Alignement multiple de séquences

Plan de la présentation

I. Introduction – Définition et signification biologique
II. Modèles de comparaisons
III. Alignements pour le score ``sum-of-pairs''
    1. Méthode exacte
    2. Accélération de la méthode exacte
    3. Heuristique bornée
IV. Alignement phylogénétique
V. Heuristiques usuelles
    1. Méthode progressive
    2. Méthode itérative
    3. Méthode par points d'ancrage

I. Introduction à l'alignement multiple

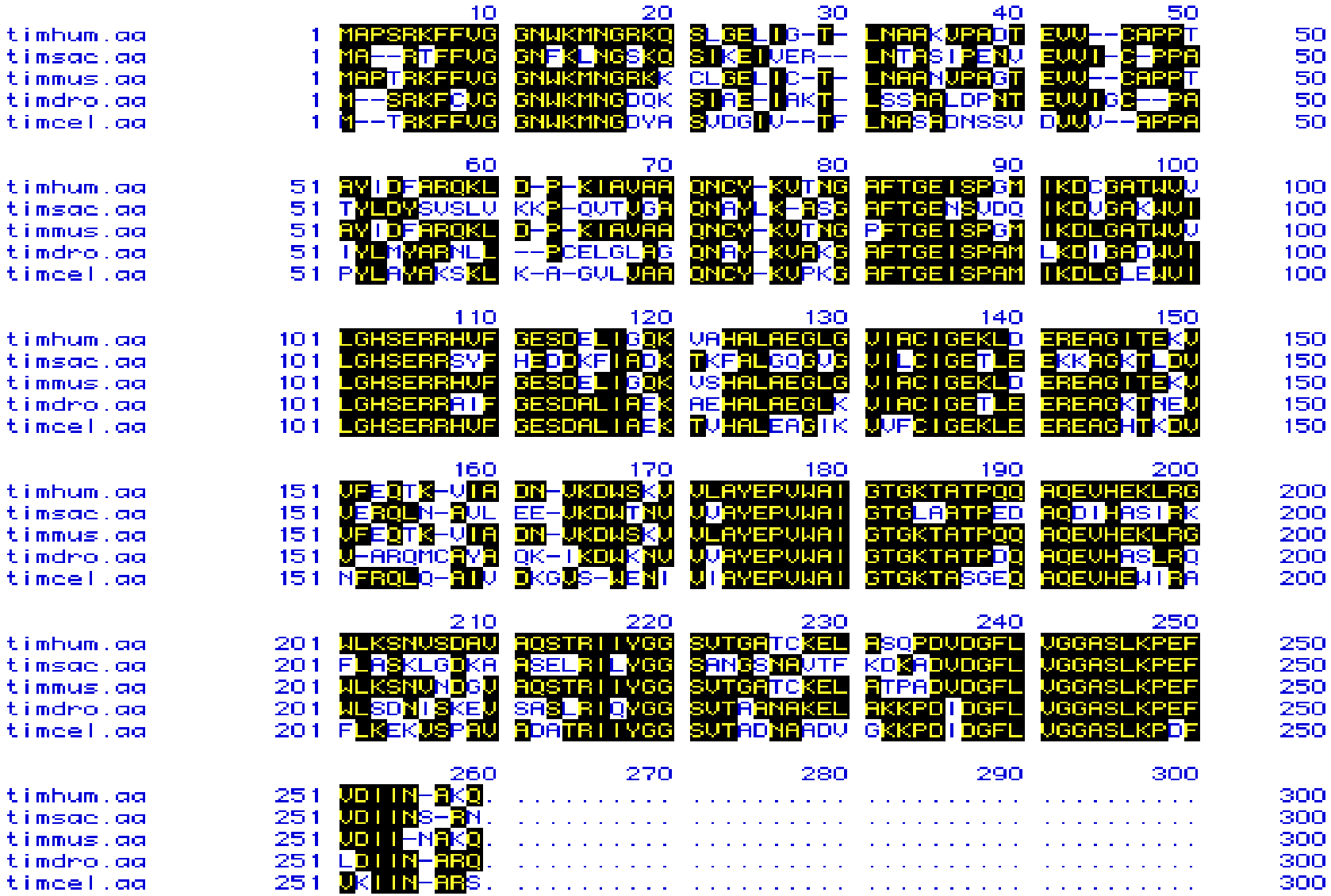
Généralisation de l'alignement de 2 séquences

Données: Un ensemble de séquence homologues (nucléotides ou AA): S_1, S_2, …, S_k

Alignement multiple: Matrice A = (a_{ij}), $1 \le i \le k$; $1 \le j \le l$.

a_{ij} symboles de l'alphabet ou '-', tq contaténation des caractères à la ligne i produit S_i

$$\begin{bmatrix} A & A & G & A & A & - & A \\ A & T & - & A & A & T & G \\ C & T & G & - & G & - & G \\ C & C & - & A & G & T & T \\ C & C & G & - & G & - & - \end{bmatrix}$$


