aliniere multiplă

Biblio:

cap 2. din “Biological sequence analysis”, Durbin et al

cap. 6 și 9 din “An introduction to Bioinformatics algorithms”, N.Jones, P. Pevzner

cap 6 din “Algorithms in Bioinformatics. A practical introduction”, W.K. Sung

Alinierea secventelor - reminder

- Scopul alinierii: obținerea de informații privind similaritatea dintre secvențe
- Elemente cheie ale procesului de aliniere:
  - Stabilirea tipului de aliniere (globală, locală, multiplă)
  - Stabilirea matricilor de scor (pt nucleotide/aminoacizi) care vor fi utilizate pentru a evalua calitatea alinierii
  - Alegerea algoritmului de construire a alinierii (algoritm exact vs. algoritm aproximativ sau euristic)
  - Stabilirea metodelor statistice utilizate pentru a evalua calitatea alinierii

Alinierea multipla

- Alinierea multiplă are ca scop identificarea de similarități între mai multe secvențe ADN sau de aminoacizi (proteine)

- Similaritatea identificată este cu atât mai semnificativă cu cât este adevărată pentru mai multe secvențe: aceasta sugerează prezența unor regiuni conservate în cadrul mai multor ramuri evolutive

- Identificarea similarităților multiple este utilă în proiectarea experimentelor pentru testarea și modificarea funcțiilor unor proteine specifice, în predicția funcției și structurii proteinelor și în identificarea de noi membri în familiile de proteine

Alinierea multipla

- Alinierea a două secvențe: matrice cu două linii și L coloane (L=nr de elemente ale alinierii)
- Alinierea a K secvențe: matrice cu K linii și L coloane

Exemplu: K=3

(n1=lg. secvenței 1, n2=lg. secvenței 2, n3=lg. secvenței 3)

L>=max{n1,n2,n3}

x = ATGC, y = AATC, z = ATGC

A–TGC

AAT– C

–ATGC

Alinierea multipla – interpretare geometrica

x: 0 1 1 2 3 4 indicele elementului curent

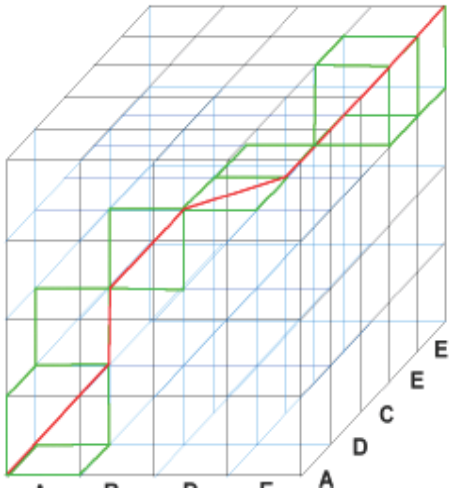
A –T G C

y: 0 1 2 3 3 4

A A T–C

z: 0 0 1 2 3 4

– A T GC



Alinierea este echivalenta cu urmatorul traseu:

(0,0,0)->(1,1,0)->(1,2,1)->(2,3,2)->(3,3,3)->(4,4,4)

ce poate fi interpretat ca o soluție a problemei turistului în cazul 3-dimensional

Alinierea multipla – programare dinamica

Extindere directă a relației de recurență:

$$S_{i,j,k} = \max \begin{cases} s_{i-1,j-1,k-1} + scor(x_i, y_j, z_k) \\ s_{i-1,j-1,k} + scor(x_i, y_j, \_) \\ s_{i-1,j,k-1} + scor(x_i, \_, z_k) \\ s_{i,j-1,k-1} + scor(\_, y_j, z_k) \\ s_{i-1,j,k} + scor(x_i, \_, \_) \\ s_{i,j-1,k} + scor(\_, y_j, \_) \\ s_{i,j,k-1} + scor(\_, \_, z_k) \end{cases}$$

Matricea de scor este tridimensională

Complexitate:

7*n1*n2*n3

Caz general:

$O(2^K n^K)$

(K=nr. secvențe;

n=lungimea maximă a secvențelor)

Alinierea multiplă – programare dinamică

Observație:

- in calculul scorului unei alinieri multiple este necesară utilizarea unui scor de substituție care să exprime șansa ca elementele aliniate să fi provenit din același element inițial (prin mutație – înlocuire, eliminare, inserție)
- O variantă simplă de calcul a scorului de substituție este bazată pe utilizarea scorului corespunzător tuturor perechilor de elemente care apar pe aceeași poziție în cadrul alinierii
- In cazul a K secvențe $a_1, a_2, ..., a_K$ scorul corespunzător poziției j din aliniere este:

$$scor(a_{1j}, a_{2j}, ..., a_{Kj}) = \sum_{i1<i2} scorP(a_{i1,j}, a_{i2,j})$$

unde scorP este scorul de substituție corespunzător unei matrici de substituție (ex: BLOSUM) daca ambele simboluri sunt non-gap sau costul inserării unui gap daca cel puțin unul este gap.

Obs: în calculul sumei de mai sus este posibil ca anumite perechi să conțină două gap-uri. In acest caz scorul scorP se consideră egal cu 0.

