M2 - STL

Algorithmes sur les séquences en bioinformatique

Cours 3: Heuristiques sur les alignements de séquences, matrices de substitution et alignement multiple

Alessandra Carbone
Université Pierre et Marie Curie

Introduction au problème

Quand on cherche une séquence qui soit proche à une séquence donnée de longueur 200-500 bases dans une bdd de taille 10^9–10^{10}, nous ne pouvons pas utiliser les algorithmes d'alignement comme ceux qui nous avons présenté jusqu'ici parce qu'ils demandent trop de temps. Plusieurs approches pour résoudre ce problème ont été proposées :

1. **Implémentation en hardware des algorithmes** de programmation dynamique. Cette méthode est très chère et donc pas choisie dans la pluspart des cas.

2. Utilisation du **parallélisme au niveau de hardware**; le problème peut être distribué de façon efficace à plusieurs processeurs. Cette approche est aussi très chère.

3. **Utilisation de heuristiques** plus rapides que l'algorithme de programmation dynamique originale.

Une méthode heuristique est un algorithme qui donne seulement des solutions approximées au problème.

Les méthodes que l'on présentera sont basées sur les observations suivantes:

- même la complexité en temps linéaire est problématique quand la taille de la bdd est grande (plus que 10^9)

- un preprocessing des données est souhaitable pour réduire l'espace de recherche

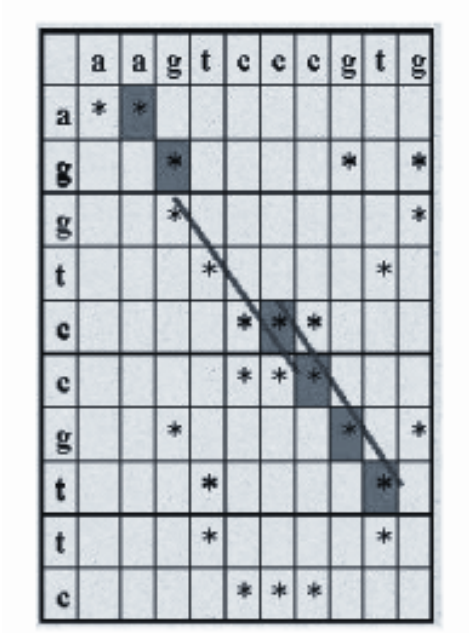
- les substitutions sont plus fréquentes que les introductions d'indels

- on s'attende que les séquences homologues contiennent plein de **segments** ayant des matches ou substitutions, mais **sans espaces/gaps.**
Ces segments peuvent être utilisés comme points de départ pour la recherche.

Algorithme FASTA

Algorithme pour la comparaison de séquences: une séquence requête est comparée avec toute les chaînes de caractères dans la bdd.

Un bon alignement locale permet la détermination de sous-séquences a matching exacte: l'algorithme utilise cette propriété et s'intéresse alors aux segments qui sont identiques entre paires de séquence. Il détermine ces régions, par exemple, avec un alignement dot-plot:



L'algorithme FASTA

1. Recherche des hot-spot, cad des plus larges sous-séquences communes

Cette étape est réalisée de façons efficace en utilisant une look-up table ou un hash. Par exemple, les données peuvent être organisées par rapport à la longueur des sous-chaînes qui matchent. Typiquement, il est recommandé d'utiliser des sous-chaînes d'ADN d'une longueur de 4-6 nucléotides et de protéines, de 1-2 acides-aminés (aa). Noter que pour aa on aura 20^2 chaînes et pour les nucléotides 4^6.

2. Sélectionner les 10 meilleurs matchings diagonales

Pour évaluer les matching de hot-spots sur les diagonales dans le diagramme dot-plot, FASTA donne des valeurs positives à chaque hot-spot et des valeurs négatives, qui diminuent avec la distance, à chaque espace entre hot-spot. La valeur d'un matching diagonale est la somme des valeurs des hot-spots et des valeurs entre hot-spots sur la diagonale.

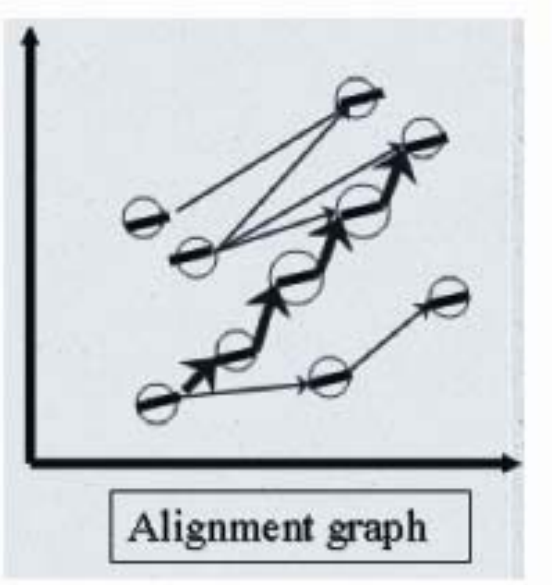
3. Calcule des meilleures scores et filtration des matching diagonales

Un matching sur la diagonale ne contient pas d'espaces parce qu'il est déduit de la diagonale. Les matching diagonales sont évalués en utilisant une matrice de substitution, et la meilleure valeur de matching diagonale est trouvée.

4. Combinaison des bons matching diagonales avec les indels

On sélectionne les matching avec des valeurs supérieures à une borne donnée et on essaye de les recombiner dans un seul alignement avec indels mais avec valeur plus forte aussi. Pour faire cela on construit un graphe dirigé pondéré tel que les nœud correspondent aux sous-alignements trouvés aux étapes précédentes, et les **poids de chaque nœud** à la valeur (calculée aux étapes précédentes) du sous-alignement associé. Les arêtes vont d'un nœud u à un nœud v si l'alignement v commence et termine à une ligne plus élevée que la ligne de terminaison de u. On donne une **valeur négative à l'arête** dépendamment du nombre d'indels qu'il faut ajouter pour aligner proprement u et v. On peut éliminer les arêtes trop chères.

Essentiellement, FASTA trouve **un chemin de poids maximale dans ce graphe pondéré**. L'alignement sélectionné spécifie un alignement spécifique entre les deux séquences. On considère à ce point seulement les alignement ayant une valeur supérieure à une certaine borne.



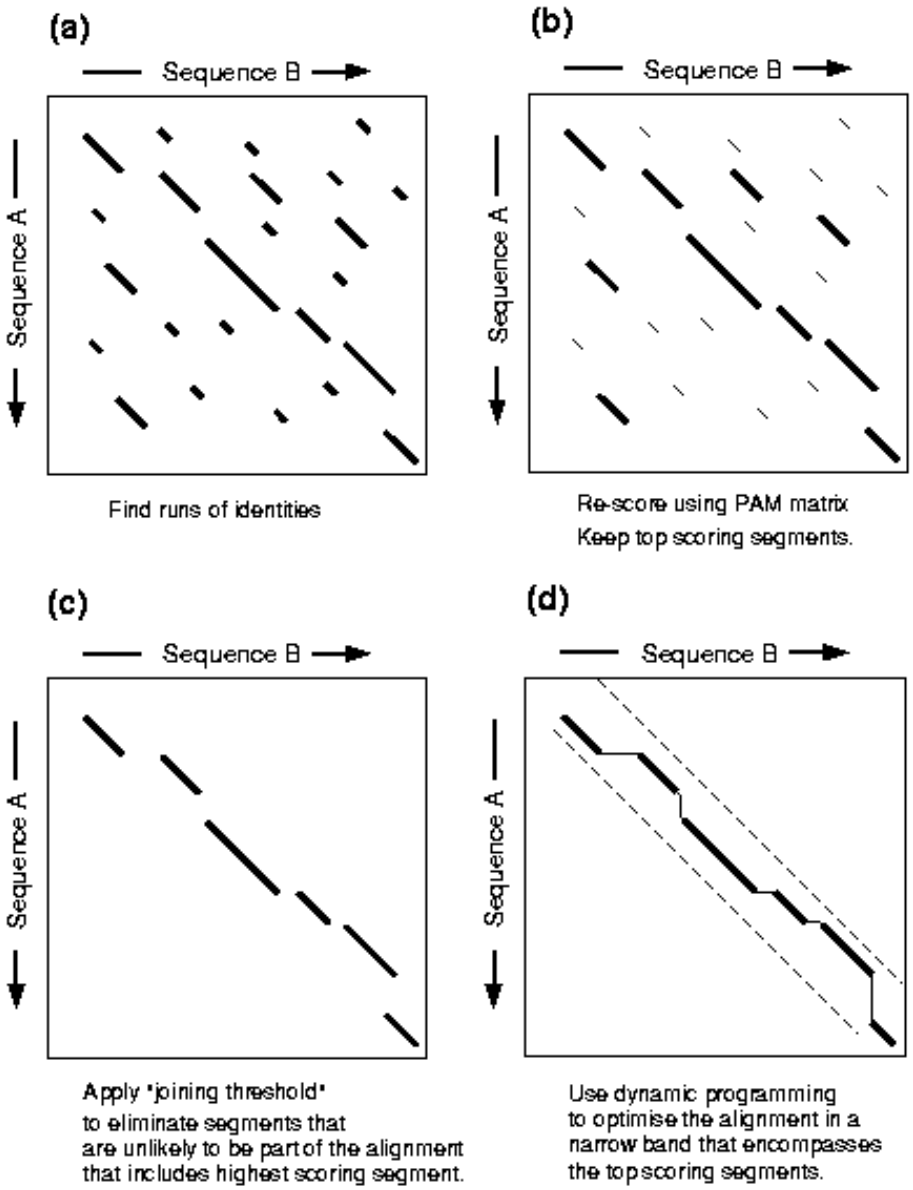
5. Calcul d'un alignement locale alternatif

Un autre alignement locale est calculé en considérant une fine bande autour de la diagonale sélectionnée à l'étape 3. En fait, il est fortement probable que le meilleur alignement locale incluant cette diagonale se trouve dans cette fine bande.
On calcule pour cette raison l'alignement locale dans cette bande, en utilisant l'algorithme de programmation dynamique usuel mais contrainte a cette bande.
L'algorithme d'alignement locale essentiellement fusionne les diagonales trouvée aux étapes précédentes pour donner un alignement qui contient des indels.
L'épaisseur de la bande est fonction de la longueur des sous-chaînes considérées a l'étape 1.

6. Ordonnancement des résultats sur les séquences de la bdd

Les séquences de la bdd sont ordonnées par rapport aux meilleurs alignements obtenus aux étapes 4 et 5. L'algorithme de programmation dynamique est utilisé pour aligner la séquence requête contre toutes les séquences résultantes de la sélection et ayant valeurs les meilleurs.

FASTA Algorithm



BLAST – Basic Local Alignment Search Tool

Avec BLAST, on voulait améliorer la vitesse de FASTA en recherchant un plus petit nombre de hot-spots optimaux. **L'idée a été d'intégrer la matrice de substitution dès la première étape de sélection des hot-spots**.

BLAST a été développé pour les séquences de protéines, et FASTA pour des séquences d'ADN.

BLAST recherche des régions à forte similarité dans les alignements sans espaces, où l'évaluation est faite par une matrice de poids.

Terminologie

Soit S_1 et S_2 deux séquences. Une paire-segment est une paire de sous-chaînes de S_1 et S_2, ayant la même longueur, alignée sans espaces.

Un segment localement maximale est un segment dont la valeur de l'alignement sans espaces ne peut pas être améliorée par extension ou raccourcissement.

Une paire-segment maximale (PSM) dans S_1 et S_2 est une paire-segment avec une valeur maximale par rapport a toutes paires-segment dans S_1 et S_2.

Pendant la comparaison avec les séquences dans une bdd, BLAST recherche toutes séquences qui apparient avec la séquence requête avec un PSM plus grand d'un certain score S. Ces paires on les appelle paires à score élevé (PSE).
La borne S est choisie de telle façon qu'il soit improbable de trouver aléatoirement une séquence dans la bdd qui rejoint un score plus élevé que S quand on la compare avec la séquence requête.

Etape 1

Etant donnée une longueur w et une borne S, BLAST trouve toutes les séquences dans la bdd que alignent avec des sous-chaînes de la séquence requête de longueur w et score plus fort que S. Tout hot-spot s'appelle « hit » dans BLAST.
A la place de demander que une sous-chaîne aligne exactement, BLAST utilise une matrice de substitution et demande que le score S soit obtenu en utilisant la matrice.

Cette stratégie permet de maintenir un w grand (cela a un impact sur la rapidité d'exécution de l'algorithme) sans sacrifier la sensibilité. Le paramètre w est d'habitude fixé a 3-5 aa ou a ~12 nucléotides.

S devient alors le paramètre critique pour l'efficacité et la sensibilité, et w est d'habitude jamais changé: si S croit, le nombre de hits décrémente et le programme sera plus rapide; si S décrois, on permettra la détection de relations entre mots plus distantes.

Etape 2

Etendre chaque hit a un segment localement maximale et tester si son score est plus grand que S, cad si la paire de séquences est PSE. Comme les matrices de substitution typiquement contiennent des valeurs négatives, l'extension de la sous-chaîne de longueur w initiale peut diminuer le score. On peut penser alors de terminer l'extension d'un hit quand la réduction de la valeur rejoint une borne de drop-off donnée.

Remarque sur l'implémentation

La première étape pourrait être implémentée en construisant, pour chaque chaîne α de longueur w toutes les chaînes de longueur w de similarité au moins S à α. Ces chaînes peuvent être mémorisées dans une base que sera accédée par l'algorithme pendant le test des séquences de la bdd.

Une amélioration obtenue en réduisant sensiblement le nombre d'expansions a été conçue. Elle amène a un algorithme qui est 3 fois plus rapide que l'algorithme décrit ci-dessus.

Query Results

```
BLASTP 1.4.7MP [15-Oct-94] [Build 23:28:23 Oct 21 1994]

Reference:  Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers,
and David J. Lipman (1990).  Basic local alignment search tool.  J. Mol. Biol
215:403-10.

Query=
        (431 letters)

Database:  owl26_0.fasta
           105,990 sequences; 32,636,414 total letters.
Searching.....................................................done


                                                                  Smallest
                                                                    Sum
                                                            High  Probabili
Sequences producing High-scoring Segment Pairs:             Score  P(N)

 UL78_HCMVA HYPOTHETICAL PROTEIN UL78. - HUMAN CYTOMEGALO ..  2193  3.3e-305
 NG1_DROME NEW-GLUE PROTEIN 1 PRECURSOR (NG-1) (SALIVARY  ..    61  0.18
 NPYR_DROME NEUROPEPTIDE Y RECEPTOR (NPY-R) (PR4 RECEPTOR) ..  52  0.23
 NG2_DROME NEW-GLUE PROTEIN 2 PRECURSOR (NG-2). - DROSOPH ..    62  0.28
 MFU20578 MFU20578 endothelin D receptor; NCBI gi: 683729 ..    47  0.42
 SSR5_RAT SOMATOSTATIN RECEPTOR TYPE 5 (SOMATOSTATIN RECE ..    52  0.67
 HH5XAHC10 HH6XAHC positional homolog of HCMV UL78; YCEI  ..    66  0.75
 POLG_LANVT GENOME POLYPROTEIN (CONTAINS: CAPSID PROTEIN  ..    44  0.86
 OXYR_PIG OXYTOCIN RECEPTOR (OT-R). - SUS SCROFA (PIG).         65  0.87
 A55597 oxytocin receptor - rat                                 65  0.87
```

Un fort score, ou préférablement, clusters de forts scores, au début de la liste, indiquent une relation possible avec la séquence requête.

Une basse probabilité, indique qu'il n'est pas possible que le match soit été généré par chance.

Scores bas avec fortes probabilités suggèrent que le match a été généré par chance.