10 20 30 40 50
timhum.aa 1 MAPSRKFFVG GNWKMNGRKQ SLGELIGTLN AAKVPADTEV V--CAPPTAY 50
timsac.aa 1 MA--RTFFVG GNFKLNGSKQ SIKEIVERLN TASIPENVEV VI-C-PPAAY 50
timmus.aa 1 MAPTRKFFVG GNWKMNGRKK CLGELICTLN AANVPAGTEV V--CAPPTAY 50
timdro.aa 1 M--SRKFCVG GNWKMNGDQK SIAELIAKLS SAALDPNTEV VIGC--PAAY 50
timcel.aa 1 M--TRKFFVG GNWKMNGDVA SVDGIV--TF LNASADNSSV DVVV--APPA 50
60 70 80 90 100
timhum.aa 51 AVIDFARQKL DPKIAVAAQN CYKVTNGAFT GEISPGMIKD CGATWVVLGH SERRHVFGES DELIGQKVAH ALAEGLGVIA CIGEKLDERE AGITEKVVFE QTKVIADNVK DWSKVVLAYE PVWAIGTGKT ATPQQAQEVH EKLRGWLKSN VSDAVAQSTR IIYGGSVTGA TCKELASQPD VDGFLVGGAS LKPEFVDIIN AKQ 200
timsac.aa 51 TVLDYSVSLV KKPQVTVGAQ NAYLKASGAF TGENSVDQIK DVGAKWVILG HSERRSYFHE DDKFIADKTK FALGQGVSVI LCIGETLEEK KAGKTLDVVF ERQLNAVLEE VKDWTNVVVA YEPVWAIGTG LAATPEDAQD IHASIRKFLA SKLGDKASEL RILYGGSANG SNAVTFKDKA DVDGFLVGGA SLKPEFVDII NSN 200
timmus.aa 51 AVIDFARQKL DPKIAVAAQN CYKVTNGAFT GEISPGMIKD LGATWVVLGH SERRHVFGES DELIGQKVSH ALAEGLGVIA CIGEKLDERE AGITEKVVFE QTKVIADNVK DWSKVVLAYE PVWAIGTGKT ATPQQAQEVH EKLRGWLKSN VNDGVAQSTR IIYGGSVTGA TCKELATPAD VDGFLVGGAS LKPEFVDIIN AKQ 200
timdro.aa 51 IVLMVARNLL PCELGLAGQN AVKVAKGAFT GEISPAMLKD IGADWVILGH SERRAIFGES DALIAEKAE HALAEGLKVI ACIGETLEER EAGKTNEVVF EQTKVIADNV KDWSKVVLAY EPVWAIGTGK TASPQQAQEV HEWIRAWLKS NVSDAVSLRI QYGGSVTAAN AKELAKKPDI DGFLVGGASL KPEFVDIIN ARQ 200
timcel.aa 51 PYLAVAKSKL KAGVLVAAQN CYKVPKGAFT GEISPAMIKD LGLEWVILGH SERRHVFGES DALIAEKTV HALEAGIKVV FCIGEKLEER EAGHTKDVVF EQTKVIADNV KDWSKVVLAY EPVWAIGTGK TASGEQAQEV HEWIRAWLKS NFRQLQAIVY GKGVSWENIV IAYEPVWAIG TGKTASPEQA QEVHEWIRAY LRQWLKSNVS DAVAQSTRII YGGSVTADNA ADVGKKPDID GFLVGGASLK PDFVDIIN ARS 300

But de l'alignement multiple

- Trouver des caractéristiques communes à une famille de protéines
- Relier la séquence à la structure et à la fonction
- Caractériser les régions conservées et les régions variables
- Déduire des contraintes de structures pour les ARN
- Construire l'arbre phylogénétique des séquences homologues considérées
- Différencier entre gènes orthologues et gènes paralogues

Représentations d'une famille de séquences

- Séquence consensus:

Y	D	D	G	A	V	-	E	A	L
Y	D	G	G	-	-	-	E	A	L
F	E	G	G	I	L	V	E	A	L
F	D	-	G	I	L	V	Q	A	V
Y	E	G	G	A	V	V	Q	A	L
Y	D	G	G	A/I	V/L	V	E	A	L

- Signature ou motif conservé: Expression régulière

  G-{EDRKHPFYW}-x (2)-[STAGCN]- {P}

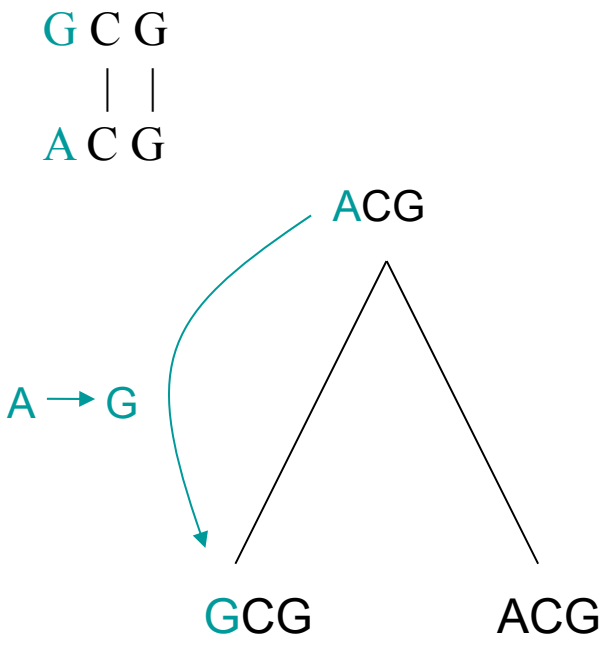
- Matrice consensus (ou profile): Taux d'apparition de chaque nuc. à chaque colonne de l'alignement multiple

C1	C2	C3	C4	C5
a	c	g	-	t
a	c	a	c	t
a	g	g	c	-
g	c	-	c	g

	C1	C2	C3	C4	C5
a	0.75	0	0.25	0	0
c	0	0.75	0	0.75	0
g	0.25	0.25	0.50	0	0.25
t	0	0	0	0	0.50
-	0	0	0.25	0.25	0.25

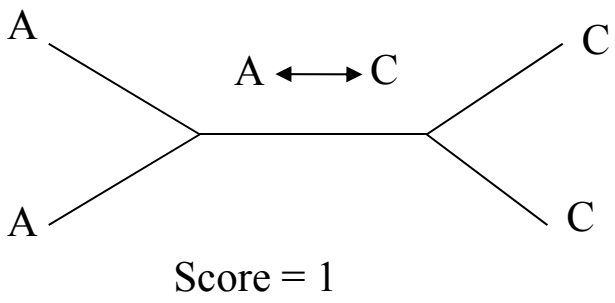
II. Modèles de comparaison

- Un *bon* alignement reflète le modèle d'évolution qui a donné lieu aux séquences
- Hypothèses:
  - les séquences à aligner descendent d'un ancêtre commun
  - Les séquences ont évolué par mutations ponctuelles