Alinierea multiplă– programare dinamică

Generalizare pentru cazul a k secvențe: $x_1, x_2, \ldots x_k$

Idee: extinderea relației de recurența de la cazul particular (k=2) la cazul general prin introducerea unei notații ajutătoare

Cazul k=2

$$S(i_1,i_2)=\max\{S(i_1-1,i_2-1)+scor(x_1[i_1], x_2[i_2]),$$
$$S(i_1-1,i_2)+scor(x_1[i_1],\_), S(i_1,i_2-1)+scor(\_, x_2[i_2])\}$$

Obs: valorile care se scad din indici sunt reprezentate de perechile: (1,1), (1,0), (0,1) adica elementele produsului cartezian {0,1}x{0,1}

Deci

$$S(i_1,i_2) =\max\{S(i_1-b_1,i_2-b_2)+scor(x_1[i_1*b_1],x_2[i_2*b_2]) \mid (b_1,b_2) \text{ din } \{0,1\}^2-\{(0,0)\}\}$$

unde x[0] este prin conventie '_'

Alinierea multiplă– programare dinamică

Generalizare pentru cazul a K secvențe: $x_1,x_2,...x_K$ de lungimi $n_1,n_2,...n_K$

In cazul general relatia de calcul a scorului de aliniere devine:

$$S(i_1,i_2,...,i_k) = \max\{S(i_1-b_1,i_2-b_2,\ldots,i_K-b_K)+scorMultiplu(x_1[i_1*b_1],x_2[i_2*b_2],\ldots,x_K[i_K*b_K]) \mid (b_1,b_2,\ldots,b_k) \text{ din } \{0,1\}^k-\{(0,0,\ldots,0)\}\}$$

unde x[0] este prin conventie '_' iar

$$scorMultiplu(x_1[j_1],x_2[j_2],\ldots,x_k[j_k]) = scor(x_1[j_1],x_2[j_2])+scor(x_1[j_1],x_3[j_3]) + \ldots + scor(x_{k-1}[j_{k-1}],x_k[j_k])$$

(suma scorurilor corespunzatoare perechilor de elemente)

Alinierea globală se construiește pornind de la $S(n_1,n_2,...,n_k)$

Ordin complexitate: $O(k^2 2^k n_1 n_2 ... n_k)$;

Concluzie: abordarea bazată pe programare dinamică este ineficientă în cazul multor secvențe

Alinierea multipla

Idee: utilizarea alinierilor la nivel de perechi pentru a construi o aliniere multiplă

Observație: Fiecare aliniere multiplă induce alinieri (nu neapărat optimale) ale tuturor perechilor de câte două secvențe

Exemplu: alinierea triplă

```
x:   AC-GCGG-C
y:   AC-GC-GAG
z:   GCCGC-GAG
```

induce următoarele alinieri la nivelul perechilor (perechile de gap-uri corespondente se ignoră):

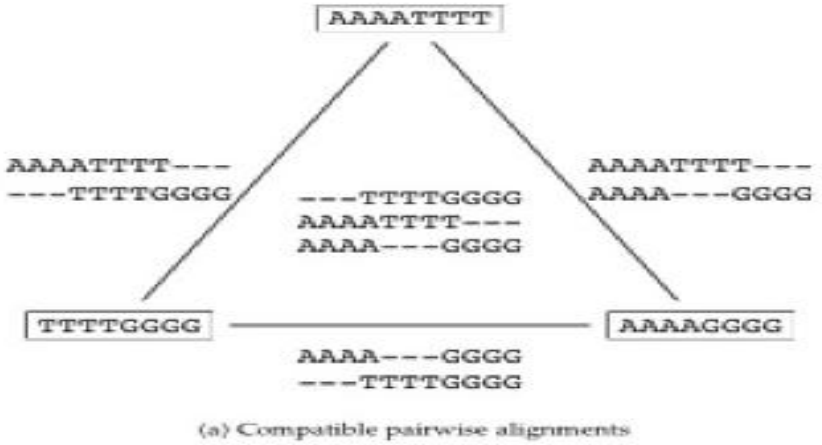
```
x: ACGCGG-C;   x: AC-GCGG-C;   y: AC-GCGAG
y: ACGC-GAC;   z: GCCGC-GAG;   z: GCCGCGAG
```

Alinierea multipla

Este adevarată și afirmația inversă ?

Pornind de la alinieri de perechi poate fi construită o aliniere multiplă ?

NU întotdeauna! Perechile de alinieri pot fi inconsistente



(a) Compatible pairwise alignments

(b) Incompatible pairwise alignments

Alinierea multipla – algoritm aproximativ

Intrare: set de k secvențe de aliniat: $x_1, x_2, ..., x_K$

Ieșire: aliniere sub-optimală

Idee algoritm (center star method):

- Se construiesc toate alinierile de perechi
- Pentru fiecare aliniere a două secvențe x_i și x_j se calculează distanța

  $d(x_i, x_j)$=numărul de poziții în care secvențele aliniate diferă
- Pentru fiecare secvență se calculează suma distanțelor corespunzătoare alinierilor cu celelalte secvențe
- Din setul de secvențe de aliniat $x_1, x_2, ..., x_K$ se alege secvența x_c (secvența centrală) care minimizează suma distanțelor față de celelalte secvențe
- Se construiește alinierea multiplă pornind de la secvența centrală și alinierile acesteia cu celelalte secvențe