Amélioration de BLAST

BLAST peut être étendu pour obtenir un alignement avec espaces:

Etape 1

Quand on considère la matrice d'alignement de deux séquences dans l'algorithme de programmation dynamique, on cherche sur les diagonales 2 sous-chaînes de longueur w tels que la distance entre elles est $\leq$ A et leur valeur est $\geq$ S. S peut prendre une valeur plus petite que dans l'algorithme originale. L'expansion est réalisée seulement a partir de cette paire de sous-chaînes.

Etape 2

On réalise des alignements locaux avec espaces comme pour FASTA. On permet la fusion de **deux** alignements locaux provenant de différent diagonales pour constituer un nouveau alignement locale composé par le premier alignement locale suivi par des indels, suivi par le deuxième alignement locale. Cet alignement locale est essentiellement un chemin dans la matrice de programmation dynamique, composé par deux sections diagonales et un chemin qui les joint et qui peut contenir des espaces.

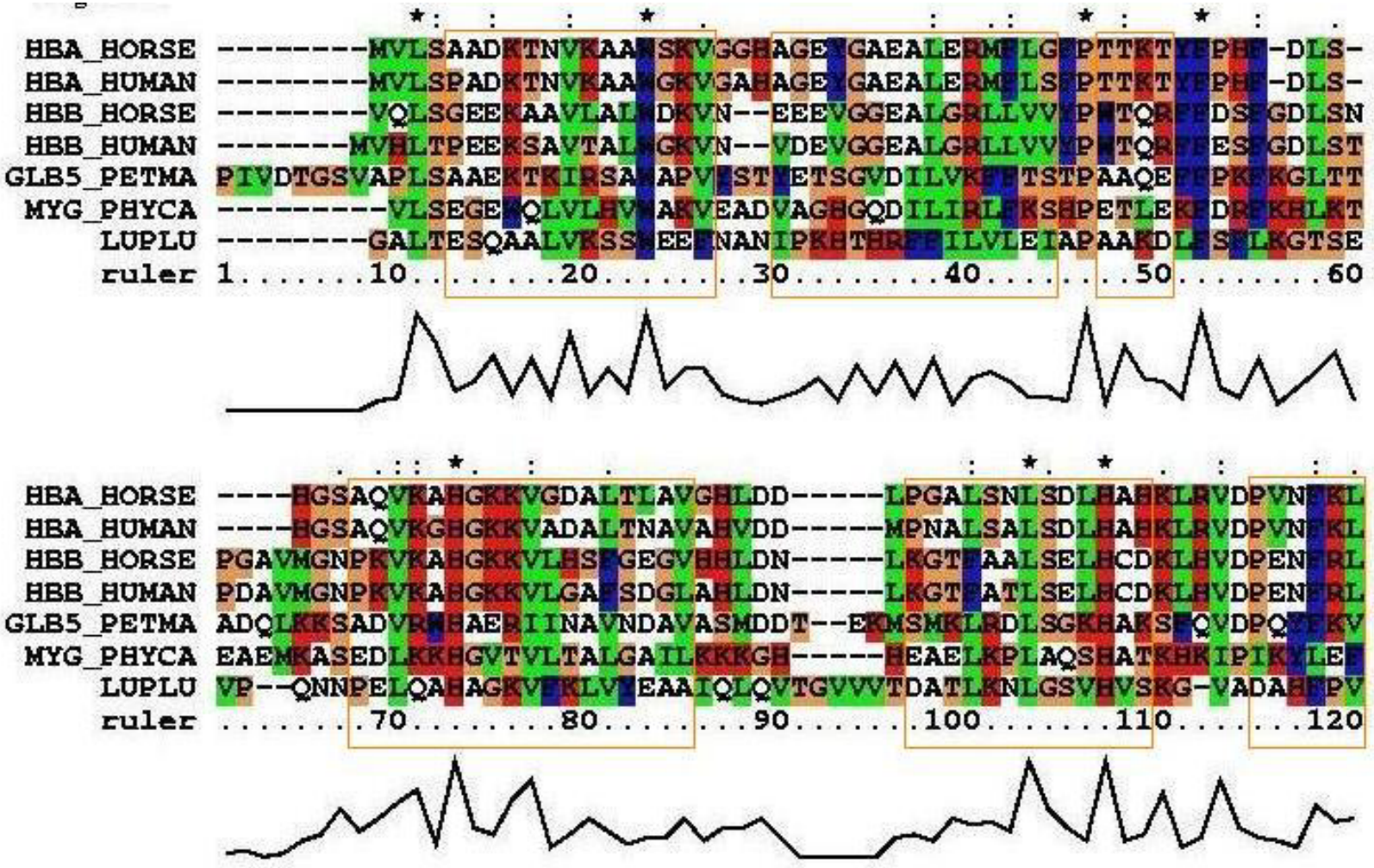
Ici, on permet un alignement locale entre différent diagonales et pas, comme en FASTA, entre diagonales proches.

PSI-BLAST : Position Specific Iterated BLAST

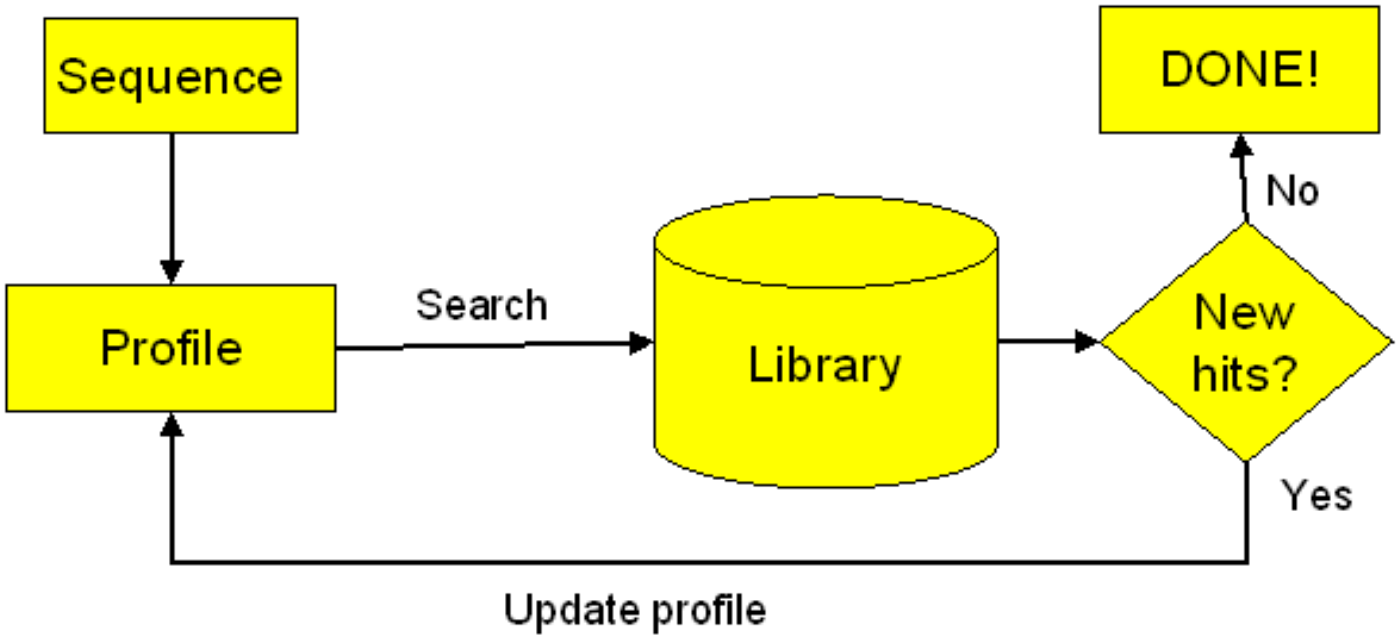
Position #	1	3	4	5	6	8		12
Profile	D	x	D	G	D/N	G	x	I x x x E

POSITION #	1	3	4	5	6	8				12
CALM_HUMAN_1	D	K	D	G	D	G	T	I	T	T K E
CALF_NAEGR_1	D	K	D	G	D	G	T	I	T	T S E
CALM_SCHPO_1	D	R	D	Q	D	G	N	I	T	S N E
CALM_HUMAN_2	D	A	D	G	N	G	T	I	D	F P E
CALF_NAEGR_2	D	A	D	G	N	G	T	I	D	F T E
CALM_SCHPO_2	D	A	D	G	N	G	T	I	D	F T E
CALM_HUMAN_3	D	K	D	G	N	G	Y	I	S	A A E
CALF_NAEGR_3	D	K	D	G	N	G	F	I	S	A Q E
CALM_SCHPO_3	D	K	D	G	N	G	Y	I	T	V E E
CALM_HUMAN_4	D	I		D	G	D	G	Q	V	N Y E E
CALF_NAEGR_4	D	I		D	G	D	N	Q	I	N Y T E
CALM_SCHPO_4	D	T		D	G	D	G	V	I	N Y E E

Eléments du profile: vecteurs de caractères de position spécifique dans l'alignement multiple


HBA_HORSE
HBA_HUMAN
HBB_HORSE
HBB_HUMAN
GLB5_PETMA
MYG_PHYCA
LUPLU
ruler
1.....10.....20.....30.....40.....50.....60
HBA_HORSE
HBA_HUMAN
HBB_HORSE
HBB_HUMAN
GLB5_PETMA
MYG_PHYCA
LUPLU
ruler
.....70.....80.....90.....100.....110.....120

SCHEMA D'ITERATION DE PSI-BLAST



Pour chaque profile nous pouvons calculer l'histogramme des types de caractères qui illustre leur distribution. Quand on aligne des séquences d'acides-aminés appartenant à la même famille de protéines, on obtient que certaines régions ont des profiles montrant des petites variations. Il s'agit de régions conservées qui définissent la structure et la fonctionnalité typique de la famille. Dans ce cas on veut que la matrice de substitution tient compte de l'information statistique relevée sur la conservation dans les colonnes, à fin d'améliorer le score d'alignement.

Dans la première étape de l'algorithme, on utilise BLAST avec un coût de vecteur V_i différent pour chaque colonne i. Au départ, chacun des vecteurs V_i est fixé à la valeur de la matrice de substitution associée au i-ème caractère de la séquence requête.

Les résultats a score fort que l'on obtient, nous permettrons de déterminer les profiles pour chaque colonne. On utilise BLAST de nouveau itérativement en utilisant comme requête les profiles, cad que a chaque colonne on utilise un histogramme de caractères plutôt que un caractère, et on le compare a la bdd. A chaque étape itérative il faut recalculer les profiles par rapport aux nouvelles séquences résultat. L'itération termine quand on ne trouve plus de nouvelles séquences qui s'apparient.

PSI-BLAST permet de trouver des séquences qui sont plus distantes que celles trouvées par BLAST ou FASTA.

Position Specific Score Matrix (PSSM)

SEQUENCE

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	V	-2	-4	-5	-5	-2	-4	-4	-5	-5	4	3	-4	0	-2	-4	-4	-2	-4	-2	4
2	G	-1	-1	1	1	-4	-1	0	5	-1	-5	-5	0	-2	-4	-3	0	0	-4	-4	-4
3	S	-2	2	0	1	-5	2	4	-2	0	-4	-4	0	-2	-4	-1	0	-1	-5	-2	-1
4	Q	0	1	1	-1	-4	0	0	0	-1	-2	-1	2	-1	0	-1	0	-2	2	4	-2
5	G	0	-3	-2	-2	-4	-3	-3	4	-4	-4	-4	-2	-4	-5	5	-1	0	-4	-2	-1
6	S	1	0	1	0	-2	-1	0	0	3	0	-2	0	-2	0	-2	2	0	-4	-1	-1
7	Q	-1	0	3	0	-4	3	3									0	0	-4	-2	-2
8	L	-3	-4	-5	-6	-2	-4	-5									-4	-3	-3	-1	0
9	S	0	-3	0	-2	-3	-2	-2									7	0	-5	-4	-4
10	R	-1	-3	-2	-2	-2	-1	-2									-1	-3	-1	-2	0
11	G	-1	-4	-2	-3	-5	-4	-2	7	-4	-6	-6	-4	-5	-5	-4	-1	-4	-4	-5	-5
12	Q	-3	-1	0	-1	-5	6	4	-4	-1	-4	-3	-1	4	-4	-3	-2	-3	2	-3	-2
13	K	-1	5	-2	-3	0	4	-1	-3	-2	-3	-2	4	-1	-3	-3	-2	-2	0	-3	-4
14	Q	-2	2	0	-2	-5	7	0	-3	-2	-5	-3	1	0	-5	-1	-1	-2	-4	-3	-4
15	R	-2	7	-2	-4	-3	-1	-2	-3	-2	-3	0	2	0	-3	-4	-3	-3	-5	-4	-3
16	I	0	-4	-5	-5	-3	-4	-4	-5	-5	3	3	-4	0	-2	-3	-3	-1	-5	-3	4
17	A	4	-2	-1	-2	1	-1	0	0	-3	-1	0	-1	1	-2	-3	1	0	0	-3	-1
18	I	-3	-4	-5	-5	-3	-4	-5	-6	-5	6	3	-4	1	1	-5	-4	-3	-4	-2	1
19	A	6	-2	-4	-4	0	-3	-3	0	-4	-1	-2	-3	-3	-4	-3	-1	-1	-5	-4	-1
20	R	-1	7	-2	-4	0	0	-1	-2	-2	-2	-2	1	1	-4	-4	-1	-3	-5	-4	-1

La glutamine prend des valeurs différentes dans ces deux positions

Matrices de substitution d'acides-aminés

Quand on recherche le meilleur alignement entre deux séquences, la matrice de substitution que l'on utilise peut affecter fortement les résultats. Idéalement, la matrice devrais représenter les phénomènes biologiques que l'alignement cherche a mettre en évidence. Par exemple dans le cas d'une divergence des séquences due a des mutations évolutionnaires, la matrice utilisée devrait être déduites, en principe, de façon empirique a partir de séquences ancestrales et des leurs descendants plus récents.

Exemple 1 : la matrice unité définie par $M_{ij} = \begin{cases} 1 & \text{si } i=j \\ 0 & \text{si non} \end{cases}$

Elle mesure la similarité

Exemple 2 : la matrice du code génétique. M_{ij} corresponde au nombre de substitutions minimales des bases qui est nécessaire pour convertir un codon de l'acide-aminé i dans un codon de l'acide–aminé j. Toute information sur les caractéristiques physico-chimiques des acides- amines (comme hydrophobicité, taille et charge) sont négligées.

Elle mesure la distance

PAM (Point Accepted Mutation): unités et matrices

Analyse de l'évolution des acides-aminés faite sur:

1572 mutations (acceptées) occurrent dans
71 superfamilles de séquences de protéines fortement proches

Les substitutions ne sont pas aléatoires! En particulier, il y a des substitutions d'acides-aminés qui occurrent plus fréquemment que d'autres, probablement parce qu'elles n'ont pas des effets importants sur la structure et la fonction de la protéine.

Conclusion: les protéines qui sont proches par des liens d'évolution n'ont pas besoin d'avoir les mêmes acides-aminés à chaque position. Elles peuvent avoir des acides-aminés comparables.

Notation: mutation « acceptée » : mutation qui a eu lieu et a été passée à sa progénie.

Unités PAM

Deux séquences sont a une distance évolutionnaire d' 1 unité PAM si S_1 a été convertie dans S_2 avec une moyenne de 1 événement de mutation ponctuelle acceptée per 100 acides-aminés. Dans ce cas soit la mutation n'a pas changé la fonction de la protéine soit le changement a été bénéficiale.

Note: deux séquences qui ont divergé à 1 unité PAM, pas forcement diffèrent de 1%! Cela est du au fait que une position peut avoir subit plusieurs mutations individuelle.

