

PLAN

- * Introduction

- * Définition
- * Signification biologique

- * Alignements à l'aide du score "sum-of-pairs"

- * Méthode exacte
- * Heuristiques
- * Accélération de la méthode exacte

- * Alignements phylogénétique

- * Méthode progressive
- * Méthode itérative
- * Méthode par points d'ancrage

Introduction à l'alignement multiple:

- Généralisation de l'alignement de 2 séquences
- **Données:** Un ensemble de séquences homologues (nucléotides ou acides aminés): S_1, S_2, \dots, S_k
- **Alignement multiple:** Matrice $A = (a_{ij}), 1 \leq i \leq k, 1 \leq j \leq L$ les a_{ij} sont des symboles de l'alphabet ou “-”, tel que la concaténation des caractères à la ligne i donne S_i

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

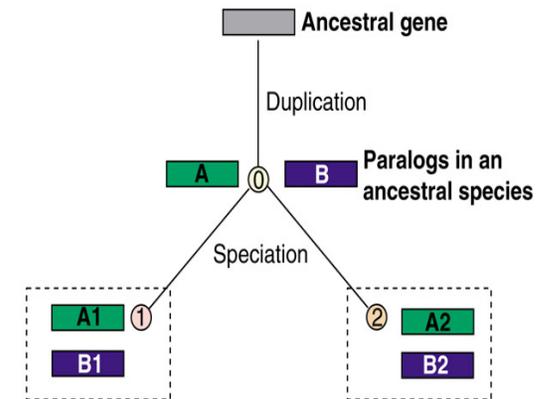
		10	20	30	40	50	
timhum.aa	1	MAPSAKFFVUG	GNWKMNGRQKQ	SLGE LIG-T	LNAAKUPAQT	EVU--CAPPT	50
timsac.aa	1	MA--RTFFUG	GNFKLNGSKQ	SIKELVER--	LNTASIPENJ	EVU--C-PPA	50
timmus.aa	1	MAPTRKFFVUG	GNWKMNGRQK	CLGE LIG-T	LNAANUPAQT	EVU--CAPPT	50
timdros.aa	1	M--SAKFDVG	GNWKMNGDQK	SIFE LIAKT	LSSALDPNT	EVUIG--PA	50
timcel.aa	1	M--TRKFFVUG	GNWKMNGDVA	SJGG IJ--TF	LNASADNSSV	DUUJ--APPA	50
		60	70	80	90	100	
timhum.aa	51	AVIDFARQKL	D-P-KIAVAA	QNCV-KUTNG	AFTGEISPST	IKDGGATWUJ	100
timsac.aa	51	TMLDVSUSLV	KKP-QUTVGA	QNAV-K-ASG	AFTGENSUDQ	IKDUKAKWVI	100
timmus.aa	51	AVIDFARQKL	D-P-KIAVAA	QNCV-KUTNG	PFTGEISPST	IKDLGATWUJ	100
timdros.aa	51	IVLMYARNLL	--PDELGLAG	QNAV-KVAKG	AFTGEISPAM	LKDIKADWVI	100
timcel.aa	51	PVLAFAKSKL	K-R-GULVAA	QNCV-KUPKG	AFTGEISPAM	IKDLGLEWVI	100
		110	120	130	140	150	
timhum.aa	101	LGHSEARRHF	GESDELISQK	VAHALAERGLG	VIACIGEKLD	EREAGITEKJ	150
timsac.aa	101	LGHSEARRSYF	HEDDKFIADK	TKFALGQGGJG	VIACIGETLE	EKKAGXTLDV	150
timmus.aa	101	LGHSEARRHF	GESDELISQK	VAHALAERGLG	VIACIGEKLD	EREAGITEKJ	150
timdros.aa	101	LGHSEARRAIF	GESDALIAEK	AEHALAERGLK	VIACIGETLE	EREAGXTNEJ	150
timcel.aa	101	LGHSEARRHF	GESDALIAEK	TVHALAERGIK	VJFCIGEKLE	EREAGHTKDV	150
		160	170	180	190	200	
timhum.aa	151	VERQLN-AVL	DN-JKDWSKJ	VJAYEPUWAI	GTGKTATPQQ	AQEVHEKLRG	200
timsac.aa	151	VERQLN-AVL	EE-JKDWITNV	VJAYEPUWAI	GTGLAATPED	AQDIHASIRK	200
timmus.aa	151	VERQLN-AVL	DN-JKDWSKJ	VJAYEPUWAI	GTGKTATPQQ	AQEVHEKLRG	200
timdros.aa	151	V-ARQMCAYV	QK-IKDWKNU	VJAYEPUWAI	GTGKTATPDQ	AQEVHASLRQ	200
timcel.aa	151	NFRQLQ-ATV	DKGUS-VENT	VJAYEPUWAI	GTGKTASGED	AQEVHEWIRA	200
		210	220	230	240	250	
timhum.aa	201	WLKSNUSDAV	AQSTRILYGG	SUTGATCKEL	ASQPDVDGFL	JGGASLKPEF	250
timsac.aa	201	FLASKLGDKR	ASELRIYGG	SANSSNAVTF	KDKADVDGFL	JGGASLKPEF	250
timmus.aa	201	WLKSNUNDGV	AQSTRILYGG	SUTGATCKEL	ATPADVDGFL	JGGASLKPEF	250
timdros.aa	201	WLSDNISKEV	SASLRIDYGG	SUTFANAKEL	AKKPDIDGFL	JGGASLKPEF	250
timcel.aa	201	FLKEKUSPAV	ADATRIYGG	SUTADNARDV	GKKPDIDGFL	JGGASLKPDF	250
		260	270	280	290	300	
timhum.aa	251	VDIIN-ARQ.	300
timsac.aa	251	VDIINS-AN.	300
timmus.aa	251	VDIIN-ARQ.	300
timdros.aa	251	LDIIN-ARQ.	300
timcel.aa	251	VKIIN-ARS.	300