Alinierea multipla – algoritm aproximativ

Exemplu:

S_1: CCTGCTGCAG
S_2: GATGTGCCG
S_3: GATGTGCAG
S_4: CCGCTAGCAG
S_5: CCTGTAGG

	S_1	S_2	S_3	S_4	S_5
S_1	0	4	3	2	4
S_2		0	1	6	5
S_3			0	5	5
S_4				0	4
S_5					0

$\Sigma_{l=1..k} D(S_1,S_l) = 13$
$\Sigma_{l=1..k} D(S_2,S_l) = 16$
$\Sigma_{l=1..k} D(S_3,S_l) = 14$
$\Sigma_{l=1..k} D(S_4,S_l) = 17$
$\Sigma_{l=1..k} D(S_5,S_l) = 18$

S_1: CCTGCTGCAG
S_2: GATG-TGCCG

S_1: CCTGCTGCAG
S_3: GATG-TGCAG

S_1: CCTGCT-GCAG
S_4: CC-GCTAGCAG

S_1: CCTGCT-GCAG
S_5: CCTG-TAG--G

S_1: CCTGCT-GCAG
S_2: GATG-T-GCCG
S_3: GATG-T-GCAG
S_4: CC-GCTAGCAG
S_5: CCTG-TAG--G

Obs:

- suma distanțelor dintre perechile de elemente ale alinierii generate cu “center star method” este cel mult de 2 ori mai mare decât valoarea optimă a sumei
- dacă secvențele de aliniat sunt de lungime O(n) atunci ordinul de complexitate a algoritmului este $O(k^2n^2)$

Sursa: W.K. Sung – Algorithms in Bioinformatics. A practical introduction. CRC Press, 2010

Alinierea multipla – abordare progresiva

- Alinierea multiplă poate fi construită în manieră progresivă, în sensul că se pornește de la alinierea a două secvențe și se extinde succesiv alinierea cu câte o secvență

- Problemă:
  - În ce ordine se adaugă secvențele la aliniere?

- Variante:
  - La fiecare etapă se decide care dintre secvențe se adaugă alinierii calculând scorul de aliniere dintre secvențele rămase și alinierea curentă
  - Se utilizează informații de ghidare (de exemplu un arbore de ghidare ale cărui noduri corespund secvențelor și ale cărui muchii conectează nodurile corespunzătoare secvențelor care ar trebui aliniate direct)

Alinierea multipla – abordare progresiva

Varianta 1 (fără informații de ghidare)

- Din cele C(K,2) perechi de secvențe se identifică cele mai similare două secvențe (alinierea caracterizată prin cel mai mare scor sau prin cea mai mică distanță)
- Cele două secvențe aliniate se “reunesc” obținându-se un set de K-1 secvențe care trebuie aliniate. Se aplică iterativ aceeași strategie la fiecare iterație alegând perechea cu scorul maxim de aliniere (pe parcursul prelucrării secvențele sunt înlocuite cu “reuniuni de secvențe aliniate”)

Probleme:

- Ce înseamnă “reuniunea” a două secvențe și cum poate fi descrisă?
- Cum va fi reprezentat rezultatul alinierii ?
- Cum se evaluază alinierea ?

Răspuns: folosind conceptul de profil al unei alinieri, alinierea în raport cu un profil și scoruri calculate pe baza profilelor

Alinierea multipla – abordare progresiva

Remember:

Profil = tabel cu frecvențele (relative sau absolute) ale nucleotidelor/ aminoacizilor corespunzătoare fiecărei coloane din matricea cu secvențele aliniate

Exemplu:

– A G G C T A T C A C C T G
T A G – C T A C C A – – – G
C A G – C T A C C A – – – G
C A G – C T A T C A C – G G
C A G – C T A T C G C – G G

A	0	1	0	0	0	0	1	0	0	0.8	0	0	0	0
C	0.6	0	0	0	1	0	0	0.4	1	0	0.6	0.2	0	0
G	0	0	1	0.2	0	0	0	0	0	0.2	0	0	0.4	1
T	0.2	0	0	0	0	1	0	0.6	0	0	0	0	0.2	0
-	0.2	0	0	0.8	0	0	0	0	0	0	0.4	0.8	0.4	0

Alinierea multipla – abordare progresiva

Exemplu: s1: **GATTCA** s3: **GATATT**

s2: **GTCTGA** s4**: GTCAGC**

Scor potrivire=1, scor nepotrivire=penalizare gap=-1

Etapa 1: se aliniază toate perechile (C(4,2)=6) și se alege alinierea cu scorul maxim

```
s2  GTCTGA                 s1  GATTCA--
s4  GTCAGC (scor = 2)      s4  G—T-CAGC(scor = 0)

s1  GAT-TCA                s2  G-TCTGA
s2  G-TCTGA (scor = 1)     s3  GATAT-T (scor = -1)

s1  GAT-TCA                s3  GAT-ATT
s3  GATAT-T (scor  = 1)    s4  G-TCAGC (scor = -1)
```

Alinierea multipla – abordare progresiva

Exemplu: s1: **GATTCA** s3: **GATATT**
s2: **GTCTGA** s4**: GTCAGC**

Etapa 2: se reunesc s2 cu s4 conducând la s2/4: GTCt/aGa/c

Etapa 3: se rezolvă subproblema alinierii secvențelor
s1, s3 si s2/4

Dificultate: a treia secvență este mai degrabă un profil (profilul alinierii dintre secvențele s2 și s4):