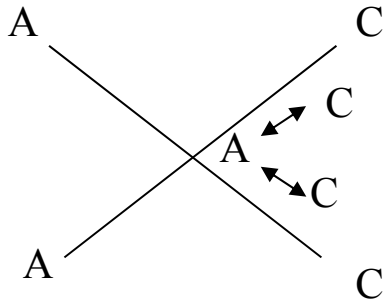


Pondération d'un alignement

- Par rapport à l'arbre phylogénétique produit. Garder l'alignement qui produit l'arbre de poids minimal. Complexité de calcul considérable



Score = 1

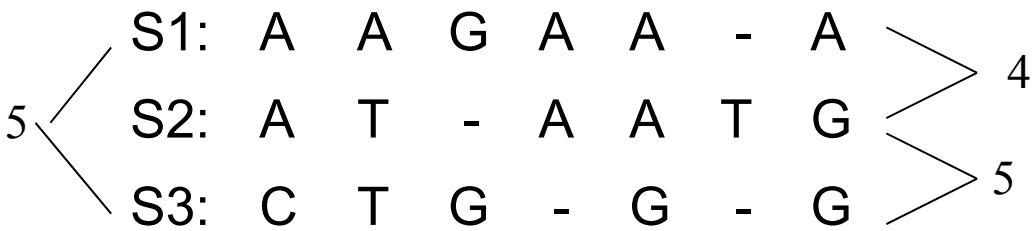


Score=2

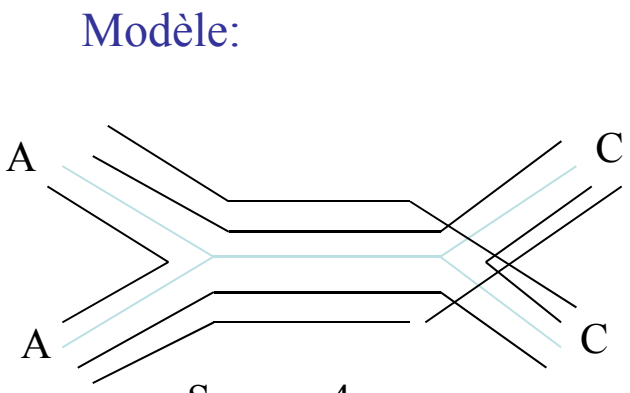
Score ``sum of pairs'' (SP)

- Généralisation du score utilisé pour l'alignement de deux séquences
- Le plus utilisé, bonnes propriétes théoriques et pratiques

Score SP d'un alignement *A* = somme des scores des alignements induits pour chaque paire de séquences dans *A*



Score SP = 14



Score = 4

III- Alignement pour le score SP
Méthode exacte

- Trouver un alignement multiple ayant un score SP minimum
- Problème NP-complet *(Wang and Jiang 1994)*
- Généralisation de l'alignement de deux séquences: si *m* séquences de taille *n*, algorithme en $O(n^m)$. Très inefficace dès que $m>5$ et $n\sim100$