Problèmes avec la notion de unité PAM :

1. La probabilité de mutation n'est pas uniforme. Nous ne connaissons pas aujourd'hui des séquences de protéines qui sont dérivées l'une de l'autre, et la manque de protéines ancestrales est surmontée en faisant l'hypothèse que les mutations sont réversibles et de même probabilité dans les deux directions. Cette hypothèse, avec la propriété additive des unités PAM qui suit de leur définition, implique que étant donnée deux séquences S_i et S_j ayant un ancêtre en commun S_{ij}

$$d(S_i,S_j)=d(S_i,S_{ij})+d(S_{ij},S_j)$$

ou d(i,j) est la distance PAM entre les séquences i et j.

2. Insertion et délétion qui peuvent apparaître pendant l'évolution sont ignorées: pas d'évaluation du coût du gap. Donc on ne peut pas être surs de la correspondance entre positions dans les séquences. L'identification des gaps ne peut pas toujours être effectuée avec certitude quand deux séquences sont distantes.

3. L'échantillon est biaisé. Examen d'un petit nombre de séquences très proches.

Vitesse d'évolution

Famille	Vitesse (PAM/100My)
Region C de la chaine k des IG	37
Serum albumine	19
Hemoglobine	12
Trypsine	5.9
Cytochrome b	4.5
Region centrale du collagene	1.7
Histone H3	0.14
Ubiquitine	0

Matrices PAM

Chaque matrice PAM est conçue pour comparer deux séquences qui ont k unités de distance. Par exemple, la PAM120 est conçue pour comparer des séquences qui ont une distance de 120 unités, et pour chaque paire d'acides-aminés (A_i, A_j), la matrice reflète la fréquence a laquelle on attend que A_i sera remplace par A_j dans les deux séquences.

La matrice PAM-1 a été calculée en considérant 71 familles de protéines pour un totale de 1572 échanges entre acides-aminés (rappelez vous que 1 unité PAM entre deux séquences signifie grosso modo que la divergence est a peu près de 1%):

M_{ij} fréquence observée de A_i qui mute dans A_j pendant 1 unité PAM
M matrice 20x20 ou les valeurs de lignes et colonnes ont somme =1

Il y a une variance importante entre les valeurs des lignes et des colonnes de PAM1. Par exemple:

	A	R	N	D	C
A	9867	2	9	10	3
R	1	9913	1	0	1
N	4	1	9822	36	0
D	6	0	42	9859	0
C	1	1	0	0	9973

Quelques entrées de PAM1 où on a écrit 10^4 M_{ij}

PAM-0 a des 1 sur les diagonales et 0 partout ailleurs.

PAM-N est obtenue en multipliant PAM1 avec elle-même N fois.

Pour des raisons pratiques on peut dériver de PAM-N la matrice suivante:

$$c_{ij} = \log \frac{f(i)M^n(i,j)}{f(i)f(j)} = \log \frac{M^n(i,j)}{f(j)}$$

ou f(i) et f(j) sont des fréquences observées d'acides-aminés A_i et A_j. Chaque entrée représente le résultat de l'évolution ($f(i)M^n(i,j)$, cad le changement observé) vis-à-vis d'une paire aléatoire ($f(i)f(j)=f(i,j)$, cad la chance de trouver une paire ij au même endroit dans deux séquences aléatoires). L'emploi de cette matrice a la place de la PAM-N permet de calculer les scores de substitution entre paires de séquences comme des sommes de log plutôt que comme des multiplications.

Attention: La définition de PAM-N est basée sur l'hypothèse que les fréquences d'acides-aminés restent constantes dans le temps et que le processus de mutation ne change pas sur des intervalles de N unités PAM.

Matrice PAM2

ORIGINAL AMINO ACID

		A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
A	Ala	9730	0	31	24	5	34	37	42	5	3	5	18	19	5	54	99	45	0	0	32
R	Arg	0	9881	5	0	0	13	0	0	17	0	0	23	18	2	0	1	0	0	0	0
N	Asn	14	7	9701	36	0	20	7	10	24	4	2	19	1	0	10	51	17	0	0	4
D	Asp	13	0	45	9757	0	27	96	8	6	0	2	8	1	0	1	26	2	0	0	4
C	Cys	1	0	0	0	9928	0	0	1	0	2	0	0	11	0	0	12	3	0	0	6
Q	Gln	12	14	15	16	0	9736	24	4	14	4	2	9	11	0	11	13	10	0	0	5
E	Glu	21	0	9	95	0	40	9726	13	4	4	4	13	1	0	17	15	12	0	0	7
G	Gly	40	0	22	13	3	11	22	9870	1	0	2	5	0	0	17	42	8	0	0	7
H	His	2	19	20	4	0	15	3	0	9865	4	3	6	0	3	0	10	5	11	4	1
I	Ile	1	0	3	0	3	4	3	0	4	9703	22	4	22	14	2	3	14	0	0	70
L	Leu	4	0	3	3	0	4	7	2	6	52	9899	6	99	19	0	5	7	0	0	24
K	Lys	17	65	37	13	0	23	21	5	14	9	6	9845	11	0	6	22	14	0	4	13
M	Met	2	7	0	0	5	4	0	0	0	7	14	2	9672	5	0	5	2	0	0	12
F	Phe	2	3	0	0	0	0	0	0	4	18	10	0	18	9879	0	5	2	30	74	2
P	Pro	23	0	9	1	0	13	13	7	0	3	0	3	0	0	9850	11	5	0	0	4
S	Ser	59	2	67	28	27	22	16	26	17	4	3	14	23	6	15	9598	69	0	0	7
T	Thr	30	0	25	3	8	20	14	6	8	24	5	10	11	3	8	76	9759	0	0	20
W	Trp	0	0	0	0	0	0	0	0	4	0	0	0	0	8	0	0	0	9941	7	0
Y	Tyr	0	0	0	0	0	0	0	0	4	0	0	2	0	51	0	0	0	17	9909	0
V	Val	27	0	7	5	18	12	10	6	3	156	22	12	82	3	8	9	25	0	0	9783

REPLACEMENT AMINO ACID

Figure 9-7. Mutation probability matrix for the evolutionary distance of 2 PAMs. An element of this matrix, M_{ij}, gives the probability that the amino acid in column j will be replaced by the amino acid in row i after a given evolutionary interval, in this case 2 accepted point mutations per 100 amino acids. Thus, there is a .95% probability that Asp will be replaced by Glu. To simplify the appearance, the elements are shown multiplied by 10,000.

Matrices BLOSUM (BLOcks Substitution Matrix)

La matrice BLOSUM est construite à partir de bloques conservés dans les séquences alignées. Par exemple:

AAACDA - - -BBCDD
DABCDA- A-BACBB
BACDDABA-BDCAA

Le calcul des entrées de la matrice est réalisé comme suit:

q(i, j) fréquence observée de i et j étant a la même position après évolution des séquences. La valeur est calculée a partir de l'alignement: q(i, j) est la fréquence observée pour une paire d'acides-aminés {i, j} d'apparaître dans la même colonne d'alignement. Par exemple, nous avons 8 colonnes ayant 3 paires chacune; seulement deux paires sont {C, D}, donc q(C, D) = 2 / (8 * 3).

p(i) est la fréquence d'occurrence de i dans la paire {i, j}

p(i) = q(i, i) + somme de q(i, j) / 2 si i <> j

La fréquence aléatoire dérivée de la distribution d'acides-aminés

$$p(i, j) = \begin{cases} 2 * p(i) * p(j) & \text{si } i <> j \\ p(i) * p(i) & \text{si } i = j \end{cases}$$

$$\text{BLOSUM-X } (i, j) = 2 * \log_2 (q(i, j) / p(i, j))$$

BLOSUM-X est construite a partir de protéines que ont au plus X% de similarité de séquence. L'élimination de séquences ne satisfaisant pas cette contrainte est faite en éliminant les séquences d'un bloque, ou en trouvant un cluster de séquences similaires et en remplaçant le cluster complet par une nouvelle séquence.

Compairaison avec les matrices PAM :

BLOSUM utilise plus de données que PAM => statistiques plus fiables

BLOSUM utilise un alignement locale de plusieurs protéines de la même famille et PAM utilise un alignement globale => BLOSUM collectionne les statistiques a partir des seules régions qui comptent.

Les performances de BLOSUM 45, 62 et 80 et PAM 120, 160 et 250 ont été comparées. BLOSUM permet d'obtenir des alignements plus précis: PAM a manqué 30-31 positions (en moyenne sur l'ensemble test) et BLOSUM 6-9 positions.

Comme PAM, BLOSUM n'a pas d'évaluation de coût de gap.

BLOSUM-62 est la matrice plus utilisée pour l'alignement de paires de séquences et pour la recherche dans les bdd. Elle est particulièrement efficace dans l'alignement sans gaps.

BLOSUM-50 semble plus efficace pour l'alignement avec gaps.

	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	X	Y	Z
A	4	-2	0	-2	-1	-2	0	-2	-1	-1	-1	-1	-2	-1	-1	-1	1	0	0	-3	-1	-2	-1
B	-2	6	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-1	-3	2
C	0	-3	9	-3	-4	-2	-3	-3	-1	-3	-1	-1	-3	-3	-3	-3	-1	-1	-1	-2	-1	-2	-4
D	-2	6	-3	6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-1	-3	2
E	-1	2	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-1	-2	5
F	-2	-3	-2	-3	-3	6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	-1	3	-3
G	0	-1	-3	-1	-2	-3	6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-1	-3	-2
H	-2	-1	-3	-1	0	-1	-2	8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	-1	2	0
I	-1	-3	-1	-3	-3	0	-4	-3	4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1	-1	-3
K	-1	-1	-3	-1	1	-3	-2	-1	-3	5	-2	-1	0	-1	1	2	0	-1	-2	-3	-1	-2	1
L	-1	-4	-1	-4	-3	0	-4	-3	2	-2	4	2	-3	-3	-2	-2	-2	-1	1	-2	-1	-1	-3
M	-1	-3	-1	-3	-2	0	-3	-2	1	-1	2	5	-2	-2	0	-1	-1	-1	1	-1	-1	-1	-2
N	-2	1	-3	1	0	-3	0	1	-3	0	-3	-2	6	-2	0	0	1	0	-3	-4	-1	-2	0
P	-1	-1	-3	-1	-1	-4	-2	-2	-3	-1	-3	-2	-2	7	-1	-2	-1	-1	-2	-4	-1	-3	-1
Q	-1	0	-3	0	2	-3	-2	0	-3	1	-2	0	0	-1	5	1	0	-1	-2	-2	-1	-1	2
R	-1	-2	-3	-2	0	-3	-2	0	-3	2	-2	-1	0	-2	1	5	-1	-1	-3	-3	-1	-2	0
S	1	0	-1	0	0	-2	0	-1	-2	0	-2	-1	1	-1	0	-1	4	1	-2	-3	-1	-2	0
T	0	-1	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	0	-1	-1	-1	1	5	0	-2	-1	-2	-1
V	0	-3	-1	-3	-2	-1	-3	-3	3	-2	1	1	-3	-2	-2	-3	-2	0	4	-3	-1	-1	-2
W	-3	-4	-2	-4	-3	1	-2	-2	-3	-3	-2	-1	-4	-4	-2	-3	-3	-2	-3	11	-1	2	-3
X	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Y	-2	-3	-2	-3	-2	3	-3	2	-1	-2	-1	-1	-2	-3	-1	-2	-2	-2	-1	2	-1	7	-2
Z	-1	2	-4	2	5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-1	-2	5

BLOSUM 62

Matrice de Gonnet

Matrice unique, calculée d'après un très large échantillon (bdd SwissеProt) en utilisant de **l'alignement entre paires de séquences**.

Une mesure de distance entre acides-aminés classique a été utilisée pour estimer l'alignement entre protéines. En suite, les résultats ont été utilisées pour estimer une nouvelle matrice de distance, qui a été utilisée en suite pour estimer l'alignement entre protéines et pour calculer une nouvelle matrice, et ainsi de suite de façon itérative jusqu'a convergence.

Il a été observé que les matrices de distance (toutes initialement normalisées a 250 PAM) étaient différentes si calculées a partir d'ensembles de protéines proches ou d'ensemble de protéines distantes. Pour cette raison Gonnet a été proposée comme une matrice a employer dans un premier filtrage de protéines proches, suivie par des raffinements basés sur PAM.

Evaluation du coût des gaps

Les alignements ont été aussi utilisés pour estimer des pénalités de gaps appropriées.
Les données ont suggeré que P, la probabilité que un gap soit de longueur k, suit la relation

$$10\ln(P)= -36.31+7.44\ln(\text{PAM distance}) – 14.93\ln(k)$$

Une telle relation représente la réponse plus précise, mais dans le cas ou la distance PAM n'est pas connue, alors une estimation suggère

$$10\ln(P)=-20.63-1.65(k-1)$$

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	11.5																			
S	0.1	2.2																		
T	-0.5	1.5	2.5																	
P	-3.1	0.4	0.1	7.6																
A	0.5	1.1	0.6	0.3	2.4															
G	-2.0	0.4	-1.1	-1.6	0.5	6.6														
N	-1.8	0.9	0.5	-0.9	-0.3	0.4	3.8													
D	-3.2	0.5	0.0	-0.7	-0.3	0.1	2.2	4.7												
E	-3.0	0.2	-0.1	-0.5	0.0	-0.8	0.9	2.7	3.6											
Q	-2.4	0.2	0.0	-0.2	-0.2	-1.0	0.7	0.9	1.7	2.7										
H	-1.3	-0.2	-0.3	-1.1	-0.8	-1.4	1.2	0.4	0.4	1.2	6.0									
R	-2.2	-0.2	-0.2	-0.9	-0.6	-1.0	0.3	-0.3	0.4	1.5	0.6	4.7								
K	-2.8	0.1	0.1	-0.6	-0.4	-1.1	0.8	0.5	1.2	1.5	0.6	2.7	3.2							
M	-0.9	-1.4	-0.6	-2.4	-0.7	-3.5	-2.2	-3.0	-2.0	-1.0	-1.3	-1.7	-1.4	4.3						
I	-1.1	-1.8	-0.6	-2.6	-0.8	-4.5	-2.8	-3.8	-2.7	-1.9	-2.2	-2.4	-2.1	2.5	4.0					
L	-1.5	-2.1	-1.3	-2.3	-1.2	-4.4	-3.0	-4.0	-2.8	-1.6	-1.9	-2.2	-2.1	2.8	2.8	4.0				
V	0.0	-1.0	0.0	-1.8	0.1	-3.3	-2.2	-2.9	-1.9	-1.5	-2.0	-2.0	-1.7	1.6	3.1	1.8	3.4			
F	-0.8	-2.8	-2.2	-3.8	-2.3	-5.2	-3.1	-4.5	-3.9	-2.6	-0.1	-3.2	-3.3	1.6	1.0	2.0	0.1	7.0		
Y	-0.5	-1.9	-1.9	-3.1	-2.2	-4.0	-1.4	-2.8	-2.7	-1.7	2.2	-1.8	-2.1	-0.2	-0.7	0.0	-1.1	5.1	7.8	
W	-1.0	-3.3	-3.5	-5.0	-3.6	-4.0	-3.6	-5.2	-4.3	-2.7	-0.8	-1.6	-3.5	-1.0	-1.8	-0.7	-2.6	3.6	4.1	14.2

Les éléments de la matrice sont 10ln(P/Q), où P est la probabilité que les acides-aminés soient alignés, et Q est la probabilité que ces acides-aminés soient alignés par chance.