Buts de l'alignement multiple:

- ❑ Trouver des **caractéristiques communes** à une famille de protéines
- ❑ Caractériser les **régions conservées** et les **régions variables**
- ❑ Dédire des **contraintes structurales** pour les ARN
- ❑ Différencier entre **gènes orthologues** et **gènes paralogues**
- ❑ Construire l'**arbre phylogénétique** des séquences homologues considérées

Différences entre orthologues et paralogues:

- ~ **Séquences homologues:** séquences qui ont évolué à partir d'un ancêtre commun.
- ~ **Orthologues:** évolution par mutations ponctuelles et spéciation.
- ~ **Paralogues:** évolution par duplication + mutations
- ~ En général, les gènes orthologues **conservent la même fonction**. Pas le cas pour les paralogues.
- ~ Les gènes orthologues ont des contraintes évolutives similaires. Pas le cas des gènes paralogues. Un alignement multiple devrait pouvoir permettre de distinguer entre paralogues et orthologues



© notes de cours de Nadia El-Mabrouk

Différentes 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 et motif conservé: Expression régulière

$$A[(((GGT|AC)(C|T)\{3,6\})|AC)]$$

Différentes représentations d'une famille de séquences:

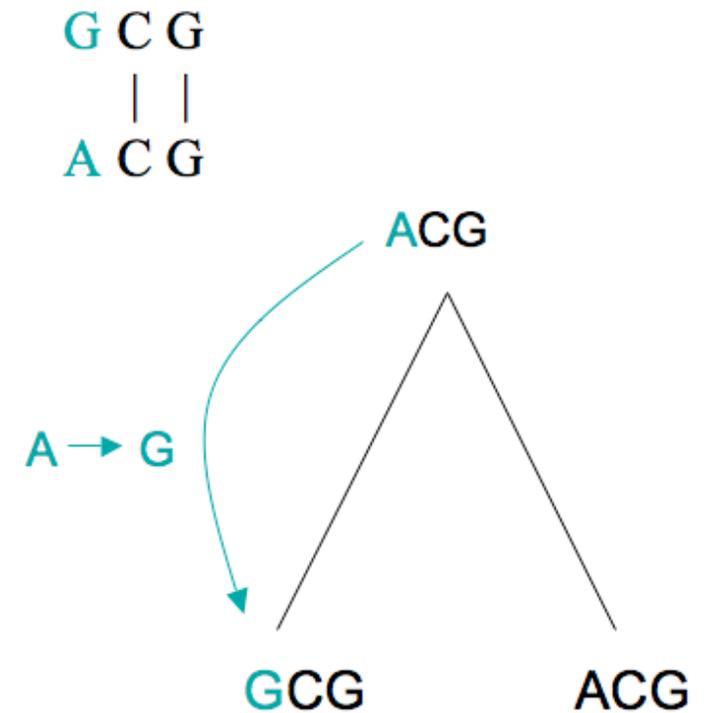
- **Matrice consensus (ou profile):** Taux d'apparition de chaque nucléotide à chaque colonne de l'alignement multiple

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

Pondération d'un alignement:

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



© notes de cours de Nadia El-Mabrouk

Score “sum of pairs” (SP):

Définition: Étant donné un alignement multiple \mathcal{A} , l'**alignement induit** pour deux séquences S_i et S_j est la restriction de l'alignement à ces deux séquences. Un appariement formé de deux “-” est ignoré.

Score SP d'un alignement \mathcal{A} : Somme des scores des alignements induits pour toutes les paires de séquences dans \mathcal{A} .

Exemple:

$$\begin{array}{r} 5 \left\{ \begin{array}{l} S_1 = A A G A A - A \\ S_2 = A T - A A T G \\ S_3 = C T G - G - G \end{array} \right. \begin{array}{l} 4 \\ 5 \end{array} \end{array} \quad SP = 4 + 5 + 5 = 14$$

SP est le score le plus utilisé: pas vraiment de justification théorique mais facile à calculer.

Alignement à l'aide du score “sum of pairs”

□ Méthode exacte:

- Trouver un alignement multiple ayant un score SP minimum
- Généralisation de l'alignement de 2 séquences
- Problème NP-complet
- Si on a m séquences de taille n l'algorithme exacte fonctionne en $O(n^m)$
- Très inefficace si $m > 5$ et $n > 100$

$$100^5 = 10000000000$$

Solution exacte pour $n=3$ (score SP)

- Soient S, T, U trois séquences de taille respective n_1, n_2 et n_3
- $D(i, j, k)$: Score SP de l'alignement optimal de $S[1..i], T[1..j]$ et $U[1..k]$
- $c(i, j), c(i, k)$ et $c(j, k)$: scores des appariements respectifs de $(s_i, t_j), (s_i, u_k)$ et (t_j, u_k)
- **Relations de récurrence:** conditions initiales
 - Si $D_{ST}(i, j)$ est le score de l'alignement optimal de $S[1..i]$ et $T[1..j]$
 $D_{SU}(i, k)$ est le score de l'alignement optimal de $S[1..i]$ et $U[1..k]$
et $D_{TU}(j, k)$ est le score de l'alignement optimal de $T[1..j]$ et $U[1..k]$

Alors:

$$D(i, j, 0) = D_{ST}(i, j) + (i + j)d$$

$$D(i, 0, k) = D_{SU}(i, k) + (i + k)d$$

$$D(0, j, k) = D_{TU}(j, k) + (j + k)d$$

Solution exacte pour $n=3$ (score SP)