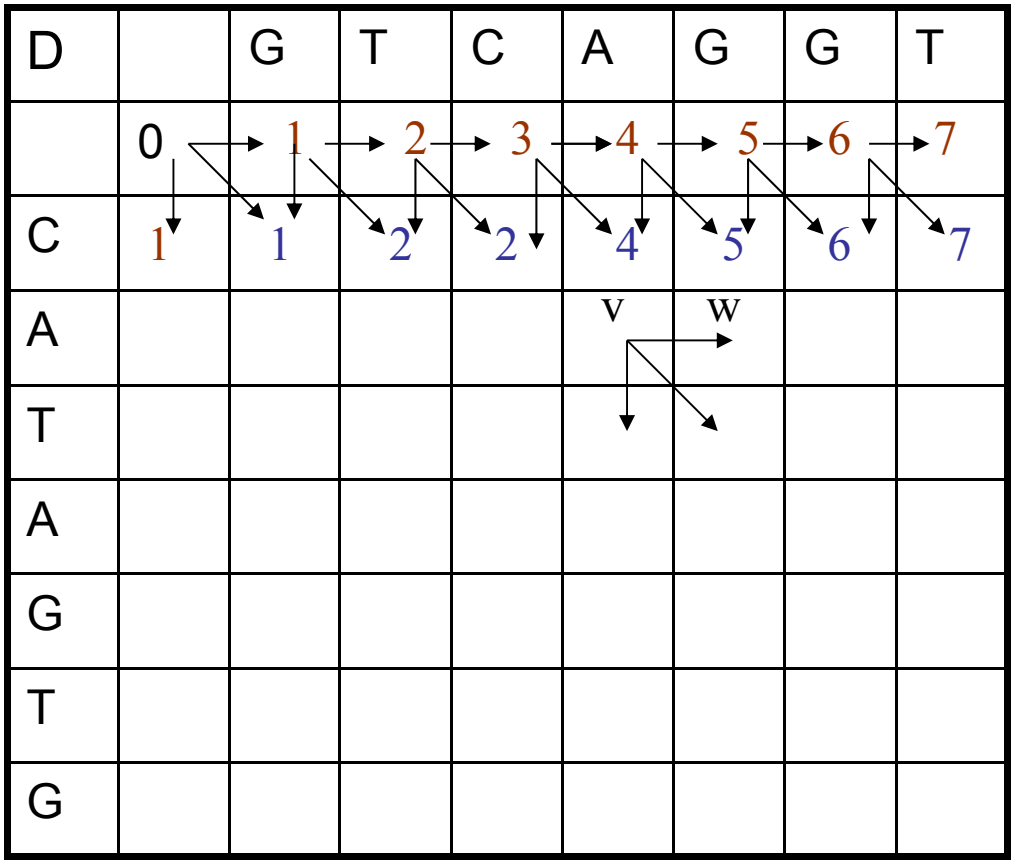
Solution exacte pour *n*=3

- On considère la distance d'édition avec pondération de l'alphabet.
- S,T,U trois seq. de tailles *n1, n2, n3*
- *D(i,j,k):* Score SP de l'al. op. de S[1..i], T[1..j], et U[1..k];
  b: score d'un blanc; *c(i,j):* sore de l'appariement *(S [i],T [j]).*
- Pour chaque case (i,j,k), examiner 7 cases voisines:
  - *d1= D(i-1,j-1,k-1)+c(i,j)+c(i,k)+c(j,k)*
  - *d2= D(i-1,j-1,k)+c(i,j)+2b; d3= D(i-1,j,k-1)+c(i,k) +2b;*
    d4= D(i,j-1,k-1)+c(j,k)+2b
  - *d5= D(i-1,j,k)+2b ; d6=D(i,j-1,k)+2b; d7=D(i,j,k-1)+2b.*
  - *D(i,j,k) = min(d1,d2,d3,d4,d5,d6,d7)*
- $D_{ST}(i,j)$: Score de l'al. Op. de S[1..i] et T[1..j]…
  - $D(i,j,0) = D_{ST}(i,j) +(i+j)b$; $D(i,0,k) = D_{SU}(i,k)+(i+k)b$;
    $D(0,j,k) = D_{TU}(i,k)+(i+k)b$

Algorithme MSA (Lipman *et al.* 1989)

- Calculer les alignements optimaux pour chaque paire de séquences
- Trouver un alignement multiple provisoire par une heuristique rapide: *z*
- Effectuer la programmation dynamique en scrutage avant dans un espace d'alignement restreint

Programmation dynamique avec scrutage avant



Les flèches vont de (i,j) à (i,j+1), (i+1,j) et (i+1,j+1)

p(v,w): Poids de la flèche de v à w

p(w): Valeur provisoire de D(w). Après calcul de D(v):

p(w) = min(p(w), D(v)+p(v,w))

Valeur de D(w) = valeur de p(w) après considération de tous les voisins de w

Programmation dynamique avec scrutage avant

D		G	T	C	A	G	G	T
	0	1	2	3	4	5	6	7
C	1	1	2	2	4	5	6	7
A	2	2	2					
T								
A								
G								
T								
G								

Les flèches vont de (i,j) à (i,j+1), (i+1,j) et (i+1,j+1)

p(v,w): Poids de la flèche de v à w

p(w): Valeur provisoire de D(w). Après calcul de D(v):

p(w) = min(p(w), D(v)+p(v,w))

Valeur de D(w) = valeur de p(w) après considération de tous les voisins de w

Algorithm:

- *q=(0,0)* (liste contenant les cases à considérer)
- Tant que *q* n'est pas vide faire

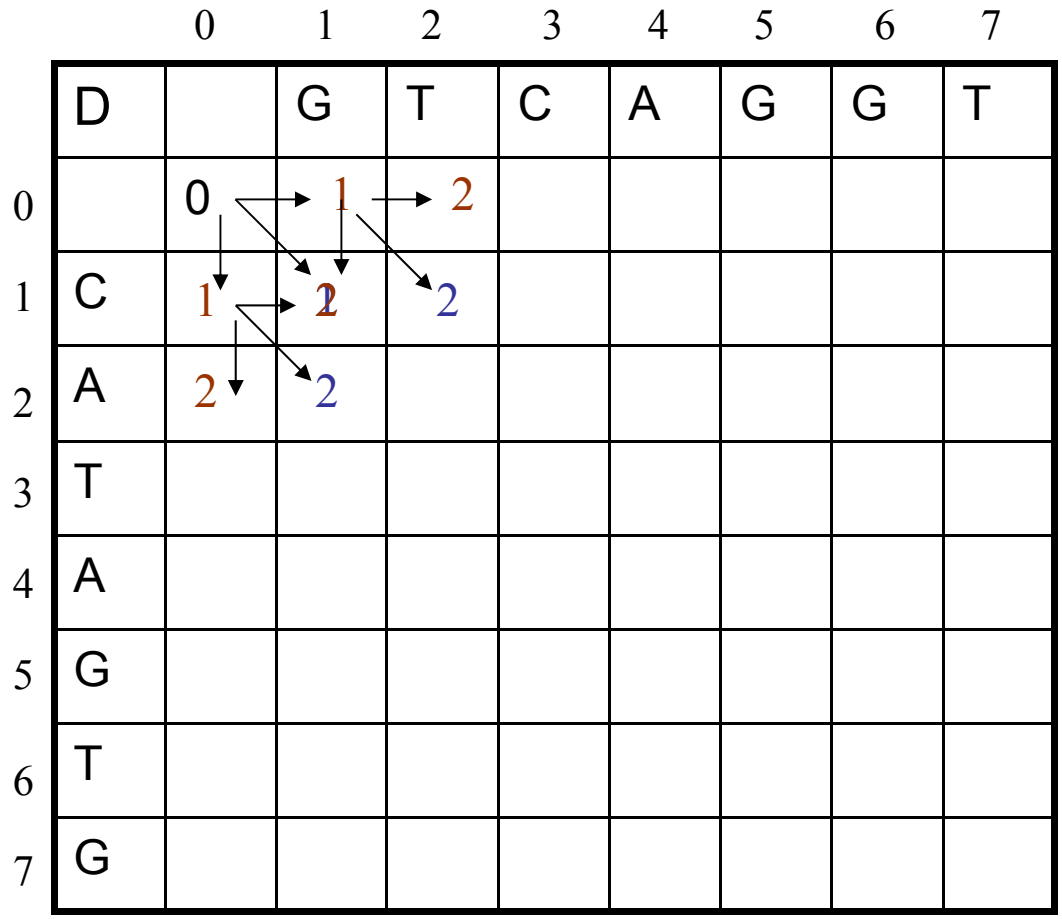
  v = première case de q;