Choix d'une matrice

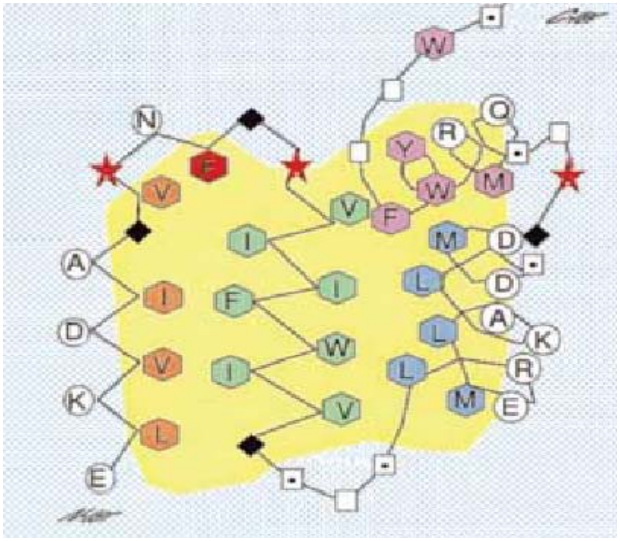
Toutes les matrices sont meilleurs pour des alignement globaux comparés à des alignements locaux.

Les performances sont très sensibles aux pénalités de gaps

Pour les recherches dans les bdd, en moyenne:

Gonnet > BLOSUM 50 et BLOSUM 62 > PAM250.

Matrices définies sur les bloques hydrophobes



Une matrice définie pour les bloques et une matrice définie hors les bloques (présentées dans le 2ème cours).

Application: détection de protéines homologues

A homologue à B : A et B ont un ancêtre commun

Méthode de détection de nouvelles protéines homologues :
alignement entre paires de séquences et scores de sélection appropriés

Construction de scores de sélection: tests sur les alignements de paires

Significativité d'un alignement: test statistique basé sur les séquences aléatoires

Pour déterminer si un alignement est statistiquement significatif on peut réaliser un test de permutation comme suit:

a. Réarranger les résidus aléatoirement dans une ou toutes les séquences
b. Aligner les séquences permutées
c. Noter les scores obtenus pour cet alignement

On répète ces étapes un grand nombre de fois, et on génère en suite une distribution des scores d'alignement que l'on a obtenus pour des séquences arrangées aléatoirement. On peut alors regarder la distribution du maximum de N scores de séquences aléatoires indépendantes. Si la probabilité que ce maximum soit plus grande que le meilleur score observé est petite, alors l'observation peut être considérée comme significative.

Test de fiabilité de la prédiction avec des protéines de structure connue

E ensemble de séquences a structure similaires
X ensemble de séquence obtenu par alignement de paires

On veut évaluer le processus de sélection des séquences. L'idéale serait de sélectionner par alignement l'ensemble E, donc X=E. En générale on a des faux positifs dans X ainsi que des séquences qui n'ont pas été détectées.

Sélectivité: pourcentage de séquences de E qui sont dans X.

Sensitivité: pourcentage de séquence de X qui sont dans E.

Il faut toujours calculer les deux!

Test pour l'évaluation de la précision des alignements (possiblement multiples)

Cela demande un large ensemble d'alignements de références fiables. Cet ensemble sera alors considéré comme vrai et employé dans les test.

On prendra en considération un ensemble de protéines dont on connaît la structure et on considérera les alignements de séquences induites par l'alignement structurale.

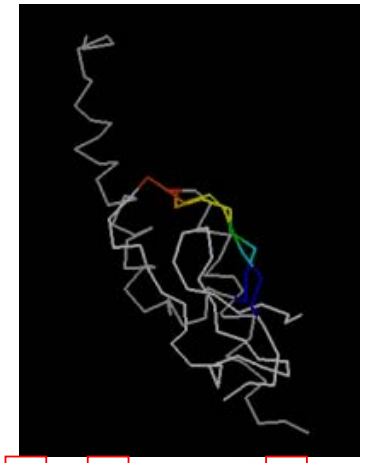
Ils existent plusieurs critères de sélection des régions correctement alignées:

- Toutes les régions sans gaps (problèmes avec: régions conservées sont limitées au noyau de la protéine, pas d'intérêt dans la plupart des coils).
- les structures secondaires conservées dans les séquences alignées (problèmes avec: effet de bord, définition de coil)
- la distance RMSD (Root Mean Square Deviation) entre structures et une borne souhaitable associée.

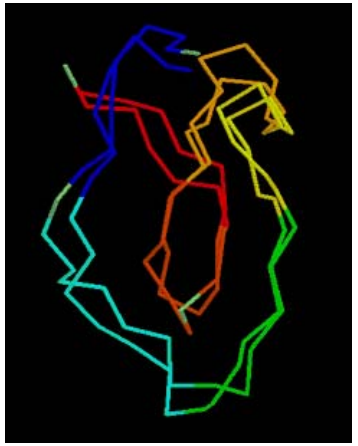
RMSD = $\sqrt{\Sigma_i (q_i^e - q_i^t)^2 / nsites}$, ou Σ_i et la somme sur i=1,...,n positions (sites),
q_i^e est la charge effective a la position i,
q_i^t est la charge test a la position i.

Comment utiliser le score d'alignement et l'information sur les bloques hydrophobes pour détecter protéines homologues par filtrage de large bases de données ?

Weak Homology : some difficult cases



```
1fleI     .AQEPVKGPVSTKPGSCPIILIRCAMLNPPNRCLKD.TDCPGIKKCCEGSCGMACFVPQ....
1a0aA     MKRESHKHAEQARRNRLAVALHELASLIPAEWKQQNVSAAPSKATTVEAACRYIRHLQQNGST
```

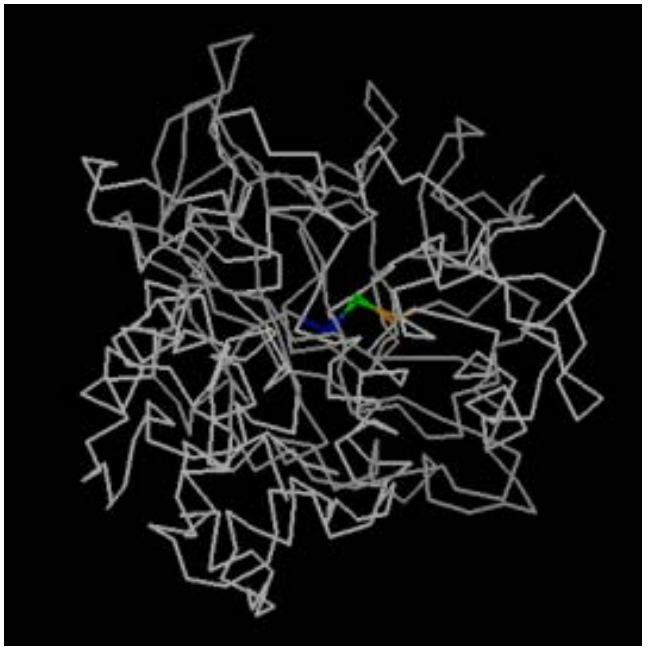


Small blocks of identities

```
1fleI       AQEPVKGPVSTKPGSCPIILIRCAMLNPPN....RCLKDTDCPGIKKCCEGSCG.MACFVP.Q
1udkA       .........NEKSGSCPDMSM...PIPPLGICKTLCNSDSGCPNVQKCCKNGCGFMTCTTPVP
```

Non homologous
with some small blocks of identities

```
1cnsA  SVSSIVSRAQFDRMLLHRNDGACQAKGFYTYDAFVAAAAAFSGFGTTGSADVQKREVAAFLAQTSHE.TTGG.WATAPDGAFAWGYCFKQERGASSDYCT
1j83A  ...QPTAPKDFSSGFWDFNDGTTQ..GFGV.....NPDSPITAINVENANN..ALKISNLNSKGSNDLSEGNFWANVRISADIWG....QSINIYGD..T

       PSAQWPCAPGKRYYGRGPIQLSHNYNYGPAGRAIGVDLLANPDLVATDATVSFKTAMWFWMTAQPPKPSSHAVIVGQWSPSGADRAAGRVPGFGVITNII
       KLTMDVIAP.......TPVNVS..IAATPQSSTHG...WGNP.......TRAIR..VWTNNFVAQTDGTYKATLTISTNDSPNFNTIATDAADSVVTNMI

       NGGIECGHGQDSRVADRIGFYKRYCDILGVGYGNNLDCYSQRPFA
       ...LFVGSNSDNISLDNIKFTK.......................
```



Similar score to weakly homologous proteins

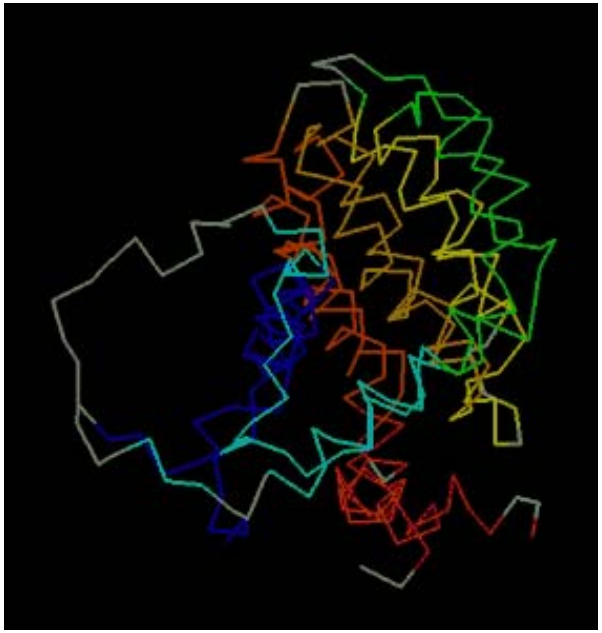
Homologous
with no block of identities

```
1bwwA  TPDIQMTQSPSSLSASVGDRVTITCQASQ..DIIKYLNWYQQKPGKAPKLLIYEASNLQAGVPSRFSGSGSGTDYTFTISSLQPEDIATYYCQQYDSL.P
1cdcA  .......RDSGTVWGALGHGINLNIPNFQMTDDIDEVRW..ERGSTLVAEFKRKMKPFLK..SGAFEILANG...DLKIKNLTRDDSGTYNVTVYSTNGT

                                                                      ---

       YTFGQGTKLQIT.
       RILDKALDLRILE
```

Non-homologous : but partial homology

```
1k3kA  ..........MDEDVL..PGEVLAIEGIFMACGLN.EPEYLYHPLLSPIKLYITGLMRDKESL.......FEAMLANVRFHSTTGIDQLGLSMLQVSGDG
1gjhA  MAHAGRTGYDNREIVIKYIHYHLSQRGYEWDAGDDVEENRTEAPEGTESEVVHLTLRQAGDDFSRRYRRDFAEMSSQLHLTPFTARGRFATVVEELFRDG

                                                                                    - - -
       NMNWGRALAILTFGS.FVAQKLSNE.PHLRDFALAVLPAYAYEAIGPQWFRARGGWRGLKAYCTQVLTDDDDLEHHHHHH
       .VNWGRIVAFFEFGGVMCVESVNREMSPLVDNIALWMTEYLNRHLH.TWIQDNGGWDAFVELYGPSMR............
```

Globin-like domain in common



Analyse de la SCOP40

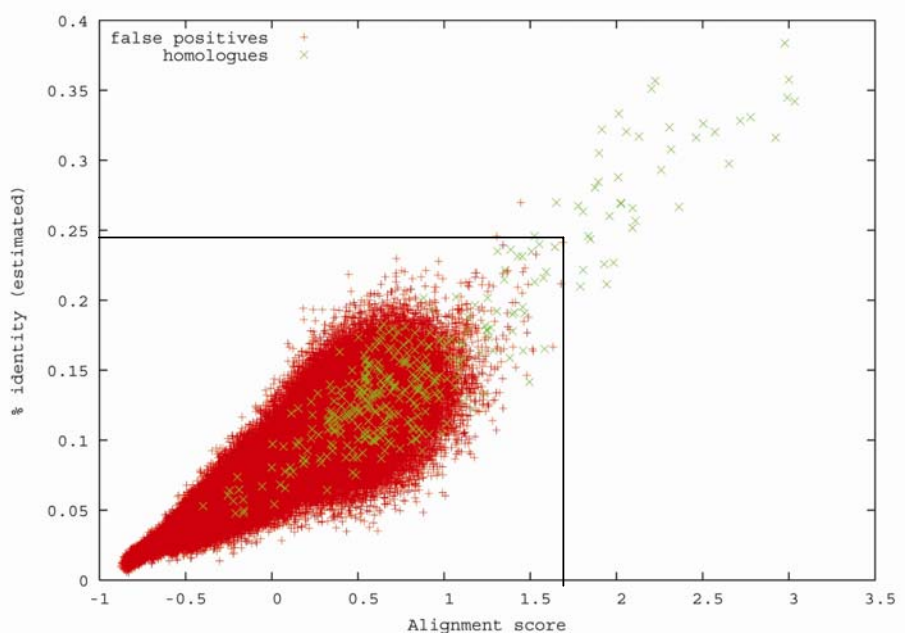
35 protéines de structure différente sélectionnées aléatoirement dans la SCOP40

(>5300 protéines)
(>100 résidus)

Aligner 35 séquences-requête avec >5300 protéines

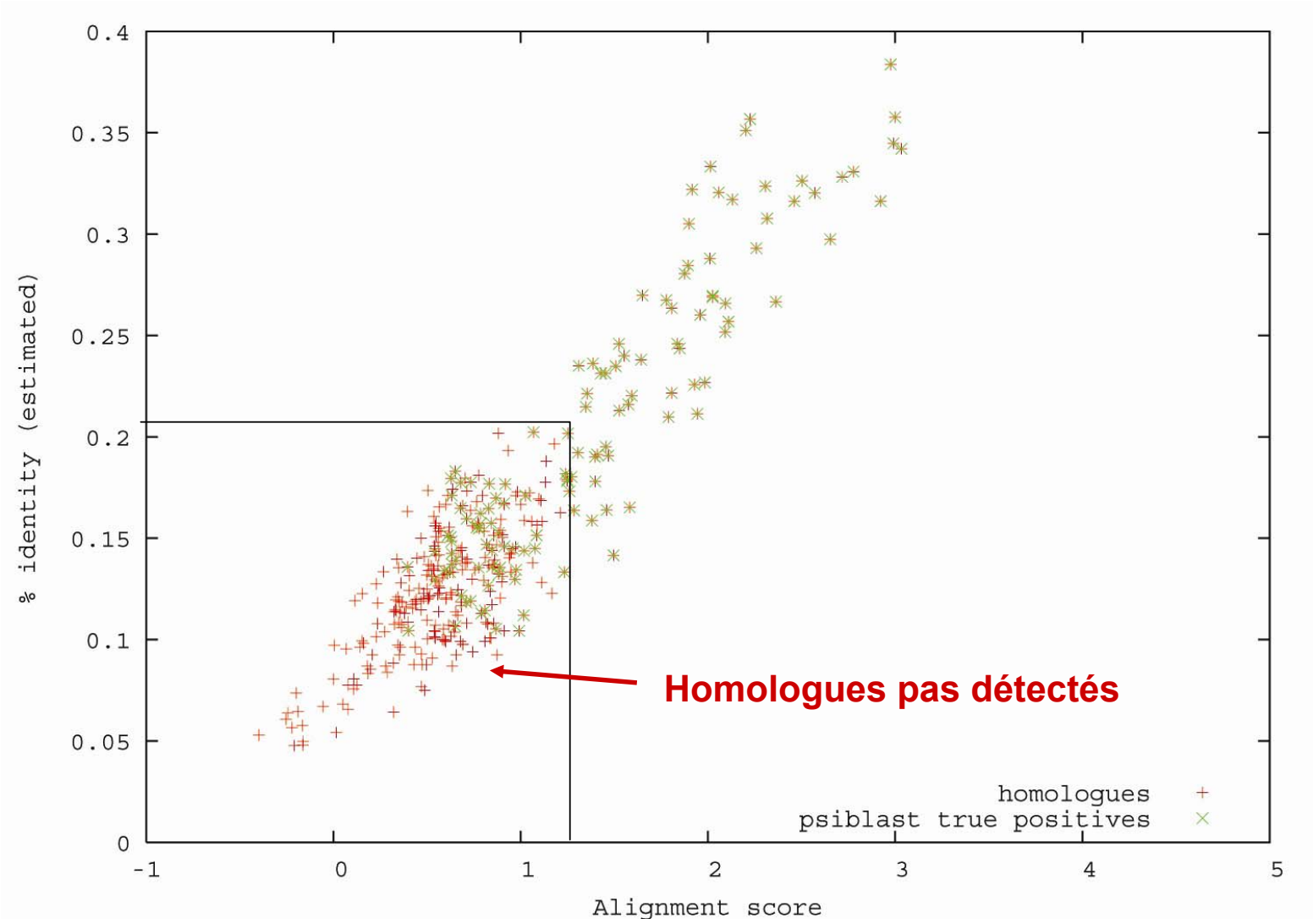
classifiées dans SCOP selon leur structure et fonction