- **Relations de récurrence:** Pour chaque cas (i,j,k) , examiner les 7 cases voisines

$$d_1 = D(i-1, j-1, k-1) + c(i, j) + c(i, k) + c(j, k)$$

$$d_2 = D(i-1, j-1, k) + c(i, j) + 2d$$

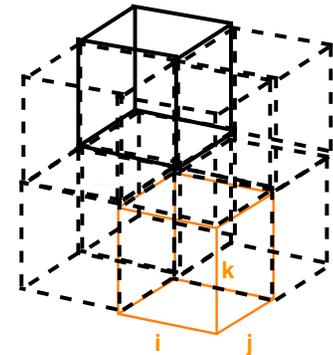
$$d_3 = D(i-1, j, k-1) + c(i, k) + 2d$$

$$d_4 = D(i, j-1, k-1) + c(j, k) + 2d$$

$$d_5 = D(i-1, j, k) + 2d$$

$$d_6 = D(i, j-1, k) + 2d$$

$$d_7 = D(i, j, k-1) + 2d$$



$$D(i, j, k) = \min\{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$$

Nombre constant de comparaisons pour chaque case: **complexité en temps de $O(n_1n_2n_3)$**

Heuristiques (score SP)

1) Alignement consistant avec un arbre:

Définition: Soit S un ensemble de séquences distinctes et \mathcal{T} un arbre dont chaque noeud est étiqueté par une séquence de S . Un alignement \mathcal{A} de S est dit **consistant avec \mathcal{T}** si, pour tout couple de séquences (S_i, S_j) reliées par un arc dans \mathcal{T} l'alignement induit pour (S_i, S_j) a un score optimal.

La notion d'alignement consistant est, bien entendu, très liée à l'arbre considéré.

Deux séquences non adjacentes dans l'arbre ont, en général, un score induit supérieur $D(S_i, S_j)$

S : un ensemble de m séquences distinctes

\mathcal{T} : un arbre dont chaque noeud est étiqueté par une séquence de S

Théorème: Un alignement \mathcal{A} de S consistant avec \mathcal{T} peut être produit en temps polynomial

Méthode:

- **Étape initiale:** Choisir 2 séquences quelconques adjacentes S_i et S_j dans l'arbre et former un alignement de score $D(S_i, S_j)$
- **Étape x:** Choisir une séquence S' non alignée, adjacente à une séquence S_i déjà alignée.

\bar{S}_i : séquence correspondant à S_i dans l'alignement (séquence avec des "-")

Former un alignement optimal entre S' et \bar{S}_i avec la règle supplémentaire qu'un appariement formé de deux "-" à un score nul

\bar{S}' : séquence correspondant à S' dans l'alignement de S' et \bar{S}_i

$O(mn^2)$

- si aucun nouvel espace n'est inséré dans \bar{S}_i , alors rajouter \bar{S}' à l'alignement déjà existant

- si un nouvel espace est ajouté à la position l dans \bar{S}_i , alors rajouter des espaces à la position l de chacune des séquences de l'alignement

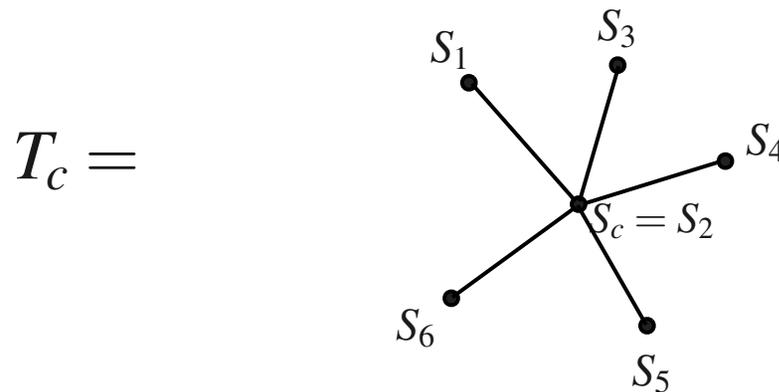
2) Alignement SP par la méthode de l'étoile centrale:

S : un ensemble de m séquences distinctes

Séquence centrale S_c : Séquence de S telle que la somme des distances de cette séquence à toutes les autres séquences de S est minimale.