  Supprimer *v* de *q; D(v)=p(v);*

  Si w=(i,j+1) pas dans q, le rajouter a la fin de q;

  p(w)=min(p(w),D(v)+p(v,w));

  Même chose pour w=(i+1,j) et w=(i+1,j+1)



q: ~~(0,0)~~ ~~(0,1)~~ ~~(1,0)~~ (1,1) (0,2) (1,2) (2,0) (2,1)

Accélération de l'alignement SP exact

- $ID_{ST}(i,j)$: Score de l'al. Op. de $S[i..n]$ et $T[j..n]$.
  Définition similaire pour $ID_{SU}(i,k)$ et $ID_{TU}(j,k)$.
- z = score d'UN alignement multiple de S, T, U

Observation:

Score SP pour $S[i..n], T[j..n], U[k..n]$ supérieur à $ID_{ST}(i,j) + ID_{SU}(i,k) + ID_{TU}(j,k)$

Si $D(i,j,k) + ID_{ST}(i,j) + ID_{SU}(i,k) + ID_{TU}(j,k) > z$, alors (i,j,k) ne peut pas faire partie d'un chemin optimal

Aucun scrutage avant n'est nécessaire pour (i,j,k). Plus important, certaines cases ne sont jamais introduites dans la liste q.

Observation ampirique: Cette méthode peut aligner efficacement jusqu'à 6 séquences de longueur 200. Efficacité dépend beaucoup de la val. z initiale

Heuristique bornée pour le score SP

- Heuristique: Algorithme qui n'est pas garanti d'obtenir la solution optimale. Utilisé pour des problèmes difficiles (NP-complets)
- Heuristique bornée: On sait dans quel intervalle se situe la solution
- Heuristique pour le score SP: Algorithme garanti d'obtenir un alignement dont le score est au plus deux fois plus élevé que le score d'un alignement optimal.

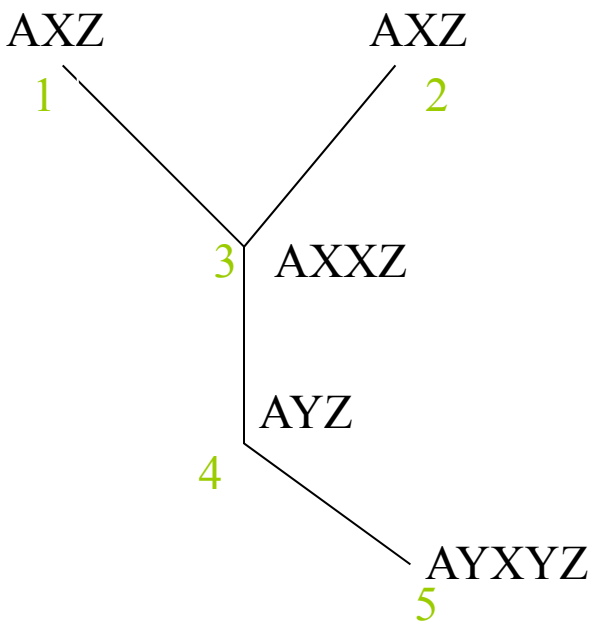
Alignement consistant avec un arbre

S: Ensemble de séquences;

T: Arbre reliant les séq. de ***S***

A: Alignement multiple de ***S***

A consistant avec ***T*** ssi: pour tout couple de séquences S_i, S_j reliées par un arc, S_i et S_j sont alignées de façon optimale dans *A*



3: A X X - Z
1: A X - - Z
2: A - X - Z
4: A Y - - Z
5: A Y X X Z

Méthode

- Choisir deux séquences qques adjacentes dans l'arbre et former un alignement optimal *A*
- Choisir une séquence non encore alignée S_i, adjacente à une séquence alignée S_j
- Aligner S_i et S_j.
- Incorporer l'alignement à *A*.
  - Si un nouvel espace a été rajouté dans S_j , rajouter un espace à chaque ligne à la colonne correspondante dans *A*

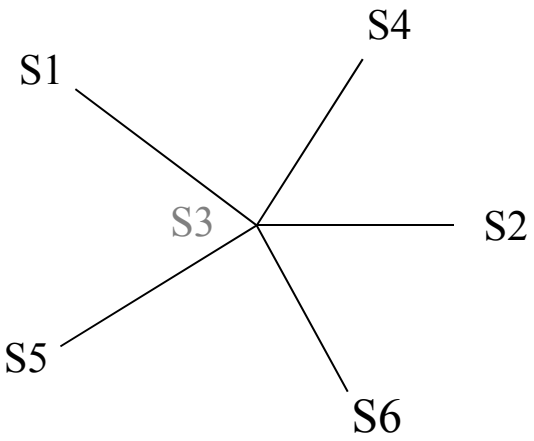
Complexité: *k* séquences de taille *n* ,

$O(kn^2)$

Arbre étoile

S: ensemble de séquences

- Séquence centrale S_c: Séquence de ***S*** tq la somme des distances à toutes les autres séquences de ***S*** est minimale.
- Arbre étoile T_c: Arbre en étoile, connectant toutes les séquences de ***S***, et de racine S_c



S={S1,S2,S3,S4,S5,S6}

Trouver un Alignement consistant avec l'arbre étoile

k = nb de séquences, n = taille de chaque séquence

Complexité:

- Trouver la séquence centrale S_c:

  $O(k^2n^2)$

- Alignement A_c consistant avec T_c:

  $O(kn^2)$

Bornes

- d(*A*): Score SP de l'alignement multiple *A*
- A_c : Alignement consistant avec l'arbre étoile
- $d_c(Si,Sj)$: Score induit par A_c pour Si, Sj
- $D(Si,Sj)$: Score d'un alignement optimal de Si et Sj
- A^*: Alignement multiple optimal de ***S***
- $d^*(Si,Sj)$: Score induit par A^*

Si le score considéré vérifie l'inégalité triangulaire:

$$e(x,z) \leq e(x,y)+e(y,z)$$

alors

$$d_c(S_i,S_j) \leq d_c(S_i,S_c) + d_c(S_c,S_j) = D(S_i,S_c)+D(S_c,S_j)$$

Et donc:

$$d(A_c)/d(A^*) \leq 2(k-1)/k < 2$$

IV. Alignement phylogénétique

Données: Un ensemble de séquences *S,* et un arbre phylogénétique *T* pour *S.* On considère la distance d'édition entre deux séquences.

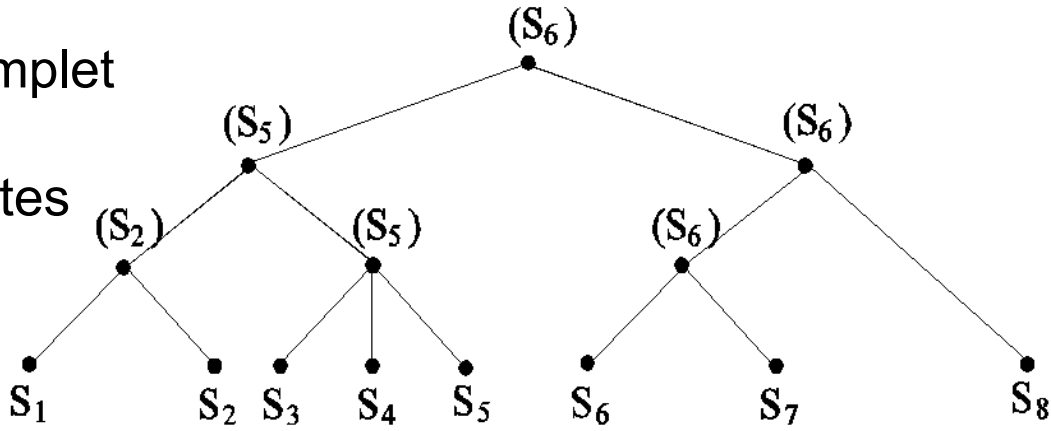
Problème: Trouver un étiquettage des nœuds internes de T qui minimise la score de *T* (somme des poids des arcs)

L'arbre *T* avec étiquettage de ses nœuds internes est appelé **alignement phylogénétique.**

Un alignement phylogénétique *T** induit un alignement de S: c'est l'alignement consistant avec T*.

Problème de l'étiquettage: NP-complet

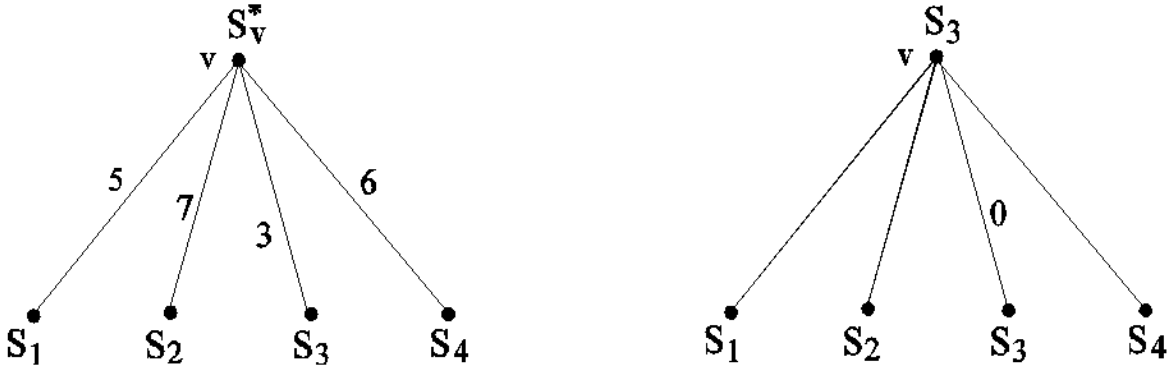
Alignement soulevé: Les étiquettes Sont des séquences de ***Sq***



Alignement soulevé optimal: borne sup pour l'al. phyl. opt.

- T*: alignement phylogénétique optimal
- On veut construire un alignement soulevé T^S à partir de T*

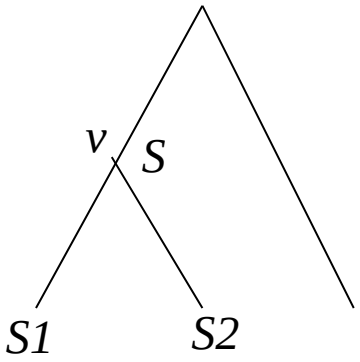
Dans T^S , v est étiquetté par la séquence de ***S*** la plus proche de S_v^*

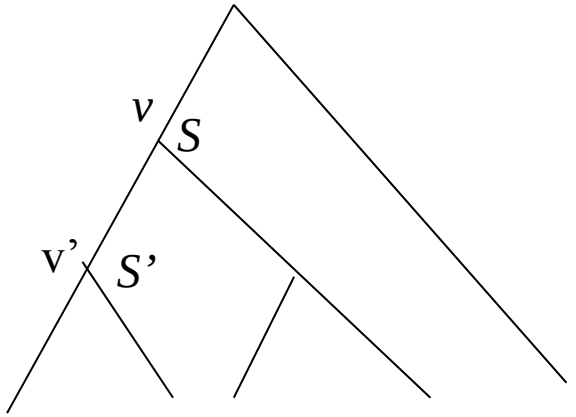


Score de $T^S \leq 2$ fois score de T*

Alignement soulevé optimal

- T_v: sous-arbre de racine *v de* T
- *d(v,S):* Score de l'al. phyl. opt. de T_v sachant que v étiquetté par *S*





$d(v,S) = D(S,S1)+D(S,S2)$ $\quad d(v,S) = \Sigma_{v'} min_{S'} [D(S,S') + d(v',S')]$

Valeur de l'al. Soulevé op. = minimum de *d(r,S)* où r racine de l'arbre

Complexité: *k* seq. de taille *n*.