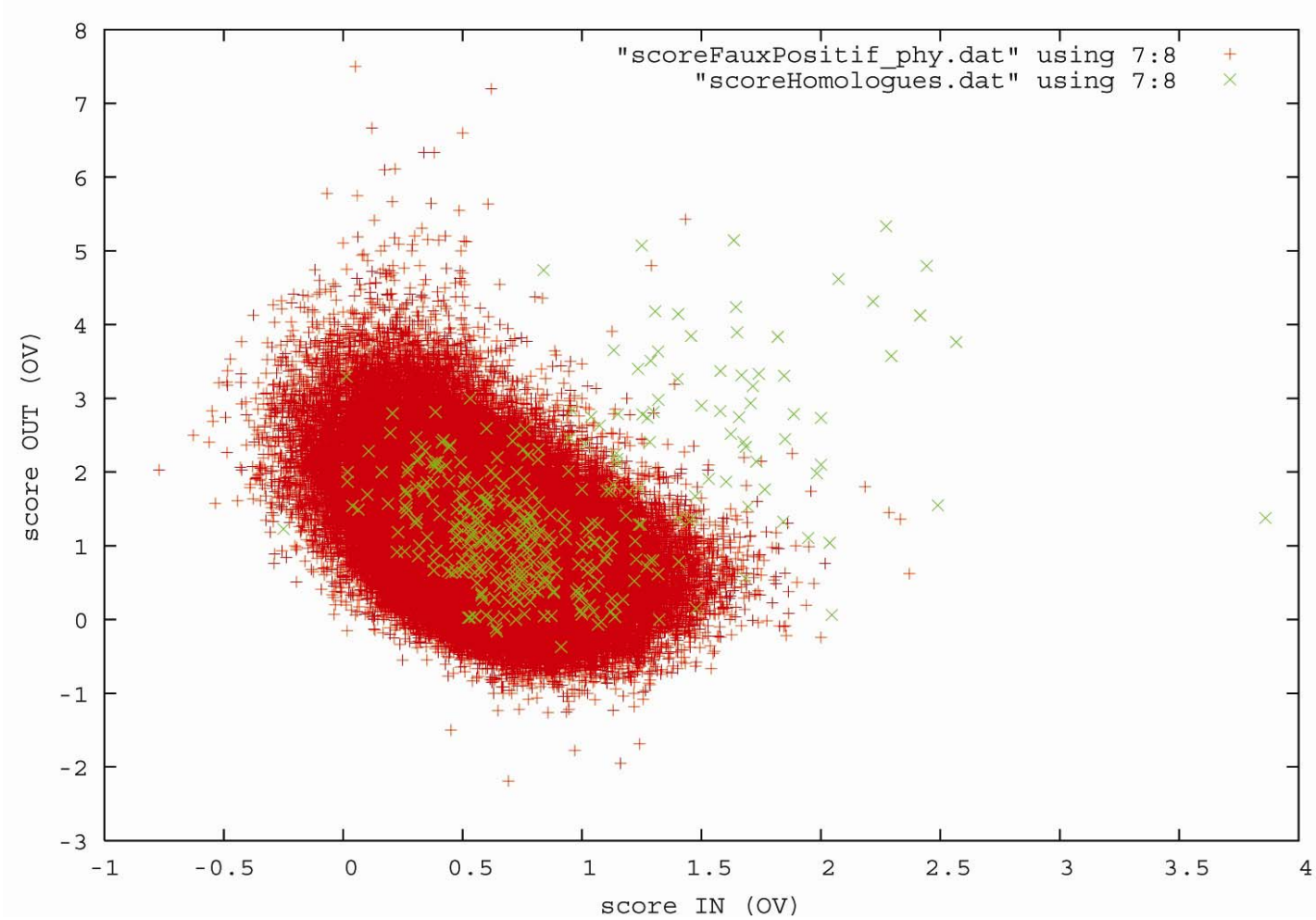
Sélectionner les candidats en utilisant des **paramètres de sélection** appropriés



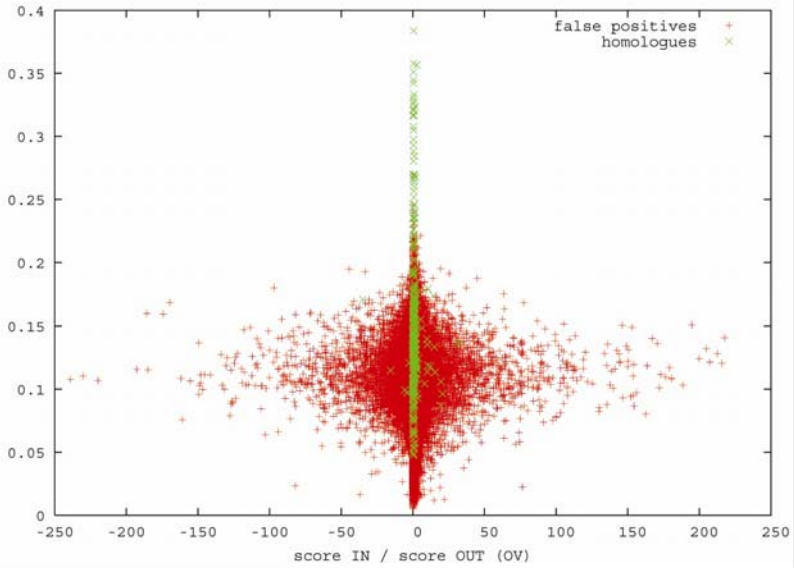
obtenu avec PHYBAL

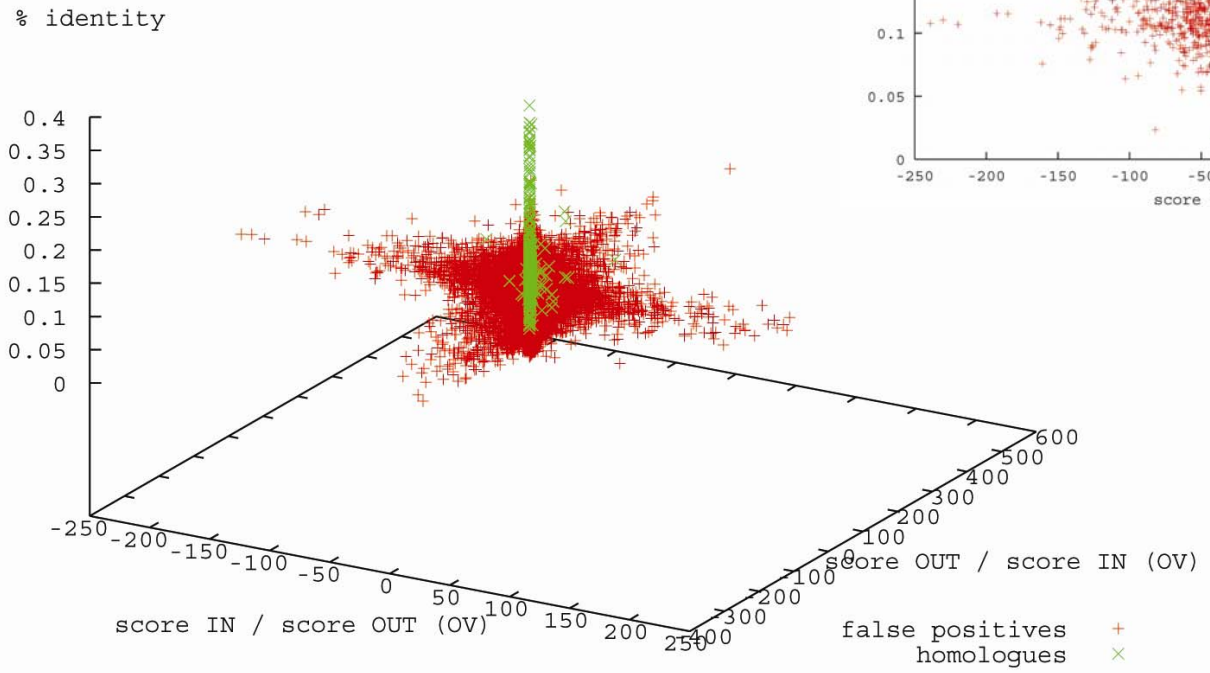
Comparaison des scores a la sélection de psi-BLAST



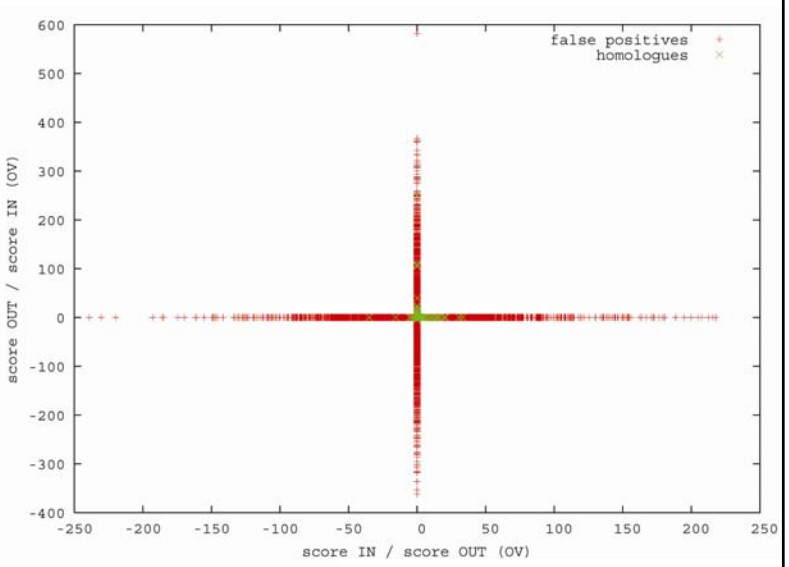

"scoreFauxPositif_phy.dat" using 7:8
"scoreHomologues.dat" using 7:8
score OUT (OV)
score IN (OV)
8
7
6
5
4
3
2
1
0
-1
-2
-3
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4

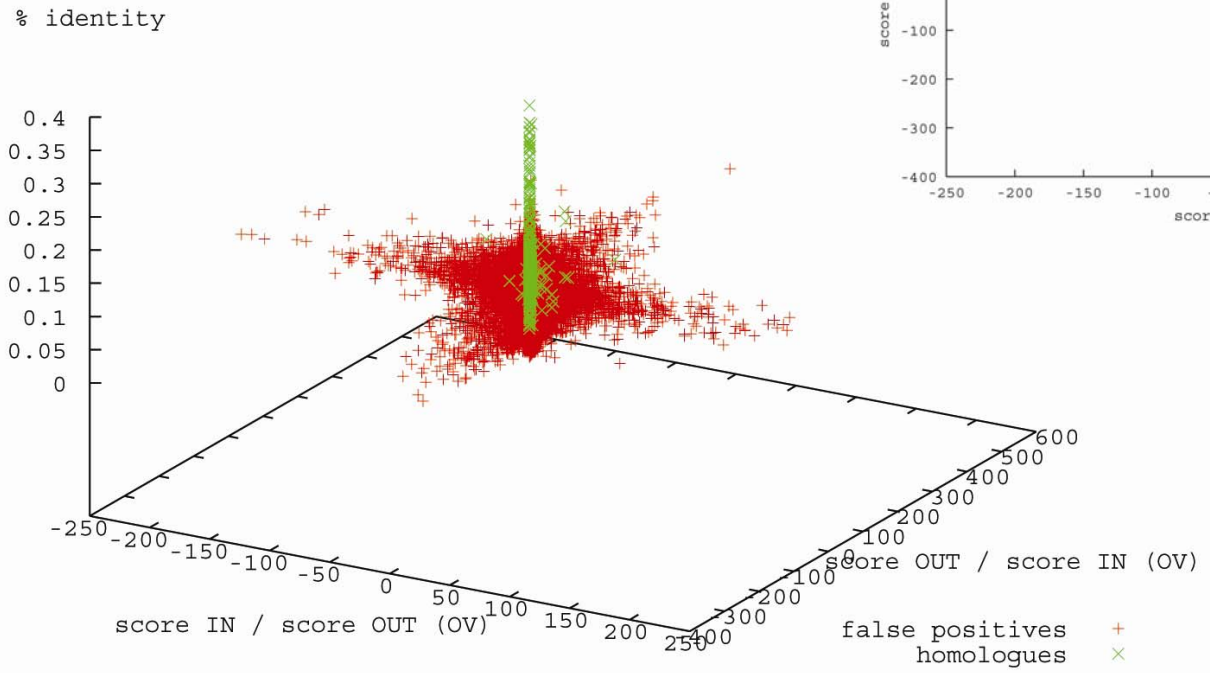
Sont les scores obtenus pour les régions IN et OUT des bloques hydrophobes proportionnelles?



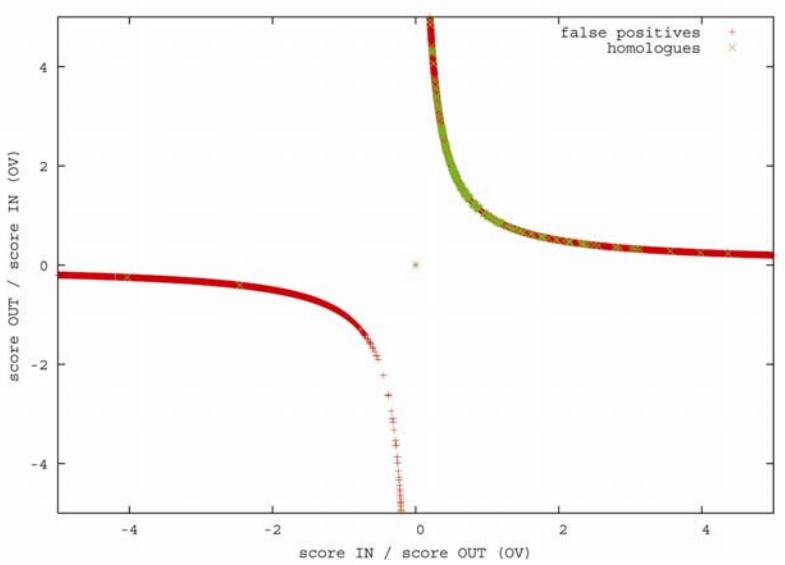


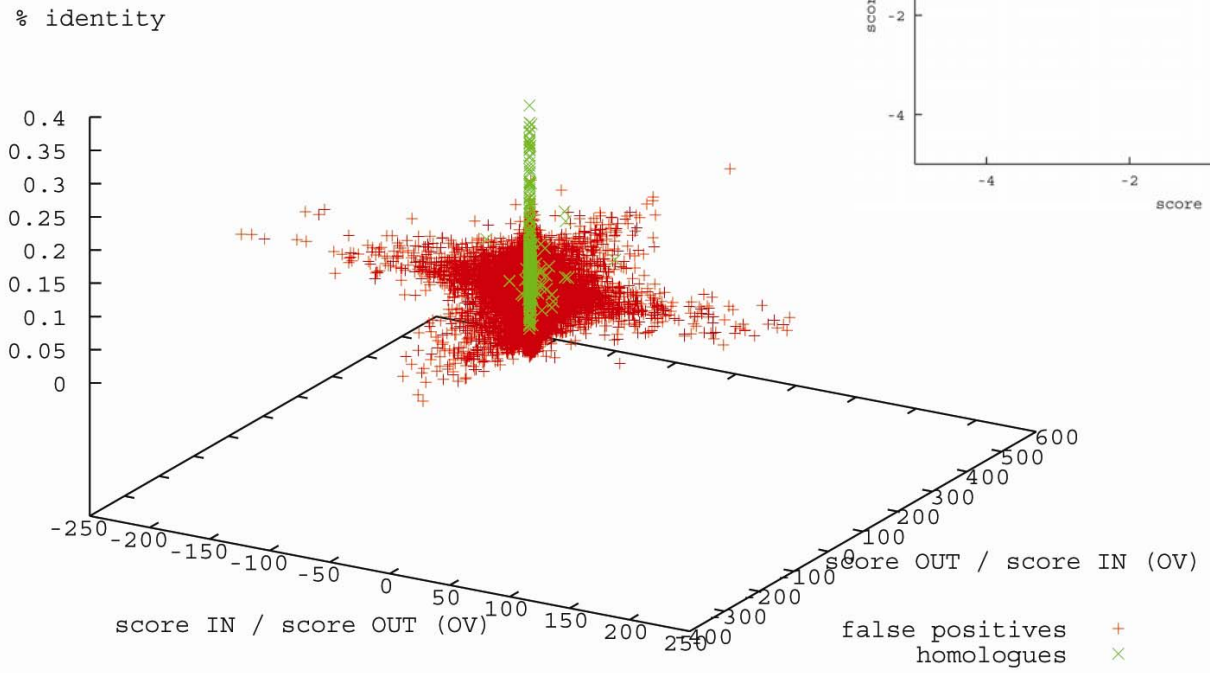
Sont les scores obtenus pour les régions IN et OUT des bloques hydrophobes proportionnelles?





