Alignement multiple

30 octobre 2023 08:57

Entrée : séquence $S = \{S_1, S_2, \ldots, S_m\}$

Sortie : séquences $S_1', S_2', \ldots, S_m'$

- toutes de la même longueur
- $\forall i = 1 \ldots m$, S_i' est obtenu de S_i par l'insertion de gaps "_"
- critères d'optimisation variable

Critère populaire : "sum-of-pairs" (SP)
somme des paires
On additionne le score entre chaque paire de séquences alignées.

ex : S_1 CAACGAT — match +2
S_2 CAAGGGGAT — autre −1
S_3 GAACG — gap avec gap 0
S_4 CACGAT

−1 +2 +2 +2 +2 0 0 −1 −1

sol : S_1' C A A C G _ _ A T] 9]
S_2' C A A G G G G A T] 5]
S_3' G A A C G _ _ _ _]
S_4' C A − C G _ _ A T]

$$score(S_i', S_j') = \sum_{k=1}^{|S_i'|} M[S_i'[k], S_j'[k]]$$

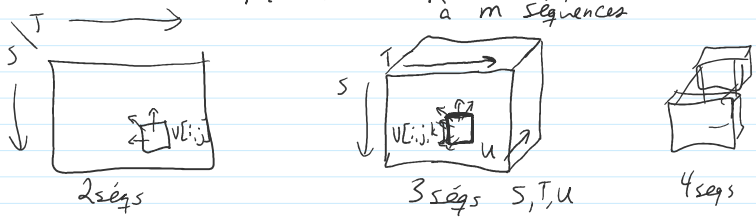
$$SP(S_1', S_2', \ldots, S_m') = \sum_{i=1}^{m} \sum_{j=i+1}^{m} score(S_i', S_j')$$

- Trouver les $S_1', S_2', \ldots, S_m'$ qui minimisent le score SP est NP-complet — sûrement pas d'algo polynomial
- Heuristiques / approximation / algo exact exponentiel

Algo exact exponentiel

Temps $O(n^m \cdot 2^m)$ — n = longueur max des S_i, m = # de séquences

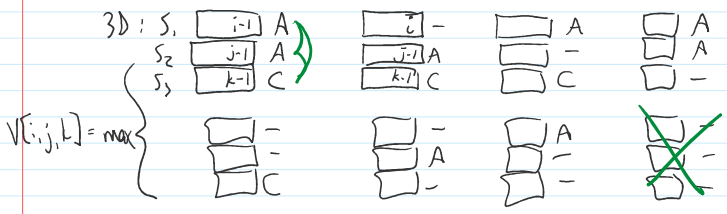
Généralisation de la prog. dynamique pour 2 séquences à m séquences



- $m = 3$ (3 séquences S_1, S_2, S_3)

$V[i,j,k]$ = score optimal d'un alignement multiple entre $S_1[1..i]$, $S_2[1..j]$, $S_3[1..k]$

2) : A / A ; - / A ; A / -



De façon plus générale

m séquences $S_1, S_2, \dots, S_m$

$V[i_1, i_2, \dots, i_m]$ = score opt entre $S_1[1..i_1], \dots, S_m[1..i_m]$

$V[i_1, i_2, \dots, i_m] = \max$ { A ou – 2 choix; C ou – 2 choix; A ou – 2 choix; … … ; 2 choix }

ce max doit énumérer $2^m - 1$ combinaisons de gaps/non-gaps

$$V[i_1, i_2, \dots, i_m] = \max_{(b_1, b_2, \dots, b_m) \in \{0,1\}^m} \left(V[i_1 - b_1, i_2 - b_2, \dots, i_m - b_m] + score(b_1, b_2, \dots, b_m) \right)$$

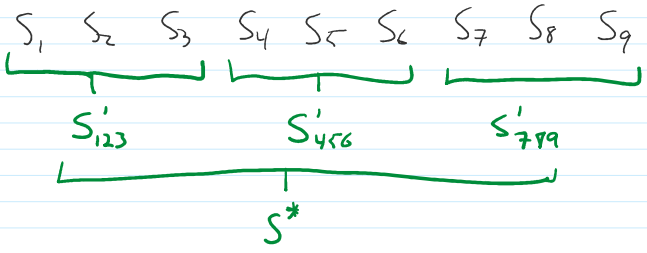
vecteurs de m bits

Complexité: V est une table de dimension $|S_1| \cdot |S_2| \cdot \dots \cdot |S_m| \in O(n^m)$