	G	T	C	t/a	G	a/c
A	0	0	0	0.5	0	0.5
C	0	0	1	0	0	0.5
G	1	0	0	0	1	0
T	0	1	0	0.5	0	0
-	0	0	0	0	0	0

Apare astfel necesitatea alinierii unei secvențe cu un profil sau a două profile între ele

Alinierea multipla – abordare progresiva

- La alinierea dintre o secvență și un profil sau la alinierea a două profile scorul de potrivire corespunzător unei poziții se poate calcula ca medie a scorurilor tuturor perechilor de elemente aflate pe poziția respectivă

- Exemplu: scorul de potrivire între elementele din
  secvența s1: **GATTCA** și alinierea
  s2/4: **GTCt/aGa/c** se poate calcula astfel:
  - Pentru perechile în care intervin doar nucleotide (de exemplu A și G sau T și C) se folosește direct valoarea corespunzătoare din matricea de substitutie
  - Pentru perechile în care intervin mai multe nucleotide se calculează media aritmetică a scorurilor corespunzătoare tuturor perechilor posibile. De exemplu pentru perechea (T, a/c) scorul va fi (scor(T,A)+scor(T,C))/2

Alinierea multipla – abordare progresiva

- O altă variantă de calcul a scorului, în cazul în care intervin profile (sau alinieri), se bazează pe probabilitățile ce intervin în fiecare dintre profile și pe utilizarea unei măsuri entropice pentru scor:
  - Fiecărei coloane j a alinierii i se asociază :

$$S_E(j) = \sum_{n \in A} p_n(j) \ln p_n(j)$$

unde p_n este frecvența corespunzătoare nucleotidei/ aminoacidului n

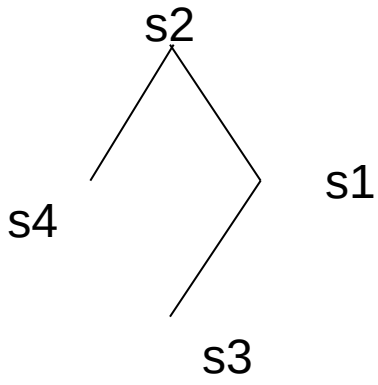
- Scorul asociat întregii alinieri este suma scorurilor coloanelor
- Suma este întotdeauna negativă; cu cât valoarea scorului este mai mare (mai apropiată de 0) cu atât este mai bună alinierea

Alinierea multipla – abordare progresiva

- Varianta 2 (utilizarea unui arbore de ghidare)

- Arbore de ghidare: arbore ale cărui noduri corespund secvențelor iar muchiile sunt etichetate cu valori ale scorului alinierii dintre perechile de secvențe corespunzătoare nodurilor.

Exemplu: s1: **GATTCA** s3: **GATATT**
s2: **GTCTGA** s4**: GTCAGC**



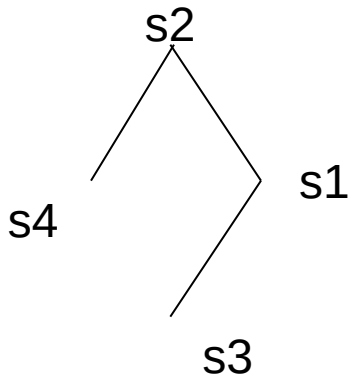
- Se aliniază s2 cu s4
- Se aliniază s1 cu s2
- Se aliniază s3 cu s1

Alinierea multipla – abordare progresiva

- Varianta 2 (utilizarea unui arbore de ghidare)

Exemplu: s1: **GATTCA** s3: **GATATT**

s2: **GTCTGA** s4**: GTCAGC**



- Se aliniază s2 cu s4:

  s2: GTCTGA

  s4: GTCAGC

- Se aliniază s1 cu s2 și se extinde alinierea anterioară pt a-l incorpora pe s1:

  s1: GAT-TCA

  s2: G-TCTGA

  s4: G-TCAGC (gap introdus în s4 pt a asigura alinierea globală cf principiului "once a gap always a gap")

- Se aliniază s3 cu s1: s1: GAT-TCA
- s3: GATAT -T

  s2: G-TCTGA

  s4: G-TCAGC

Alinierea multipla – abordare progresiva

- Algoritm de aliniere bazat pe arbore de ghidare:

Notații: T = arbore, M = set secvențe, A = set secvențe aliniate

Pas 1: alege s_i și s_j secvențe care corespund unor noduri conectate în T (se alege perechea pentru care eticheta muchiei e cea mai mare = cea mai bună aliniere). A= {s_i , s_j }; M=M- {s_i , s_j };

Pas 2: cât timp există secvențe nealiniate în M efectuează:

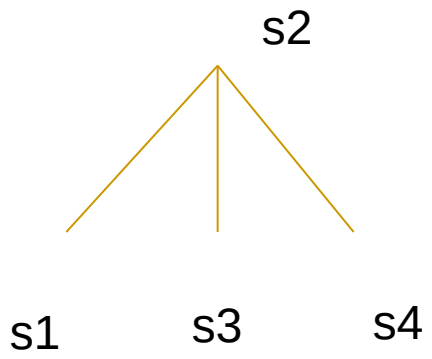
- Alege s_k din M și s_l din A astfel încât nodurile corespunzătoare să fie conectate în arbore (se poate alege perechea cu scorul maxim) și aliniază s_k cu s_l ; A=AU{s_k}, M=M-{s_k}. Pentru orice gap introdus în s_l se inserează gap în poziția corespunzătoare din toate secvențele aliniate (din A)

Alinierea multipla – abordare progresiva

- Problema: construirea arborelui de ghidare

- Există mai multe variante:
  - Arbore de tip stea: se alege un nod rădăcină care corespunde secvenței care maximizează scorul mediu de aliniere cu toate celelalte secvențe (similar metodei “center star”)
  - Arbore de construit pe baza unei matrici de distanțe între secvențe: se utilizează un algoritm specific construirii arborilor filogenetici (alg Neighbor-Joining – va fi prezentat în cursul 9).

  Obs: Este varianta folosită în ClustalW.



ClustalW

- Instrument pentru aliniere multiplă: ClustalW (Thompson et al, 1994)
- Algoritm de aliniere progresivă bazat pe un arbore de ghidare
- W – se referă la faptul ca secvențele au asociate ponderi utilizate în calculul scorului de similaritate

- **Etape:**

1.) Aliniază toate perechile (s_i, s_j) din setul de secvențe

2.) Construiește o matrice de distanțe având elementul de pe linia i coloana j, d(i,j) = 1-match(s_i, s_j)/length(aliniere s_i cu s_j)

3.) Construiește un arbore de ghidare pornind de la matricea de distanțe folosind tehnica Neighbor-Joining (vezi cursul următor referitor la analiza filogenetică)

4.) Realizează o aliniere progresivă folosind arborele de ghidare: la fiecare etapă se aliniază cele mai similare două secvențe/alinieri (conform arborelui de ghidare)

ClustalW

Caracteristici ale algoritmului ClustalW:

- Ponderile asociate secvențelor se stabilesc pe baza arborelului de ghidare astfel:
  - Secvențele având similaritate semnificativă cu alte secvențe au asociate ponderi mici
  - Secvențele divergente (aflate la distanță mare de celelalte) au asociate scoruri mai mari
- Pe parcursul procesului de aliniere se utilizează diferite matrici de substituție în funcție de gradul de similaritate dintre secvențe sugerat de arborele de ghidare. De exemplu pentru secvențe similare se folosește BLOSUM80 iar pentru secvențe disimilare se folosește BLOSUM30
- Valorile din matricile de substituție sunt translatate astfel încât să fie toate pozitive iar scorul potrivirii dintre gap-urile existente deja în aliniere și un alt simbol este setat pe 0.
- La calculul scorului dintre două secvențe sau alinieri (ca sumă de scoruri între perechi de simboluri) valoarea corespunzătoare matricii de substituție este multiplicată cu ponderile corespunzătoare secvențelor.

ClustalW

Caracteristici ale algoritmului ClustalW:

- Penalizarea pentru introducerea gap-ului este variabilă depinzând de:
  - Matricea de scor folosită, lungimile secvențelor, poziția gap-ului în secvență, prezența altor gap-uri în vecinătate, familia de aminoacizi din care fac parte simbolurile corespunzătoare din aliniere

- Resurse Web: http://www.ebi.ac.uk/clustalw/, http://www.clustal.org/

ClustalW

Exemplu (J.D. Thompson et al, Nucleic Acids Research, 1994)

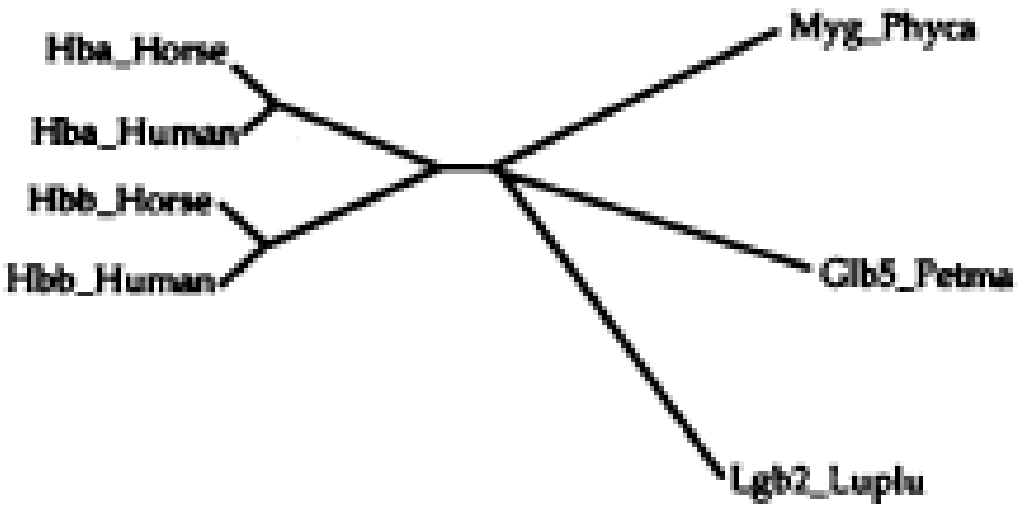
Etapa 1: construirea matricii de distanțe