Au cours d'un prétraitement, calculer tous les *D(Si,Sj)*: $O(k^2n^2)$

Pour chaque nœud v, calculer chaque d(v,S) en $O(k^2)$: $O(k^2n^2+k^3)$

V. Heuristiques usuelles
Méthodes progressives

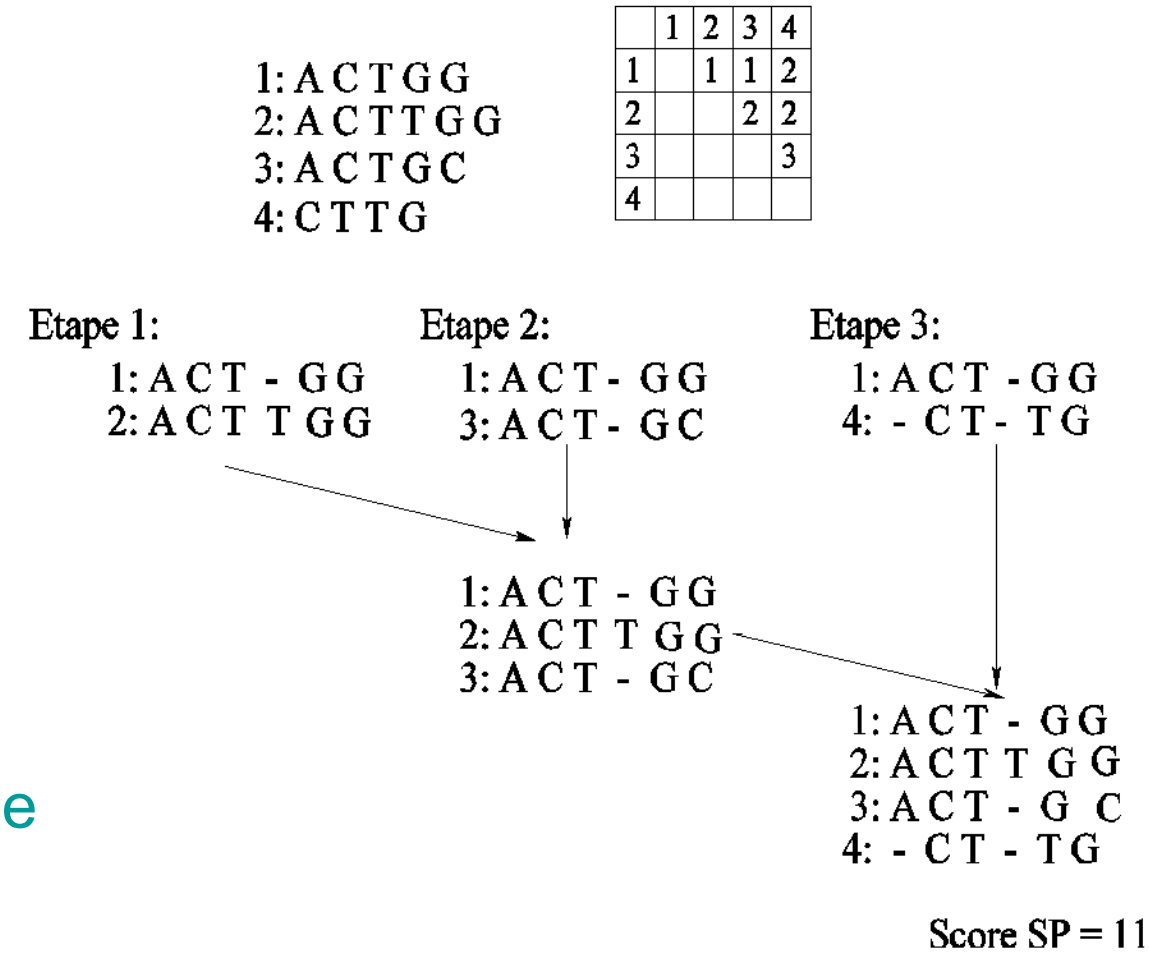
Créer un alignement multiple de ***S*** en fusionnant deux alignements de deux sous-ensembles ***S1*** et ***S2*** de ***S***

Méthode générale:

- Calculer les alignements deux à deux
- Construire un arbre guide des séquences (UPGMA, Neighbor-Joining)
- Incorporer les séquences une à une dans l'alignement multiple, en suivant l'ordre déterminé par l'arbre guide

Exemple d'alignement progressif

- Pour commencer, aligner les deux séquences de distance minimale
- À chaque étape, choisir la séquence dont la distance avec une des séquences déjà alignée est minimale