Chaque entrée prend un temps $O(2^m)$

$\Rightarrow O(n^m \cdot 2^m)$

Utilisation:

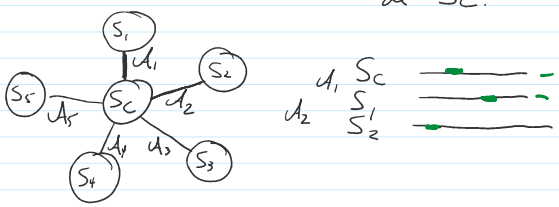


Approximation par centre-étoile

S_1 —
S_2 —
…
S_m —

- Idée: choisir S_c (séquence centrale) $c \in \{1, 2, \dots, m\}$

  Aligner tous les S_i en fonction de S_c.



Pour choisir S_c, on prend S_i qui maximise

$\sum_{j=1}^{m}$ score(S_i, S_j) // alignement

Pour choisir S_c, on prend S_i qui maximise

$$\sum_{j=1}^{m} score(S_i, S_j)$$ // score = alignement multiple entre 2 séqs

n = max $|S_i|$

m = # de séquences

choisir Sc ($S_1, S_2, \dots, S_m$)

cur Sc = null, curmax = $-\infty$

m iter — pour i = 1 .. m

m iter — score = 0

m^2 — pour j = 1 .. m

x — score += alignGlobal(S_i, S_j) $O(n^2)$

x — si score > curmax

x — cur Sc = S_i

x — curmax = score

return cur Sc

Complexité: $O(m^2 \cdot n^2)$

ex: S_1 A C A A T

S_2 C C C T T

S_3 A A G A T

S_4 A G A

On suppose que $S_c = S_1$

A: $S_c = S_1$ A C - A - A T

S_2 - C - C C T T

S_3 A - G A - - T

S_4 A G - A - - -

Ajout de S_2

Alignement opt S_c - S_2

S_c A C A - A T

S_2 - C C C T T

Ajout de S_3

S_c A C - A A T

S_3 A - G A - T

Ajout de S_4

S_c A C A A T

S_4 A G A - -

algo Centre Etoile ($S_1, S_2, \dots, S_m$)

$O(n^2 m^2)$ S_c = choisir Sc($S_1, \dots, S_m$) // $O(m^2 n^2)$

$A = S_c$

// Supposition: on stocke chaque // align. global calculé dans // choisir Sc

$O(m)$ iter. pour chaque $S_i \neq S_c$

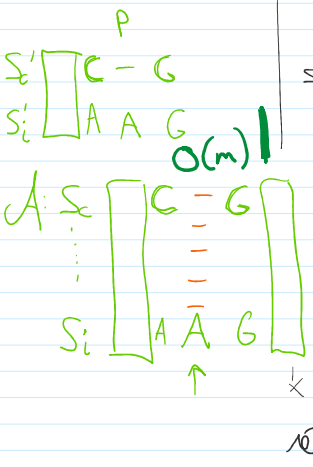
$O(1)$ (S_c', S_i') = alignementGlobal(S_c, S_i)

$O(n)$ iter pour chaque position p de S_c'

si $S_c'[p] \neq$ "-"

ajouter le caractère $S_i'[p]$ à la dernière rangée de A, à la position appropriée

p



ajouter le caractère $S_i[p]$ à la dernière rangée de A, à la position appropriée
si $S_c'[p] = "-"$

ajouter une colonne de gap dans A, entre les positions appropriées, avec $S_i[p]$ à la position appropriée