Hbb_Human	1	-					
Hbb_Horse	2	.17	-				
Hba_Human	3	.59	.60	-			
Hba_Horse	4	.59	.59	.13	-		
Myg_Phyca	5	.77	.77	.75	.75	-	
Glb5_Petma	6	.81	.82	.73	.74	.80	-
Lgb2_Luplu	7	.87	.86	.86	.88	.93	.90
		1	2	3	4	5	6

ClustalW

Exemplu (J.D. Thompson et al, Nucleic Acids Research, 1994)

Etapa 2a: construirea arborelui filogenetic (alg. Neighbor Joining conduce la un arbore fără rădăcină)

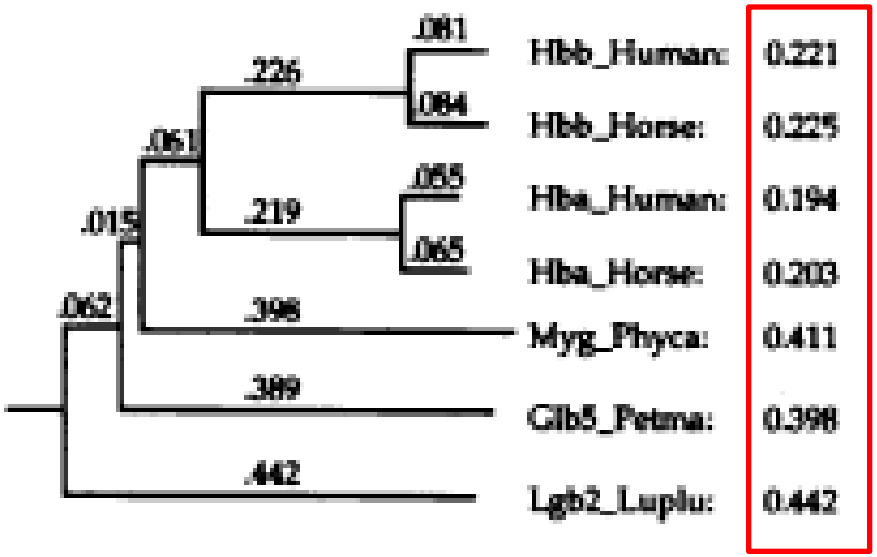


Obs: lungimile ramurilor sunt corelate cu similaritățile dintre secvențe

ClustalW

Exemplu (J.D. Thompson et al, Nucleic Acids Research, 1994)

Etapa 2b: transformarea arborelui filogenetic într-un arbore cu rădăcină și calculul ponderilor



Obs: ponderile se calculează cumulând lungimile ramurilor din arbore (lungimile ramurilor comune se partajează prin impărțirea la numărul de noduri frunză către care conduc)

Ex:

Myg_Phyca=0.398+0.015/5+0.062/6=

0.398+0.003+0.010=

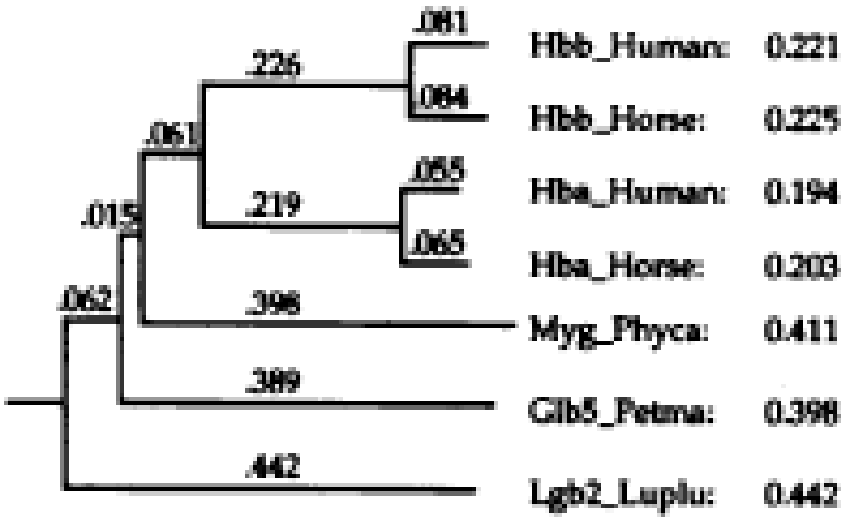
0.411

ClustalW

Exemplu (J.D. Thompson et al, Nucleic Acids Research, 1994)

Etapa 3: construirea alinierii:

- Se aliniază Hbb_human cu Hbb_Horse conducând la A1
- Se aliniază Hba_human cu Hba_Horse conducând la A2
- Se aliniază A1 cu A2 conducând la A3
- Se aliniază Myg_Phyca cu A3 conducând la A4
- Se aliniază Gib5_petna cu A4 conducând la A5
- Se aliniază Lgb2_Luplu cu A5



ClustalW

Exemplu (J.D. Thompson et al, Nucleic Acids Research, 1994)

Alinierea obținută:

Alte variante/implementări

- Clustal W2 conține ClustalW (1994), ClustalX (1997) – implementarea ClustalX oferă o interfață

  http://www.clustal.org/clustal2/

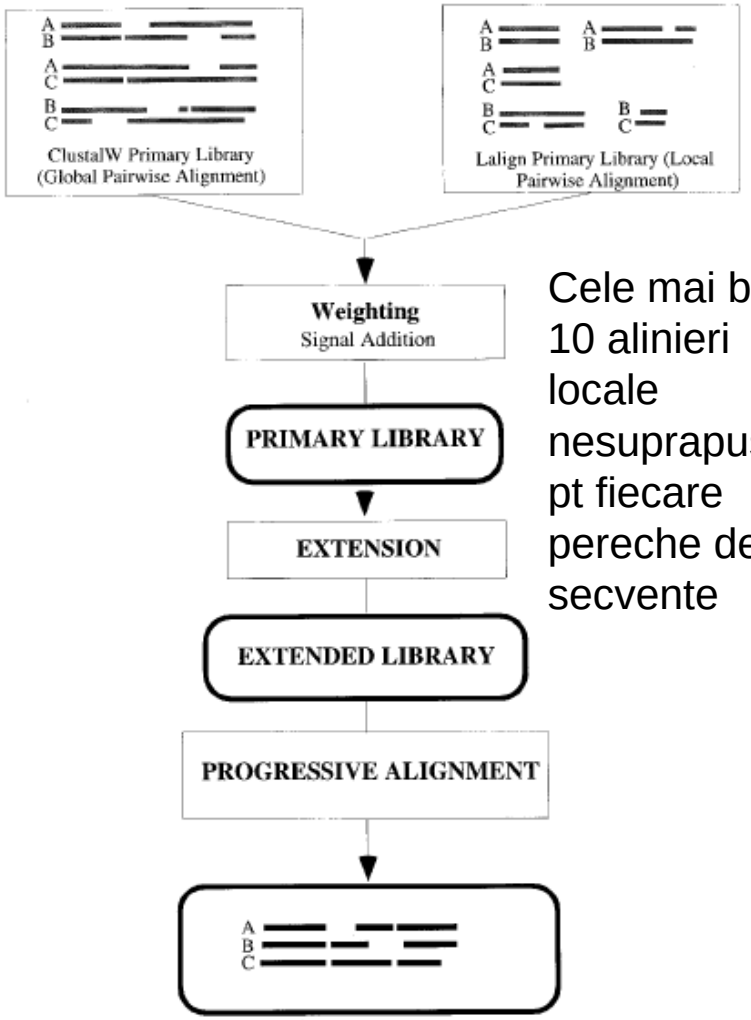
- ClustalOmega (F. Sievers et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega, Molecular Systems Biology 7: 539, 2011)
  - Scop: imbunătățire scalabilitate
  - utilizează un algoritm de complexitate O(nlog(n)) pentru construirea arborelui de ghidare (algoritmul clasic are complexitate $O(n^2)$)

  http://www.clustal.org/omega/

Alte variante/implementări

T-COFFEE (2000) - Tree-based Consistency Objective Function for alignment Evaluation
http://www.tcoffee.org/

- Scop: eliminarea dezavantajului abordării de tip greedy specifice tuturor metodelor de tip progresiv (eventualele erori ce apar în primele alinieri nu pot fi rectificate ulterior)
- Idee:
  - utilizarea în etapele inițiale atât a informațiilor privind alinierea globală cât și alinierea locală a secvențelor
  - Alinierea progresivă e similară celei din ClustalW
- Variante: M-COFFEE, R-COFFEE, Expresso



Cele mai bune 10 alinieri locale nesuprapuse pt fiecare pereche de secvente

Alte variante/implementări

MAFFT (2002) – Multiple Alignment using FFT

http://mafft.cbrc.jp/alignment/software/

- Scop: eficientizarea procesului de aliniere (de 100 de ori mai rapid decât T-COFFEE)
- Idee:
  - Pentru a identifica secvențe omoloage (despre care se poate presupune că provin de la o secvență anterioară comună) în secvențe de aminoacizi acestea sunt convertite în secvențe de valori corelate cu proprietățile fizico-chimice (valori corespunzătoare volumului și polarității)
  - Pentru două secvențe de valori x si y (asociate volumului și polarității) se determină valoarea corelației încrucișate: c(k)=x(1)*y(1+k)+x(2)*y(2+k)+... pentru diferite valori ale lui k (pentru eficientizarea calculului se folosește transformata Fourier rapidă); secvențele omoloage corespund unor valori mari ale lui c(k)
  - Pentru secvențele omoloage identificate se construiește o matrice de scor utilizată pentru un algoritm de aliniere bazat pe programare dinamică
  - Această strategie de aliniere este extinsă și în cazul grupurilor de secvențe deja aliniate (pentru un grup de secvențe valorile volumului/polarității se calculează pornind de la valorile corespunzătoare elementelor din secvențele grupului)

Alte variante/implementări

- MUSCLE (2004) – MUltiple Sequence Comparison by Log-Expectation

http://drive5.com/muscle/

Scop:

- eficientizarea alinierii unui număr mare de secvențe
- reducerea influenței alinierii inițiale asupra calității alinierii finale

Idei:

a) Pt calculul matricilor de distanțe între secvențe se folosesc :