Plusieurs implémentations de la méthode progressive

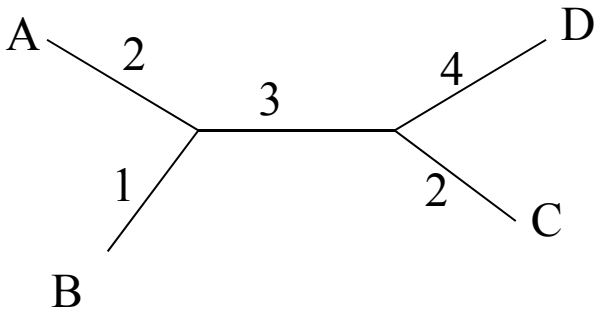
- MultAlign, ClustalW, Pileup, T-Coffee
- Diffèrent surtout par la méthode de construction de l'arbre guide
- Avantages: Rapide, simple à programmer, nécessite peu de mémoire
- Inconvénients:
  - Alignement obtenu très dépendant de l'arbre considéré
  - L'alignement ne peut pas être modifié au cours du processus
  - Produit un seul alignement

ClustalW

(Thompson, Higgins, Gibson 1994)

Algorithme progressif le plus utilisé

- Calculer les scores d'alignement de chaque paire de séquences.
- Construire un arbre guide par Neighbour-Joining
- Utiliser cet arbre pour choisir les séquences à incorporer à l'alignement. Choisir les plus petites distances à chaque fois



Effectue trois sortes d'alignements: Entre deux séquences, une séquence et une matrice consensus, ou deux matrices consensus

Scores de ClustalW

- Matrice de similarité choisie en fonction de la similarité des séquences comparées
  - 80 à 100 % identité --> Blosum80
  - 60 à 80 % identité --> Blosum60
  - 30 à 60 % identité --> Blosum45
  - 0 à 30 % identité --> Blosum30
- Scores des gaps:

  -Score d'initialisation d'un gap (SIG) + score d'extension (SEG)

  G T E A K L I V L M A N E

  G A - - - - - - - - - K L -----> SIG + 9 * GEP

  - Score des gaps dépendant des positions et des résidus supprimés (si hydrophiles, SIG plus faible)

Alignement d'une séquence avec une matrice consensus

C1	C2	C3	C4	C5
a	c	g	-	t
a	c	a	c	t
a	g	g	c	-
g	c	-	c	g

	C1	C2	C3	C4	C5
a	0.75	0	0.25	0	0
c	0	0.75	0	0.75	0
g	0.25	0.25	0.50	0	0.25
t	0	0	0	0	0.50
-	0	0	0.25	0.25	0.25

a a c - c g

C1 - C2 C3 C4 C5

Valeur d'un tel alignement?

- Matrice de pondération

	a	c	g	t	-
a	2	-3	-1	-3	-1
c	-3	2	-3	-1	-1
g	-1	-3	2	-3	-1
t	-3	-1	-3	2	-1
-	-1	-1	-1	-1	0

- Matrice consensus

	C1	C2	C3	C4	C5
a	0.75	0	0.25	0	0
c	0	0.75	0	0.75	0
g	0.25	0.25	0.50	0	0.25
t	0	0	0	0	0.50
-	0	0	0.25	0.25	0.25

- Alignement :

S: a a c – c g

C1 - C2 C3 C4 C5

p(a,C1) = 2 * 0.75 – 1 * 0.25 = 1.25

p(a,-) = -1*1 = -1 ; S(c,C2)= 2*0.75 -3*0.25 =0.75

p(-,C3) = -1 * 0.25 -1 * 0.50 + 0 * 0.25 = -0.75 …

=> Score alignement = Σ_i p(Ci,ti) =1.25 1+0.75+…= -1

Calcul d'un alignement optimal

D(i,j) : Score alignement optimal entre *S[1..i]* et *C[1..j]*

- $D(i,0) = \Sigma_{k \le i}\, p(t_k,-)$; $D(0,j) = \Sigma_{k \le j}\, p(-,C_k)$

- $D(i,j) = max\,[D(i-1,j-1)+p(t_i,C_j),\ D(i-1,j)+p(t_j,-),\ D(i,j-1)+p(-,C_j)]$

Complexité: $O(|\Sigma|\, mn)$

(*n*: nbre de colonnes de *C*; *m*: taille de *S*)

Optimisation ``itérative'' des méthodes progressives

Un problème des méthodes progressives: alignements intermédiaires ``figés''

X: GAAGTT

Y: GAC - TT 1er alignement intermédiaire

Z: GAACTG

W: GTACTG Y aurait dû être: G - ACTT

Méthode itérative

- Obtenir un premier alignement multiple de basse qualité
- Améliorer l'alignement par une suite d'itérations bien définies, jusqu'à ce que l'alignement ne puisse plus être amélioré.
- Méthodes déterministes ou stochastiques (alignement modifié au hasard)
- MultAlign, IterAlign, Praline, SAGA, HMMER…

Algorithme de Barton-Stenberg (MultAlign)

- Calculer tous les alignements deux à deux
- Choisir l'alignement de score max, une première matrice consensus
- À chaque étape,
  - choisir une paire de séquences de score max, tq exactement une des séquences est dans l'alignement partiel obtenu.
  - Aligner la nouvelle séquence avec la matrice consensus courante.
  - Mettre à jour la matrice consensus
  - Recommencer jusqu'à épuisement des séquences
- Retirer S1 et la réaligner avec la matrice consensus de l'al. restant (S2…. Sn). Recommencer avec S2,…,Sn
- Répéter le processus un nbre fixé de fois, ou jusqu'à ce que le score de l'alignement converge.

Méthode d'alignement par points d'ancrage

Basée sur la recherche de motifs (points d'ancrage, séquences consensus…).

Par exemple, MACAW:

- Rechercher un motif suffisamment long commun à une majorité de séquences
- Problème subdivisé en deux: partie gauche et partie droite par rapport au motif
- Recommencer récursivement avec chaque partie
- Les séquences ne contenant pas le motif sont alignées séparément, par score SP. Les deux sous-alignements sont ensuite fusionnés
- Lorsque les sous-séquences ne contiennent plus de bons motifs, elles sont alignées par score SP