Arbre étoile T_c : Arbre en étoile connectant toutes les séquences de S à la racine S_c

Exemple: Si $S = \{S_1, S_2, S_3, S_4, S_5, S_6\}$ et $S_c = S_2$ alors



S : un ensemble de m séquences distinctes

Complexité:

- Trouver la séquence centrale $S_c : O(m^2n^2)$
- Trouver un alignement A_c consistant avec $T_c : O(mn^2)$

Bornes:

- $d(A)$: score SP de l'alignement multiple A
- $d_c(S_i, S_j)$: score induit par A_c
- $D(S_i, S_j)$: score d'un alignement optimal de S_i et S_j
- A^* : alignement multiple optimal de S
- $d^*(S_i, S_j)$: alignement induit par A^*

Si le score (distance) utilisé vérifie l'**inégalité triangulaire** alors:

$$\text{Lemme: } d_c(S_i, S_j) \leq d_c(S_i, S_c) + d_c(S_c, S_j) = D(S_i, S_c) + D(S_c, S_j)$$

Preuve: L'inégalité découle de l'inégalité triangulaire et l'égalité découle du fait que l'alignement multiple considéré est consistant avec T_c

$$\text{Théorème: } d(A_c)/d(A^*) \leq 2(m-1)/m < 2$$

Preuve: Au tableau!!

Donc, le score de l'alignement obtenu est garanti d'être **inférieur** à **2 fois** le score de l'alignement **optimal**

Accélération de la méthode exacte

Programmation dynamique avec scrutage avant:

- **Pour l'alignement de 2 séquences:** Dès que $D(i,j)$ est calculé, les valeurs de $D(i,j+1)$, $D(i+1,j)$ et $D(i+1,j+1)$ sont mises à jour.
- **Graphe d'édition:**
 - Les sommets du graphe sont toutes les paires (i,j)
 - Les arêtes sont orientés et relient (i,j) à $(i+1,j)$, $(i,j+1)$ et $(i+1,j+1)$
 - Le poids $p(v,w)$ d'une arête (v,w) est le poids de l'opération qui fait passer v à w (insertion, suppression, substitution ou identité)
- Si on note $p(w)$, la valeur provisoire de $D(w)$. Après le calcul de $D(v)$, l'instruction suivante est effectuée:
$$p(w) := \min \{p(w), D(v) + p(v,w)\}$$
- La valeur de $D(w)$ est la valeur $p(w)$ obtenue après que tous les voisins de w est été considérés.

Algorithme:

$q \leftarrow (0,0)$; (q est une queue contenant les cases à considérer)

$p((0,0)) := 0$;

$p((i,j)) := \infty$;

Tant que q n'est pas vide faire

$v :=$ premier élément de q ;

supprimer v de q ;

$D(v) := p(v)$;

Rajouter $w=(i,j+1)$ en queue de q (si w pas dans q);

$p(w) := \min \{ p(w), D(v)+p(v,w) \}$

Rajouter $w=(i+1,j)$ en queue de q (si w pas dans q);

$p(w) := \min \{ p(w), D(v)+p(v,w) \}$

Rajouter $w=(i+1,j+1)$ en queue de q (si w pas dans q);

$p(w) := \min \{ p(w), D(v)+p(v,w) \}$

Fin tant que

Accélération de la méthode exacte (suite)

Avec 3 séquences S, T et U de taille n, le graphe d'édition a environ n^3 sommets et $7n^3$ arêtes

But: Éviter un certain nombre d'opérations par case et surtout, éviter de considérer toutes les cases