- O distanță bazată pe numărul de k-tuple comune (în cazul secvențelor nealiniate)
- O distanța de tip Kimura (în cazul secvențelor aliniate) :

$d=-\ln(1-2p-q)/2-\ln(1-2q)/4$

unde

p=proporția de substituții între nucleotide din aceeași famile (A<->G, C<->T)

iar

q =proporția de substituții între nucleotide din familii diferite

Alte variante/implementări

- MUSCLE (2004) – MUltiple Sequence Comparison by Log-Expectation

http://drive5.com/muscle/

Idei:

b) Pt alinierea a două grupări (sau profile) se utilizează un scor specific:

$$LE(x,y) = \big(1 - f_G(x)\big)\big(1 - f_G(y)\big) \log \sum_{i,j} f_i(x) f_j(x) \frac{p_{ij}}{p_i p_j}$$

Notații

x = indice de coloană în primul profil; y=indice de coloană în al doilea profil

$f_G(x)$ = fracțiune de gap-uri în coloana x a primului profil; $f_G(y)$: similar pt al doilea profil

$f_i(x)$ = frecvența simbolului i pe coloana x a primului profil; $f_j(y)$ = pt al doilea profil

p_{ij}, p_i, p_j = probabilități de apariție a perechii (i,j) sau a simbolurilor i respectiv j

Obs: $p_{ij}/(p_i\, p_j)$ poate fi înlocuit cu elementul corespunzător perechii (i,j) dintr-o matrice de substituție (de exemplu PAM)

Alte variante/implementări

- MUSCLE (2004)

Specific:

- Două etape de aliniere progresivă
- Utilizare UPGMA (in loc de Neighbour-Joining) pt construirea arborelui de ghidare
- Proces iterativ de realiniere bazat pe scindarea aleatoare a arborelui de ghidare

Sursa: R. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput, 1792-1797 Nucleic Acids Research, 2004, Vol. 32, No. 5

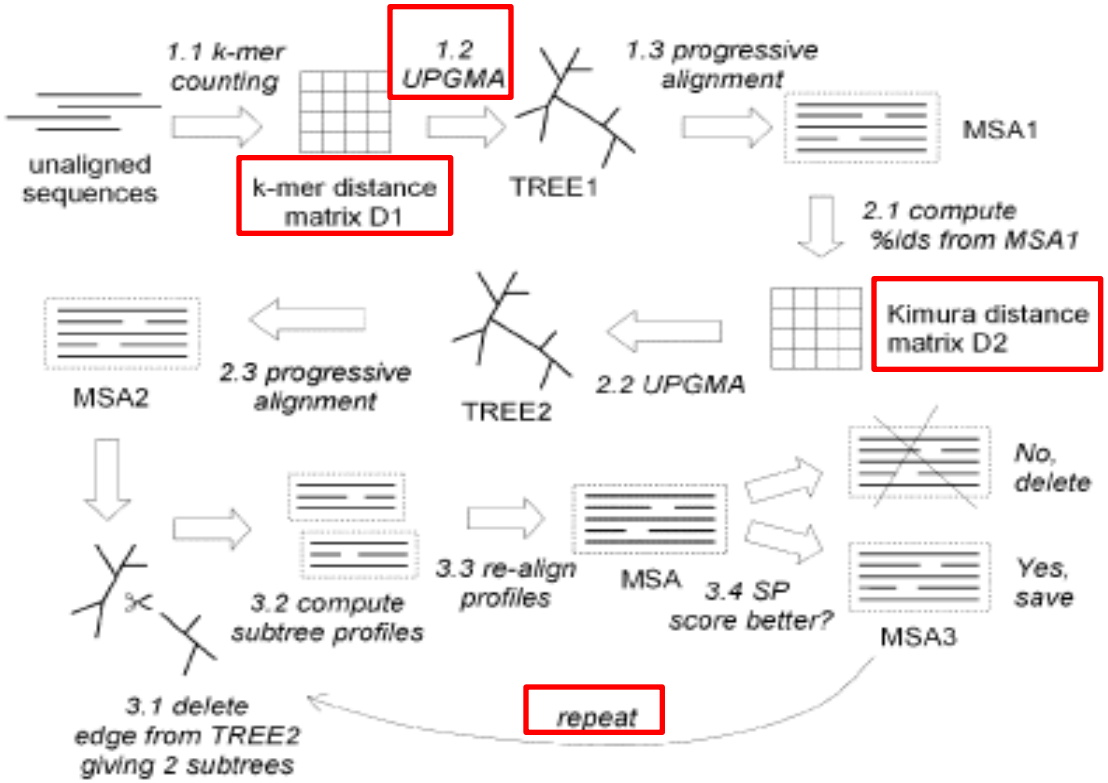


Figure 2. This diagram summarizes the flow of the MUSCLE algorithm. There are three main stages: Stage 1 (draft progressive), Stage 2 (improved progressive) and Stage 3 (refinement). A multiple alignment is available at the completion of each stage, at which point the algorithm may terminate.

Alte variante/implementări

- Resurse web:

http://www.ebi.ac.uk/Tools/msa/

- Tendința curentă:
  - Implementări eficiente ale algoritmilor exacți utilizând calculul de înaltă performanță (ex: implementări pe GPU)
  - http://gpualign.cs.put.poznan.pl/