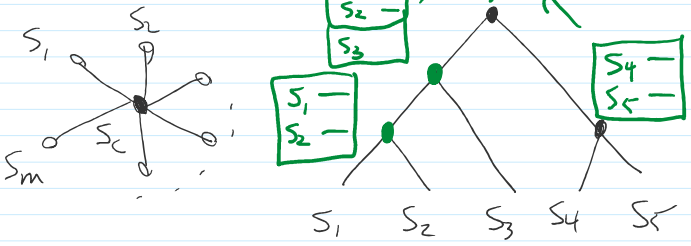
retourner A

Complexité: $O(m^2 n)$

Si on compte choisir S_c:
$O(n^2 m^2 + m^2 n) = O(n^2 m^2)$

Clustal

- Utilisation d'un arbre guide pour déterminer l'ordre d'insertion des séquences dans notre align. multiple



// note: noeud ≠ feuille
// noeud = noeud interne + feuilles

} feuilles = séquences

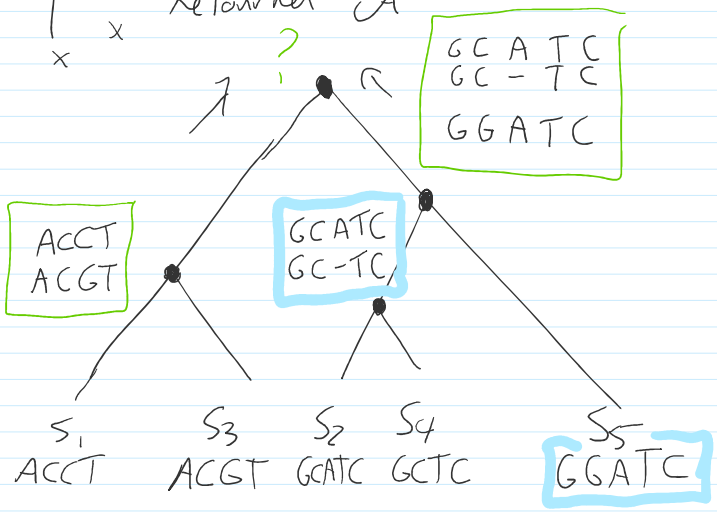
clustal(arbre guide T, $S_1, S_2, \ldots, S_m$)

quand on traite v, tous ses enfants sont traités

pour chaque noeud v de T dans un parcours post-ordre

si v = feuille,
  retourner la séquence S_i assignée à v
sinon
  soit v_1, v_2 les enfants de v
  soit A_1, A_2 les alignements à v_1 et v_2 (respectivement)
  A = combiner(A_1, A_2)

Soit $\mathcal{A}_1, \mathcal{A}_2$ les alignements de v_1 et v_2 (respectivement)
$\mathcal{A} =$ combiner$(\mathcal{A}_1, \mathcal{A}_2)$
retourner $\mathcal{A}$



S_1 ACCT
S_2 GCATC
S_3 ACGT
S_4 GCTC
S_5 GGATC

Comment combiner deux alignements multiples?

Entrée : alignements $\mathcal{A}_1, \mathcal{A}_2$
Sortie : alignement $\mathcal{A}$ avec les rangées de $\mathcal{A}_1$ et $\mathcal{A}_2$

$\mathcal{A}_1$:

B_1	B_2	B_3	B_4
A	C	C	T
A	C	G	T

On traite B_1, B_2, B_3, B_4 comme des symboles d'un nouvel alphabet

$\mathcal{A}_2$:

G	C	A	T	C
G	C	-	T	C
G	G	A	T	C
D_1	D_2	D_3	D_4	D_5

Même chose avec les D_i.

On doit aligner deux séquences

$\mathcal{A}_1'$ $B_1 B_2 B_3 B_4$
$\mathcal{A}_2'$ $D_1 D_2 D_3 D_4 D_5$

On doit définir une matrice de scores sur ce nouvel alphabet

$M[B_i, D_j] = ?$
$M[B_i, "-"] = ?$
$M["-", D_j] = ?$

$$M[B \cdot D \cdot] = \sum_{\cdot}^{|B|} \sum^{|D_j|} M[B \cdot [b], D \cdot [d]]$$