Sont les scores obtenus pour les régions IN et OUT des bloques hydrophobes proportionnelles?





Travaux en cours, Baussand et Carbone, 2006

Alignement multiple

Étant donné un ensemble de séquences

$S_1, S_2, \ldots S_n$

un alignement multiple est une tuple

$S_1', S_2', \ldots S_n'$

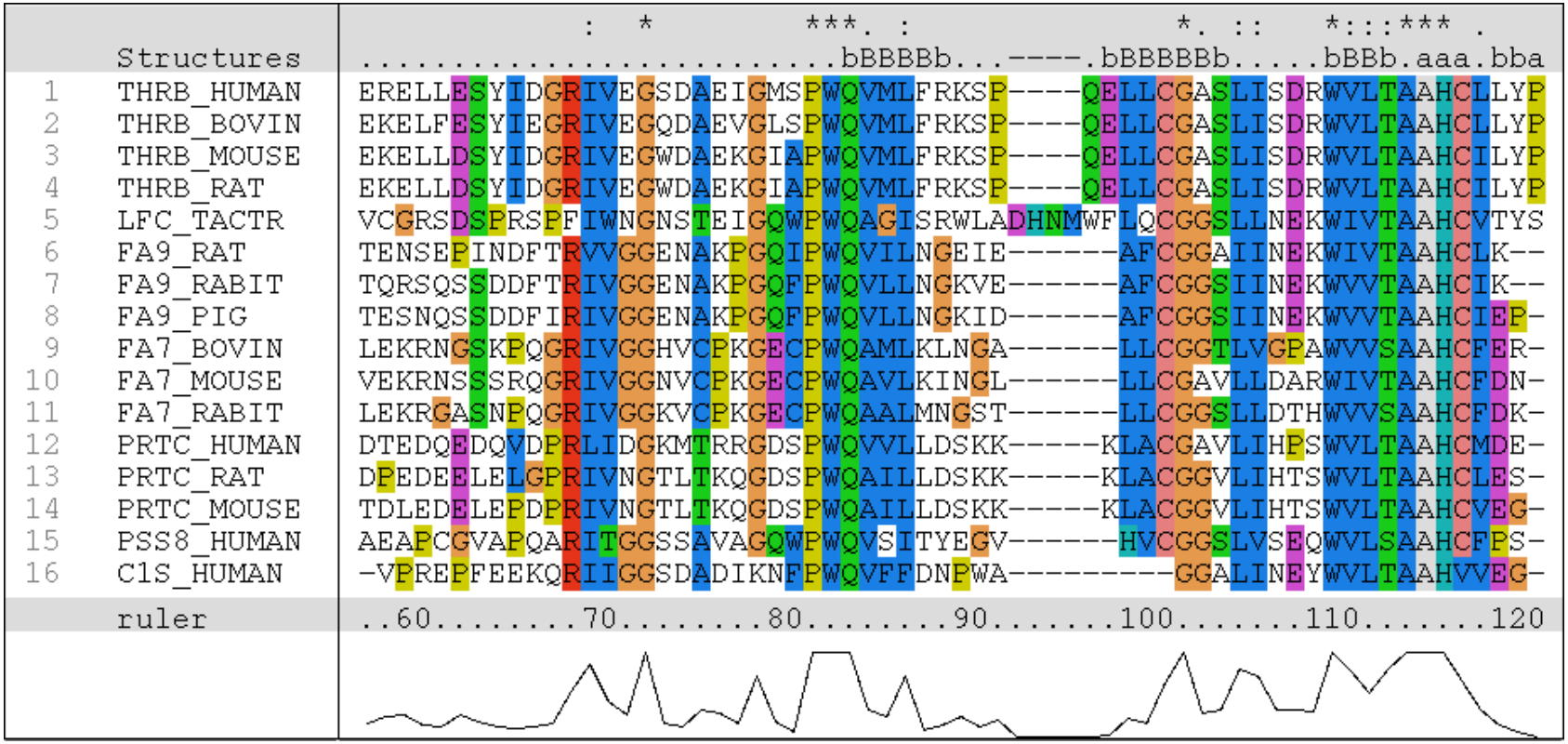
telle que

1. $|S_1'| = |S_2'| = \ldots |S_{n'}|$
2. S_i' est obtenu à partir de S_i par l'insertion d'occurrences d'espaces.

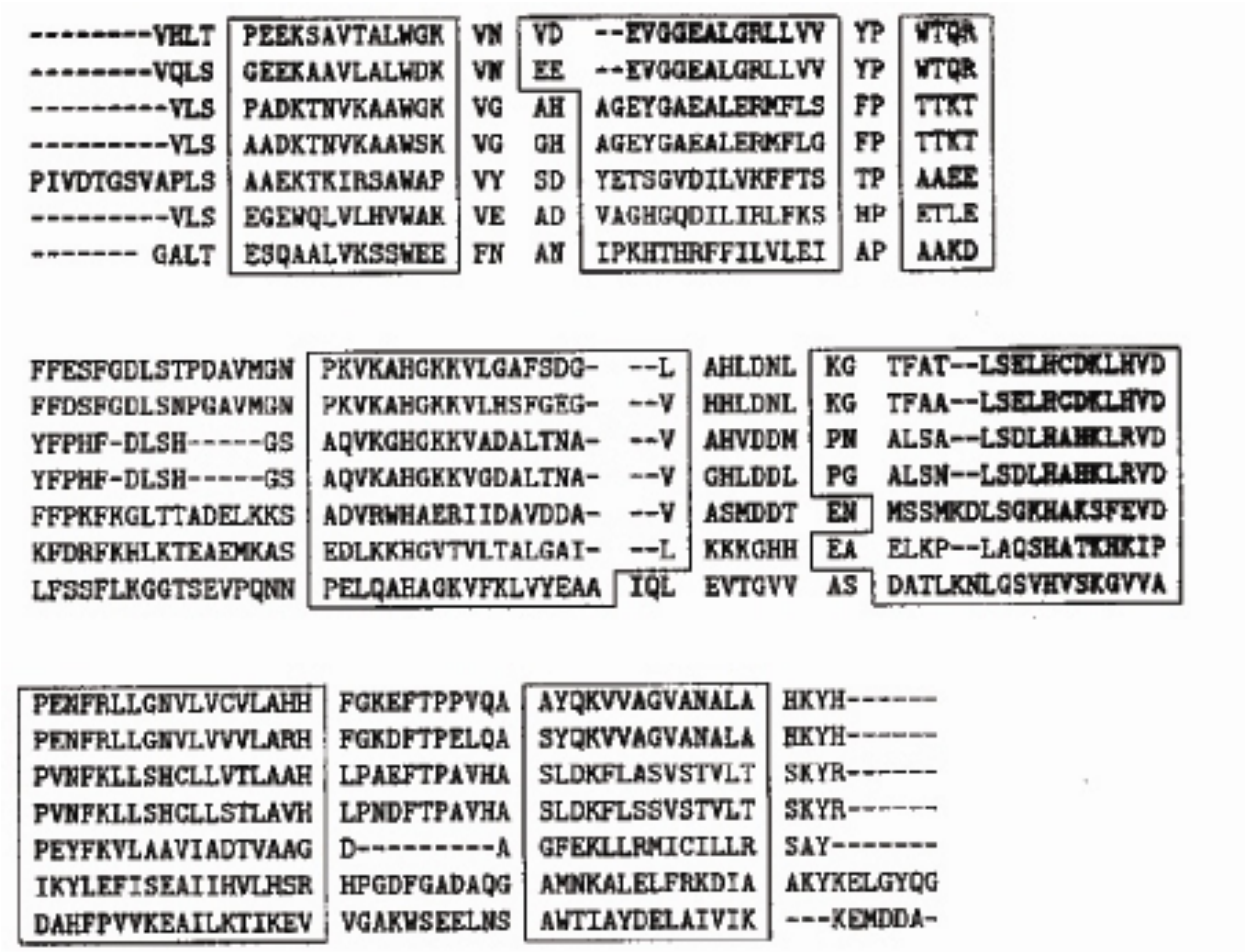
On veut déterminer un alignement optimal entre toutes les séquences, parce que la similarité multiple suggère une structure, une fonction et une origine (dans le sens de l'évolution) commune entre les différentes protéines.

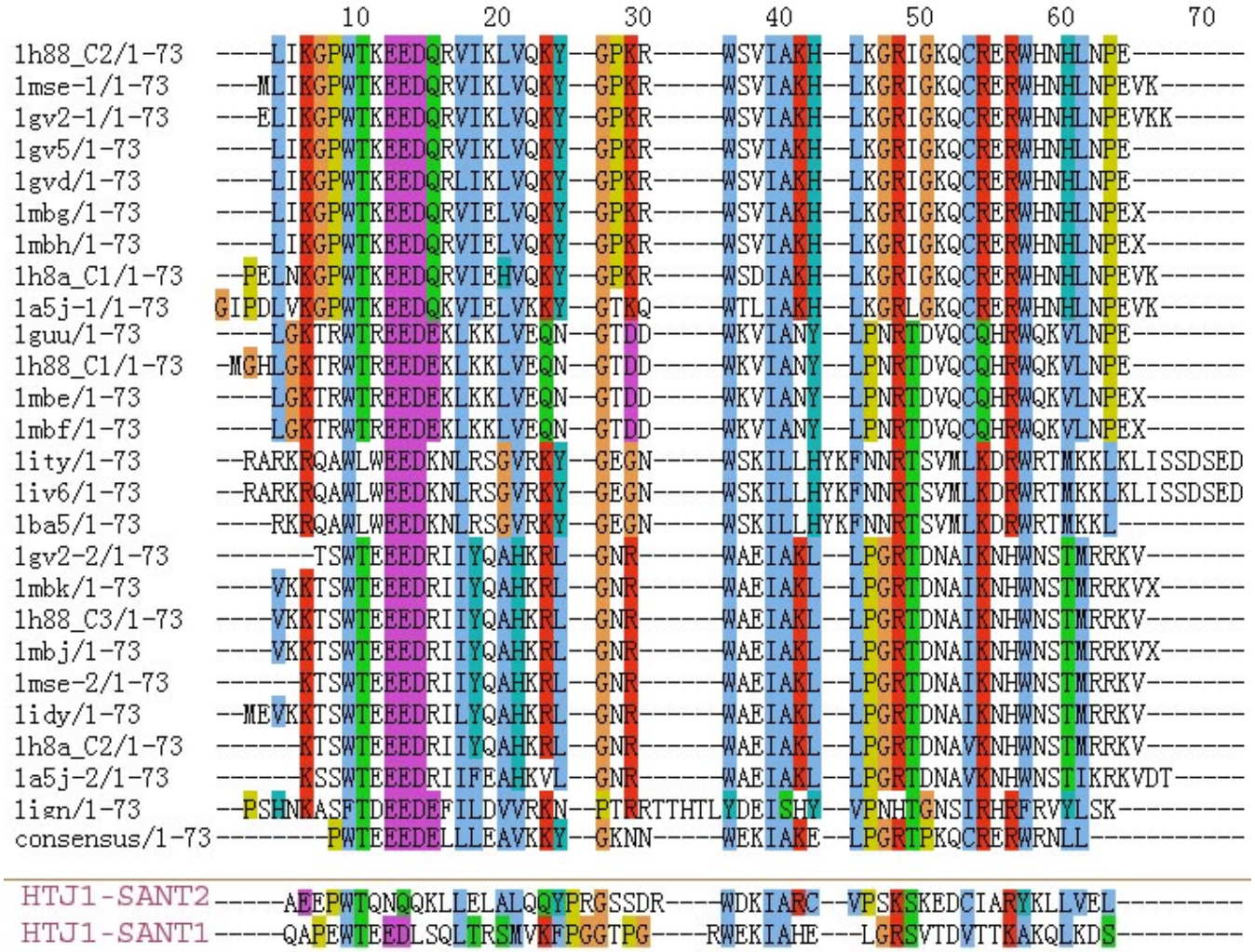
Le problème est NP-difficile (kn, où k est la longueur maximale des séquences).