- ❑ $ID_{ST}(i,j)$: Score de l'alignement optimal 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 empirique: Cette méthode peut aligner efficacement jusqu'à 6 séquences de longueur 200. Efficacité dépend beaucoup de la val. z initiale

Alignement phylogénétique

© notes de cours de Nadia El-Mabrouk

Données: Arbre phylogénétique T pour un ensemble de séquences S
 D : Distance d'édition entre deux séquences

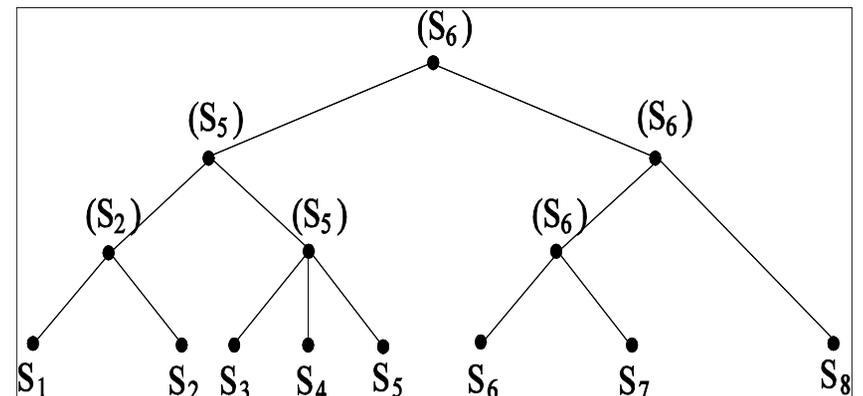
Alignement phylogénétique: étiquetage des nœuds internes de T

Problème: Trouver un étiquetage des nœuds internes qui minimise la score de l'arbre (**score de l'alignement phylogénétique**)

Alignement résultant: Alignement consistant avec T

Problème de l'étiquetage: NP-complet

Alignement soulevé: Les étiquettes sont des séquences de S

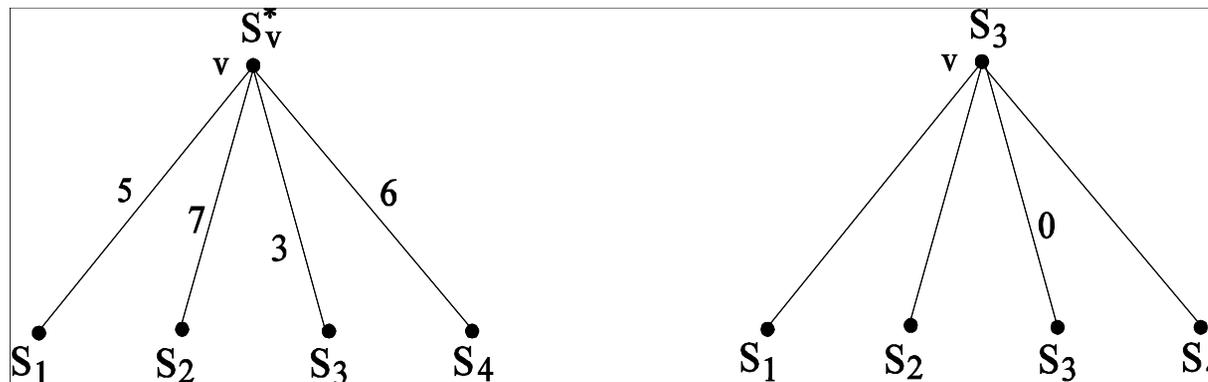


Alignement soulevé optimal

(Wang, Jiang, Lawler 1996)