$$M[B_i, D_j] = \sum_{b=1}^{|B_i|} \sum_{d=1}^{|D_j|} M[B_i[b], D_j[d]]$$

ex: $M[B_1, D_1] = M\left[\begin{matrix} A \\ A \end{matrix}, \begin{matrix} G \\ C \\ G \end{matrix}\right] = \underbrace{M[A,G] + M[A,G] + \dots + M[A,G]}_{2 \times 3 = 6 \text{ fois}}$

$$M[B_3, D_3] = M\left[\begin{matrix} C \\ G \end{matrix}, \begin{matrix} A \\ - \\ A \end{matrix}\right] = M[C,A] + M[C,-] + M[C,A] + M[G,A] + M[G,-] + M[G,A]$$

Une fois que $M[B_i, D_j]$ est calculé, $\forall i, j$, on exécute Needleman-Wunsch avec ce M et les B_i, D_j

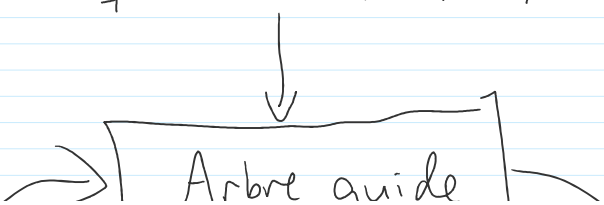
ex:

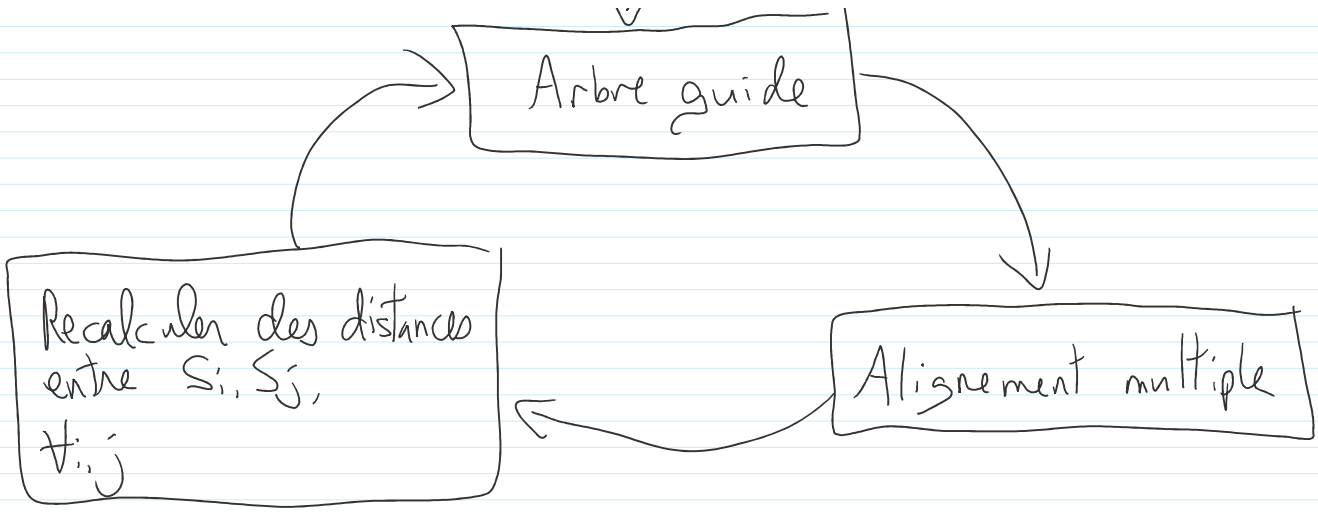
$B_1\ B_2\ -\ B_3\ B_4$
$D_1\ D_2\ D_3\ D_4\ D_5$

A	C	-	C	T	S_1
A	C	-	G	T	S_2
G	C	A	T	C	S_3
G	C	-	T	C	S_4
G	C	A	T	C	S_5

Autres approches : MUSCLE, MAFFT, ...

Séquences $S_1, S_2, \dots, S_m$




Arbre guide
Alignement multiple
Recalculer des distances entre S_i, S_j, $\forall i, j$