Dans l'alignement multiple on retrouve plus d'information que dans l'alignement entre paires :



Les 7 hélices α des globines sont encadrées :





Propriétés physico-chimiques des acides-aminés:

ED	ionizable	TQS	H-bonding
ILVAWMCF	non-polaire/hydrophobe	HY	aromatique
RK	polaire/hydrophile	P	cyclique/proline

Généralisation de la notion d'alignement par paires

- L'alignement entre 2 séquences est représenté par une matrice a 2-lignes
- De façon similaire, nous représentons l'alignement de 3 séquences comme une matrice a 3-lignes

A T _ G C G _
A _ C G T _ A
A T C A C _ A

- Score: colonnes plus conservés, meilleur alignement

Alignements = chemins dans…

- Alignement de 3 séquences: ATGC, AATC,ATGC

	A	--	T	G	C

	A	A	T	--	C

	--	A	T	G	C

Alignements = chemins dans…

0	1	1	2	3	4
	A	--	T	G	C

	A	A	T	--	C

	--	A	T	G	C

x coordonnée

Alignements = chemins dans…

0	1	1	2	3	4
	A	--	T	G	C

0	1	2	3	3	4
	A	A	T	--	C

	--	A	T	G	C

x coordonnée

y coordonnée

Alignements = chemins dans…

0	1	1	2	3	4
	A	--	T	G	C
0	1	2	3	3	4
	A	A	T	--	C
0	0	1	2	3	4
	--	A	T	G	C

x coordonnée

y coordonnée

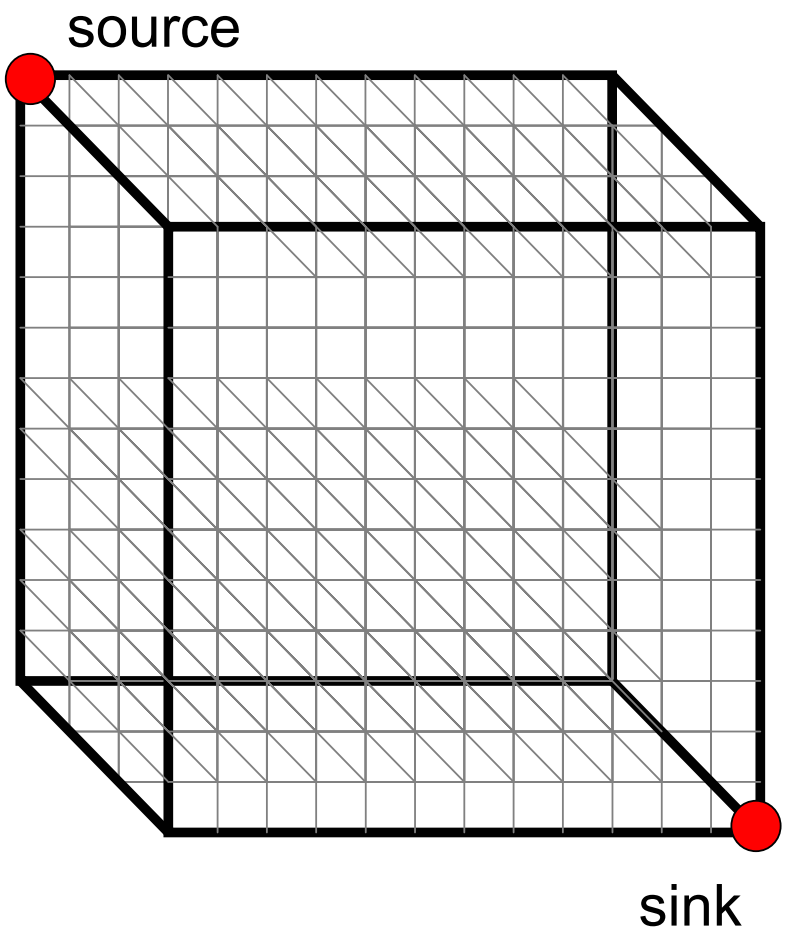
z coordonnée

- Chemin obtenu dans un espace *(x,y,z)* :

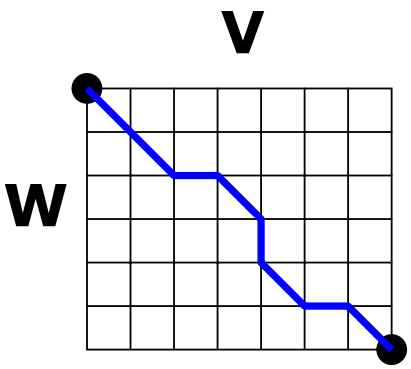
$(0,0,0) \rightarrow (1,1,0) \rightarrow (1,2,1) \rightarrow (2,3,2) \rightarrow (3,3,3) \rightarrow (4,4,4)$

Alignement de trois séquences

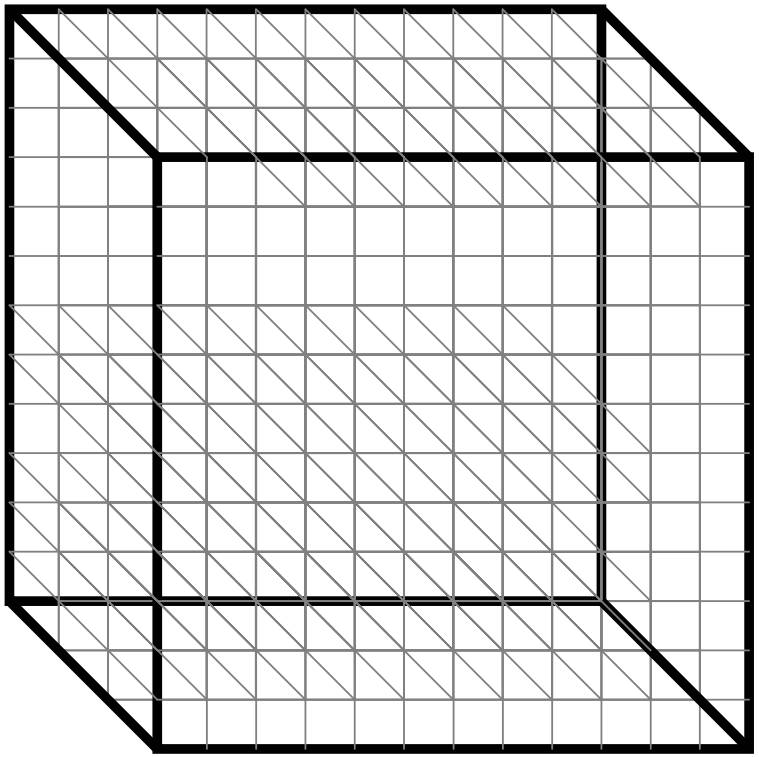
- Même stratégie que pour aligner 2 séquences
- Utiliser un cube en 3-D, “Manhattan Cube”, avec chaque axe qui représente une séquence a aligner
- Pour les alignements globales partir de la “source” jusqu’au “sink”



Alignement 2-D vs 3-D

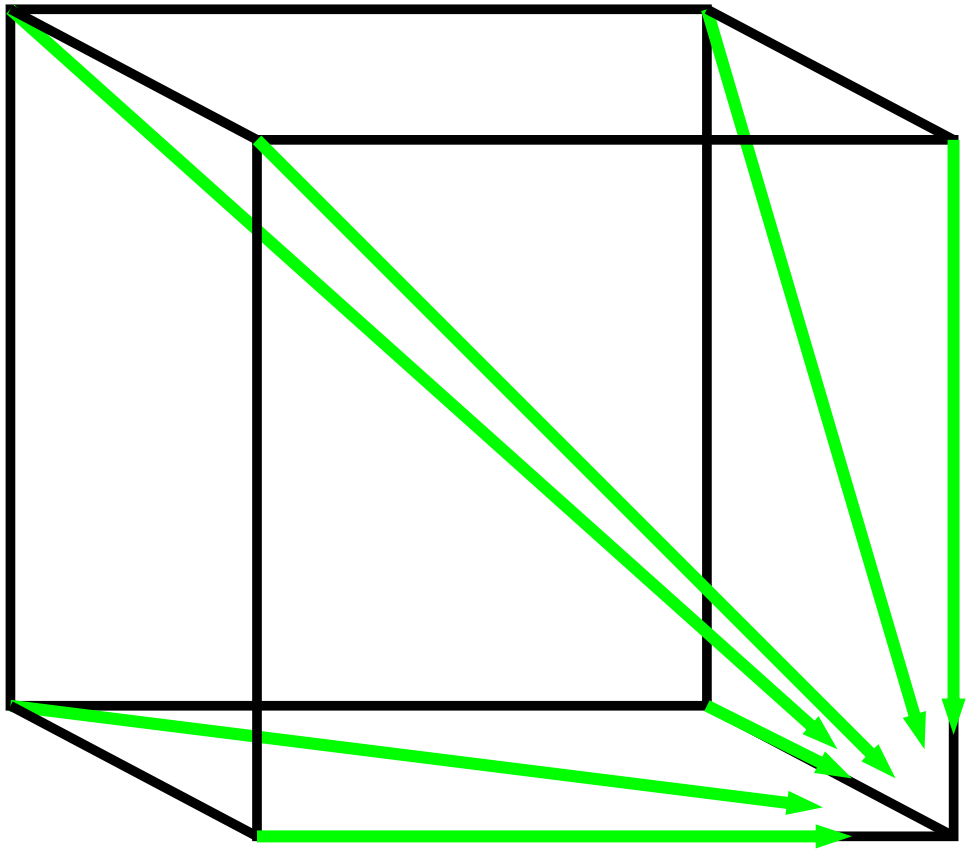


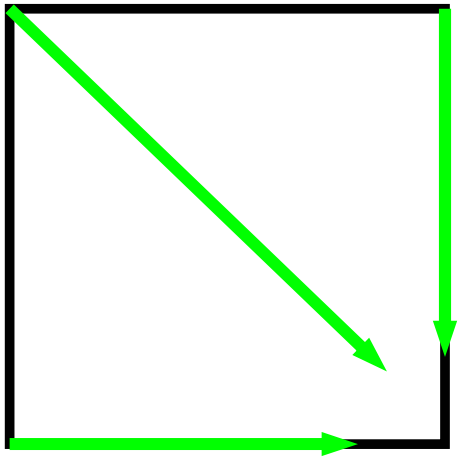
2-D edit graph



3-D edit graph

Cellule 3-D vs cellules 2-D

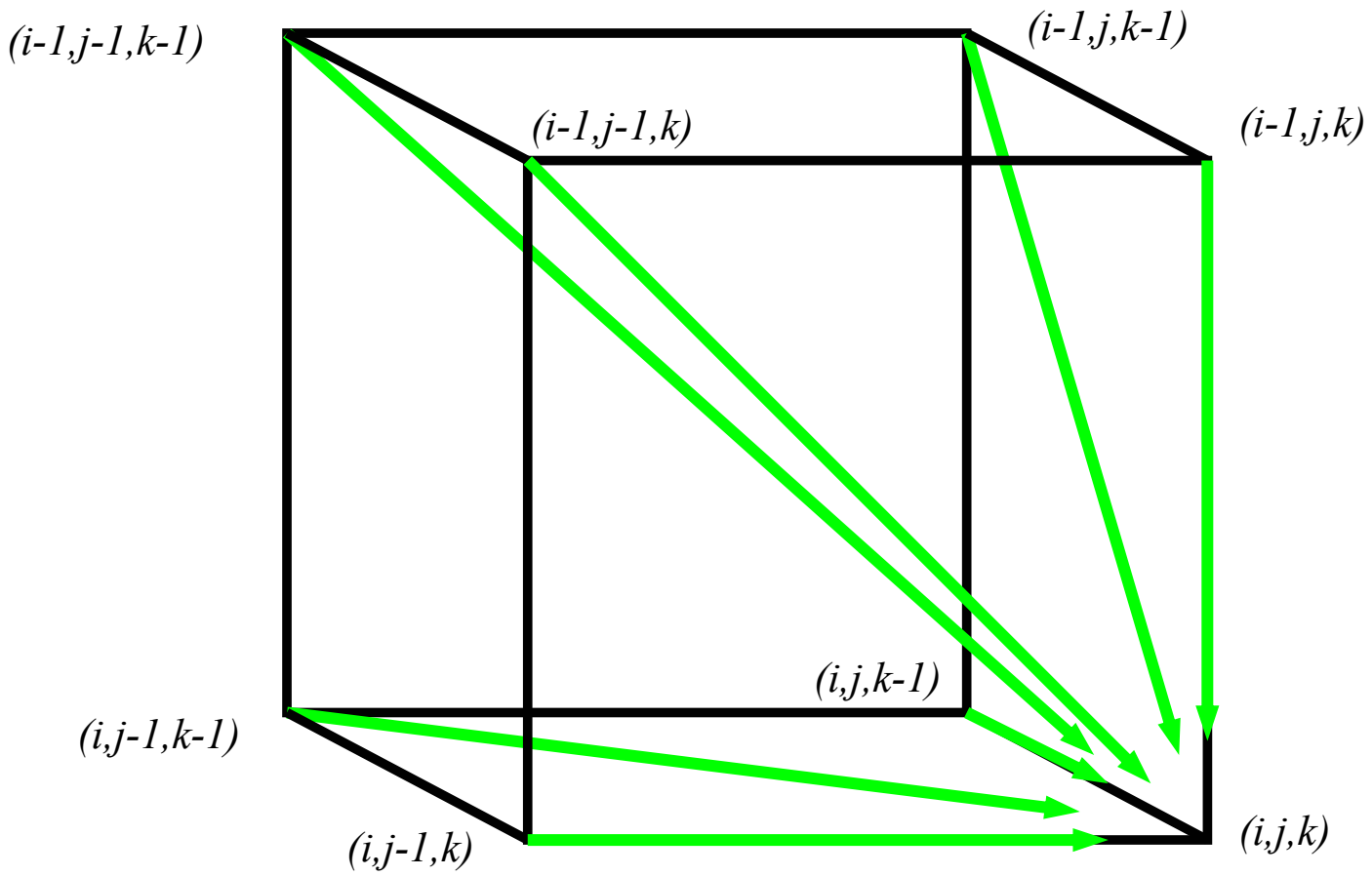




En **2-D**, 3 arcs dans chaque 1-carree

En **3-D**, 7 arcs dans chaque 1-cube

Architecture d'une cellule d'alignement en 3-D

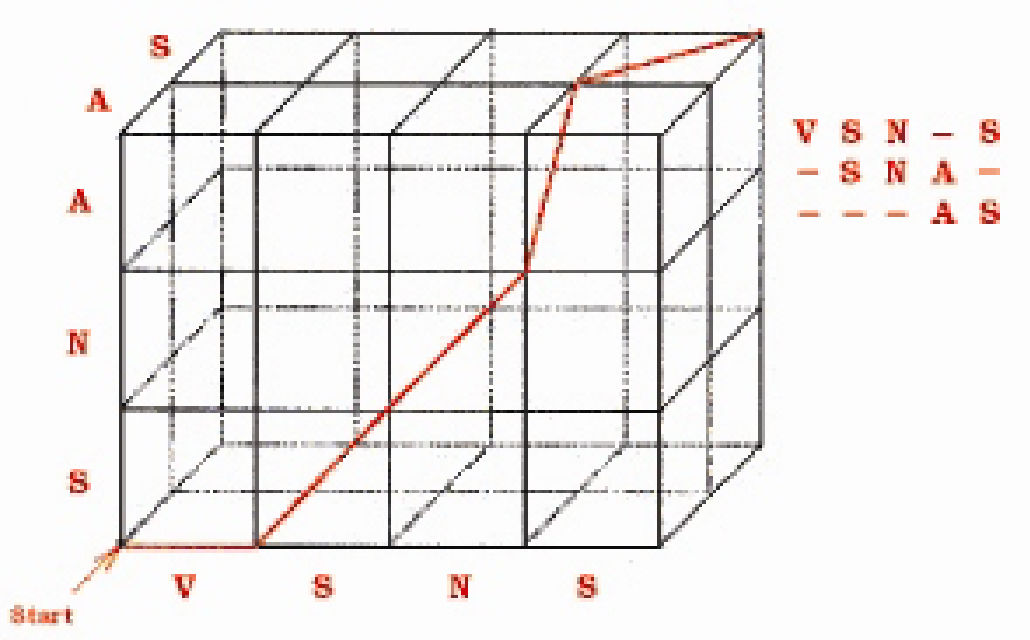


Alignement Multiple : programmation dynamique

- $s_{i,j,k} = \max$

$$
\begin{cases}
s_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k) & \text{diagonale: pas d'indels} \\
s_{i-1,j-1,k} + \delta(v_i, w_j, \_) & \\
s_{i-1,j,k-1} + \delta(v_i, \_, u_k) & \text{diagonale le long d'une face: 1 indel} \\
s_{i,j-1,k-1} + \delta(\_, w_j, u_k) & \\
s_{i-1,j,k} + \delta(v_i, \_, \_) & \\
s_{i,j-1,k} + \delta(\_, w_j, \_) & \text{diagonale le long d'un arc: 2 indels} \\
s_{i,j,k-1} + \delta(\_, \_, u_k) &
\end{cases}
$$

- $\delta(x, y, z)$ est une entree dans une matrice de score en 3-D



Chemin d'alignement pour 3 séquences

Multiple Alignement: complexité

- Pour 3 séquences de longueur ***n***, la complexité est $7n^3$; $O(n^3)$
- pour ***k*** séquences, la construction d'un Manhattan cube ***k***- dimensionnel, est faite en temps $(2^k-1)(n^k)$; $O(2^k n^k)$
- Conclusion: l'approche de programmation dynamique entre 2 séquences est facilement extensible a k séquences mais il est non-efficace du au temps exponentiel.

Solution en programmation dynamique pour r séquences

Etant données r séquences, on considère un hypercube r-dimensionnel D tel que $D(j_1,\ldots j_r)$ est le meilleur score pour l'alignement des préfixes de longueur $j_1, \ldots j_k$ des séquences $x_1, \ldots, x_r$, respectivement.