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

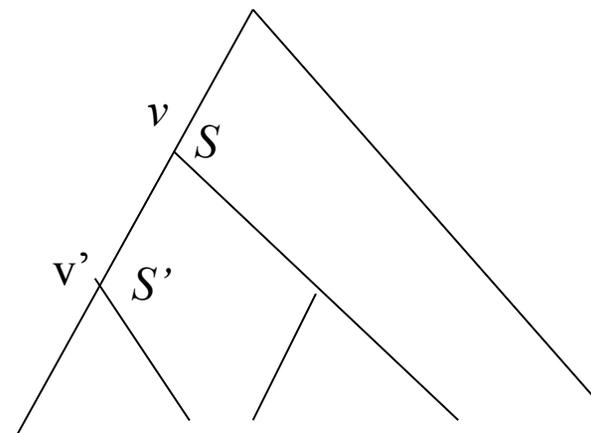
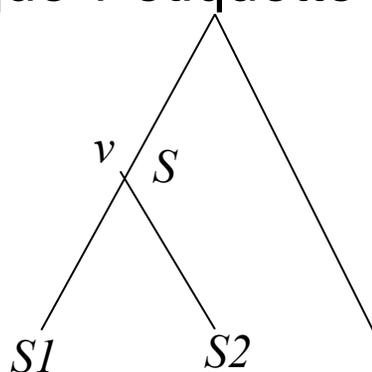
Dans T^S , v est étiqueté par la séquence de S la plus proche de S_v^*



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

Alignement soulevé optimal

- T_v : sous-arbre de racine v
- $d(v,S)$: Score de l'alignement phylogénétique optimal de T_v sachant que v est étiqueté par S



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

Valeur de l'alignement soulevé optimal = minimum de $d(r,S)$ où r racine de l'arbre.

Complexité: m séquences de taille n .

Au cours d'un prétraitement, calculer tous les $D(S_i, S_j)$: $O(m^2 n^2)$

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

Heuristiques usuelles

Méthodes progressives

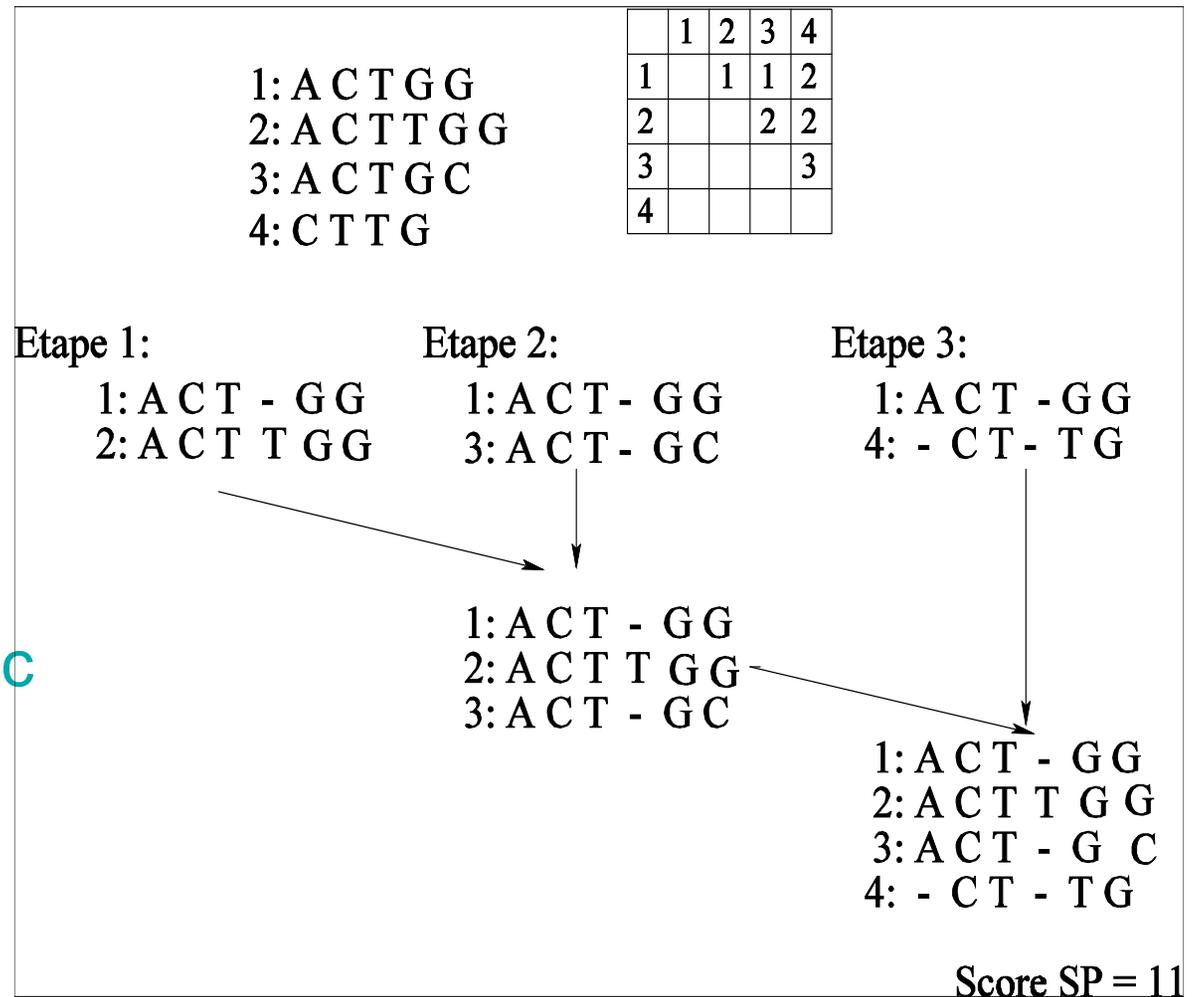
Créer un alignement multiple de S en fusionnant deux alignements de deux sous-ensembles $S1$ et $S2$ de S

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

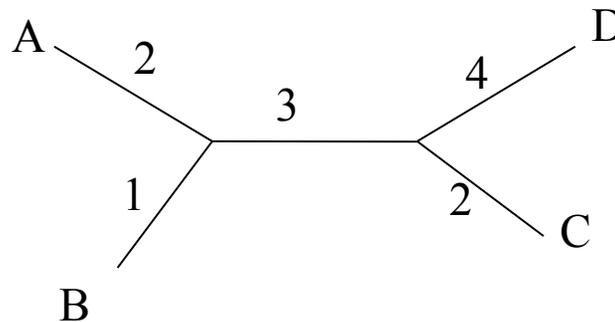
- MultAlign, ClustalW, Pileup, T-Coffee
- Diffèrent surtout par la méthode de construction de l'arbre guide
- Avantages: Rapide, simple à programmer, nécessite peu de mémoire
- Inconvénients:
 - Alignement obtenu très dépendant de l'arbre considéré
 - L'alignement ne peut pas être modifié au cours du 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 --> Blosom80
- 60 à 80 % identité --> Blosom60
- 30 à 60 % identité --> Blosom45
- 0 à 30 % identité --> Blosom30

□ **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 + 8 * SEG

- 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 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

□ 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

$$p(a,C1) = 2 * 0.75 - 1 * 0.25 = 1.25$$

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

$$p(-,C3) = -1 * 0.25 - 1 * 0.50 + 0 * 0.25 = -0.75 \dots$$

□ Alignement :

S: a a c - c g
 C1 - C2 C3 C4 C5

$$\Rightarrow \text{Score alignement} = \sum_{i,j} p(C_j, t_i) = 1.25 + 1 + 0.75 + \dots = -1$$

Calcul d'un alignement optimal

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

$$\square D(i,0) = \sum_{k \leq i} p(t_k, -) \quad ; \quad D(0,j) = \sum_{k \leq j} p(-, C_k)$$

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

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 S_1 et la réaligner avec la matrice consensus de l'al. Restant ($S_2 \dots S_n$). Recommencer avec S_2, \dots, S_n
- ❑ 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