On définie $D(0,0, \ldots 0) = 0$ et on calcule

$$D(j_1,\ldots j_r)=\min_{\varepsilon \in \{0,1\}^n,\ \varepsilon\neq 0} \{D(j_1- \varepsilon_1, j_2 - \varepsilon_2 \ldots j_r - \varepsilon_r)+ \rho(\varepsilon_1 x_{j1},\ldots, \varepsilon_r x_{jr})\}$$

où ρ est la fonction coût et $\varepsilon = (\varepsilon 1, \varepsilon 2, \ldots \varepsilon n) \in \{0,1\}^n$ est un vecteur qui indique les directions du processus d'alignement dans l'hypercube.

Complexité : La taille de l'hypercube est $O(\Pi^r_{j=1} n_j)$ ou n_j est la longueur de x_j et ou le calcule de chaque entrée considère 2^r-1 autres entrées.

Si n1=n2=…=nr=n alors la complexité en espace est de $O(n^r)$ et la complexité en temps est de **$O(2^r n^r) \cdot O(\text{calcul de la fonction } \rho)$.** Cela signifie que la recherche d'une solution exacte, en utilisant la programmation dynamique, est possible seulement quand l'on a un nombre très petit de séquences.

Le problème est **NP-complet**.

Inférence d'un alignement multiple a partir d'un alignement par paires

- D'un alignement multiple optimale, nous pouvons inférer des alignements par paires entre toutes paires de séquences, mais ils ne sont pas nécessairement optimaux.
- Il est difficile d'inférer un "bon" alignement multiple a partir d'un alignement de séquences par paires entre toutes les séquences.

Alignment multiple induit alignments par paires

Chaque alignement multiple induit un alignement par paires

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

Induit:

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

Problème inverse: construction de l'alignement multiple a partir de l'alignement par paires

Etant donne l'alignement de 3 séquences arbitraires :

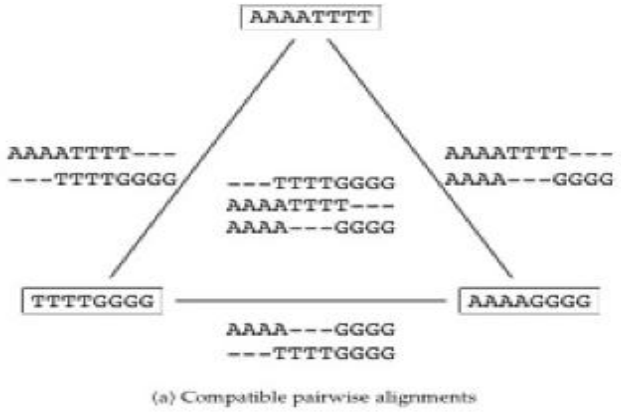
```
x: ACGCTGG-C;   x: AC-GCTGG-C;   y: AC-GC-GAG
y: ACGC--GAC;   z: GCCGCA-GAG;   z: GCCGCAGAG
```

Peut-on reconstruire un alignement multiple que les induits ?

PAS TOUJOURS

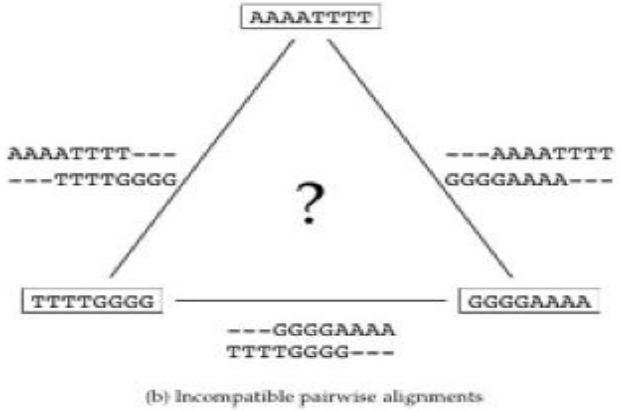
Combinaisons d'alignements par paires optimaux dans un alignement multiple

On peut combiner les alignements par paires dans un alignement multiple



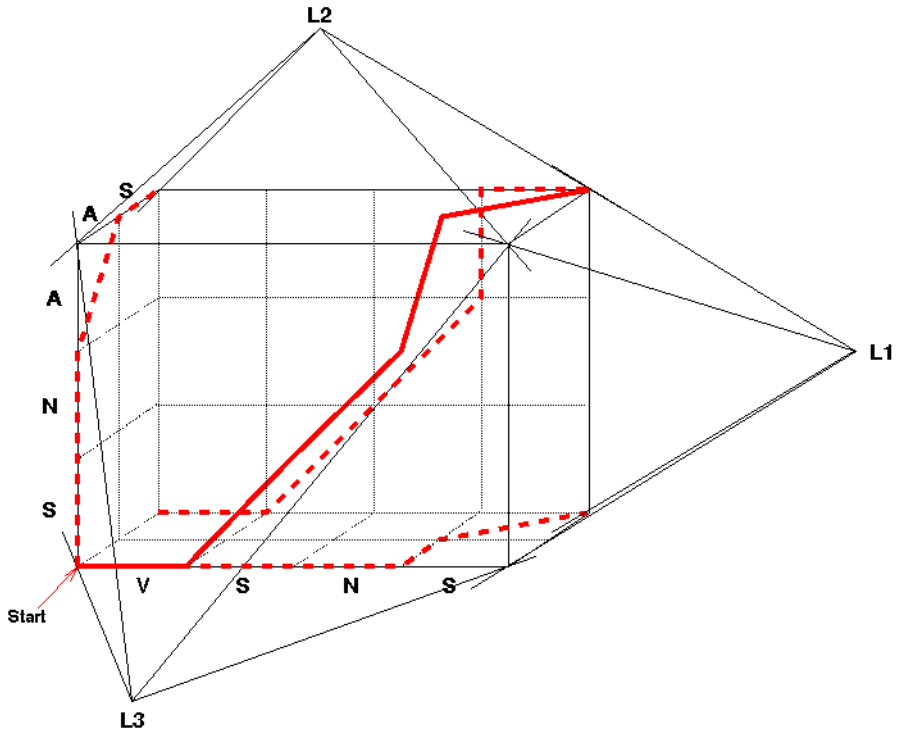
(a) Compatible pairwise alignments

On ne peut **pas** combiner les alignements par paires dans un alignement multiple



(b) Incompatible pairwise alignments

Projection de l'alignement multiple



Un alignement 3-D peut être projeté en 2-D pour représenter un alignement entre paires de séquences.

Toues les 3 projections par paires de l'alignement multiple

Entre les méthodes d'alignement multiple les plus communes on présentera:

+ Alignement d'une séquence à un profile
+ Alignement progressif : ClustalW
+ Alignement probabiliste : ProbCons

Alignement d'une séquence à un profile

Définition: pour un alignement S' de longueur l, un **profile** est une matrice $|\Sigma \cup \{-\}|$ x l ou les colonnes sont des vecteurs de probabilité qui dénotent la fréquence de chaque symbole dans la colonne d'alignement correspondante.

Alignement multiple

A B A
A B –
- B A
C A –

Fréquence des lettres par colonne

	C1	C2	C3
A	50%	25%	50%
B	0%	75%	0%
C	25%	0%	0%
-	25%	0%	50%

A la place des fréquences on peut utiliser $\log(p_i(a))/p(a)$, ou $p_i(a)$ est la fraction de a dans la colonne i et p(a) est la fraction de a dans toutes les colonnes.

Representation d'un profile d'alignement multiple

```
        -  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          1                 1        .8
C     .6             1           .4 1     .6 .2
G           1  .2                    .2        .4  1
T     .2                   1     .6                .2
-     .2        .8                          .4 .8 .4
```

Représentation d'un profile d'alignement multiple

```
  -  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     1                 1        .8
C  .6          1           .4  1     .6 .2
G           1  .2                    .2        .4  1
T  .2                   1     .6                .2
-  .2           .8                       .4  .8 .4
```

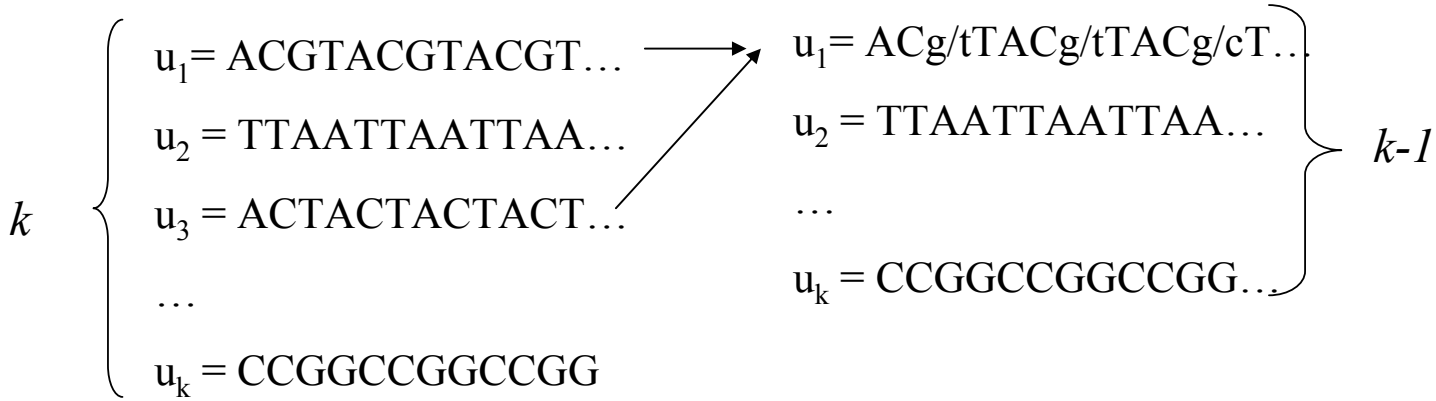
Nous avons aligner une **séquence contre une séquence**

Peut-on aligner une **séquence contre un profile?**

Peut-on aligner un **profile contre un profile?**

Alignement multiple: approche gourmand

- Choisir la paire la plus proche de sequences et combiner les dans un profile, en reduisant l'alignement de *k* sequences a un alignement de *k-1* sequences/profiles. **Repeter**
- Il s'agit d'une methode gourmande heuristique



Méthode gourmande : exemple

- Considérer ces 4 séquences

s1 GATTCA
s2 GTCTGA
s3 GATATT
s4 GTCAGC

Approache gourmand : exemple (cont'd)

- Il y a $\binom{4}{2}$ = 6 possibles alignements

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

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

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

Approache gourmand : exemple (cont'd)

s_2 *and* s_4 are closest; combine:

s2 GTCTGA
s4 GTCAGC

} $s_{2,4}$ (profile) GTCt/aGa/cA

new set of 3 sequences:

s_1 GATTCA
s_3 GATATT
$s_{2,4}$ GTCt/aGa/c

An alignement entre une séquence B et un profile P (ou les deux ont la même longueur) peut être évalué comme $\Sigma_{j=1}^{m}\sigma(p_j,b_j)$. On peut utiliser la programmation dynamique pour trouver le meilleur alignement de la séquence pour le profile.

Dans l'alignement entre paires de séquences l'étape principale consiste dans la définition du score entre deux positions x et y: $\sigma(x,y)$. Pour une lettre x et une colonne y d'un profile, on prend $\sigma(x,y)$ égale a la probabilité que x soit dans la colonne y. Cette valeur dépend de la fréquence des occurrence de x dans la colonne y. Le score de $\sigma(x,-)$ doit être aussi défini.

Average Profile Score (total) = -5
Average Profile Score (no gaps) = -6

Cons	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
E	-2	0	-9	1	5	-10	-8	-5	-8	0	1	-7	-3	-1	0	-5	2	-2
D	-2	0	-9	1	1	-10	-2	-4	-9	0	-1	-10	-6	-1	0	-1	0	-3
E	-2	0	-15	3	11	-16	-10	-5	-14	0	1	-14	-9	-2	0	-2	3	-3
G	-2	-2	-9	-1	0	-8	-1	-3	-9	0	-4	-9	-6	-1	0	-3	-2	-6
E	-9	7	-26	13	26	-24	-15	-4	-25	0	7	-24	-15	0	0	-4	9	-3
E	-8	3	-24	7	23	-20	-12	-1	-24	0	3	-22	-13	0	0	-10	12	0
E	-7	-6	-17	-2	8	-8	-15	-7	-6	0	-5	-10	-6	-9	0	-12	-1	-11
Y	-15	-10	-15	-17	-11	18	-20	6	-10	0	-11	-8	-6	-13	0	-20	-6	-12
V	2	-11	-10	-8	1	-12	-14	-11	2	0	-6	-4	-2	-11	0	-10	-4	-11
V	4	-32	-8	-29	-21	-11	-27	-27	25	0	-20	9	6	-29	0	-15	-20	-28
E	-16	16	-52	28	75	-43	-28	0	-48	0	15	-48	-30	0	0	-13	30	0
K	-2	-4	-14	-9	3	-20	-12	-5	-21	0	19	-15	-8	-2	0	-9	4	14
T	-9	-36	-10	-32	-31	-3	-41	-30	42	0	-30	21	10	-31	0	-28	-30	-32
L	-8	-28	-6	-25	-20	-3	-29	-20	17	0	-16	21	11	-21	0	-21	-15	-16
D	-4	13	-13	21	8	-22	-8	-9	-22	0	-4	-26	-17	0	0	-5	0	-11
H	-3	-2	-7	-4	0	-8	-8	4	-11	0	2	-10	-6	0	0	-7	0	1
R	-7	-8	-18	-12	1	-17	-16	-4	-18	0	13	-12	-5	-3	0	-13	5	18

Profile pour les chromo – domaines classiques

Alignement d'alignements

- Etant donnes deux alignements, peut-on les aligner entre eux ?

```
x GGGCACTGCAT
y GGTTACGTC--    Alignement 1
z GGGAACTGCAG

w GGACGTACC--    Alignement 2
v GGACCT-----
```

Alignement d'alignements

- Nous pouvons utiliser l'alignement des profiles correspondants

```
x GGGCACTGCAT
y GGTTACGTC--     Alignement combiné
z GGGAACTGCAG
w GGACGTACC--
v GGACCT-----
```

Alignement progressif

- *L'alignement progressif* est une variation de l'algorithme gourmand munie d'une stratégie de choix de l'ordre de l'alignement plus intelligente.
- L'alignement progressif marche bien pour des séquences proches, mais empire pour des séquences
  - Les gaps dans les séquences consensus sont permanent
  - Utilise les profiles pour comparer les séquences

ClustalW

- Alignement multiple le plus utilise aujourd'hui
- 'W' signifie 'weighted' (parties différentes de l'alignement sont pesées différemment).

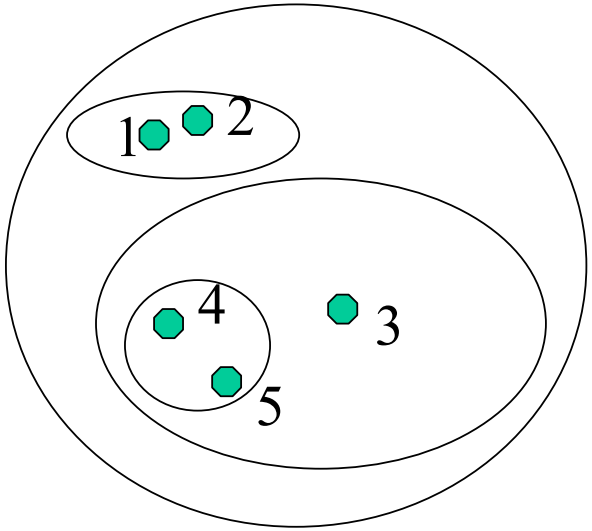
Alignement progressif

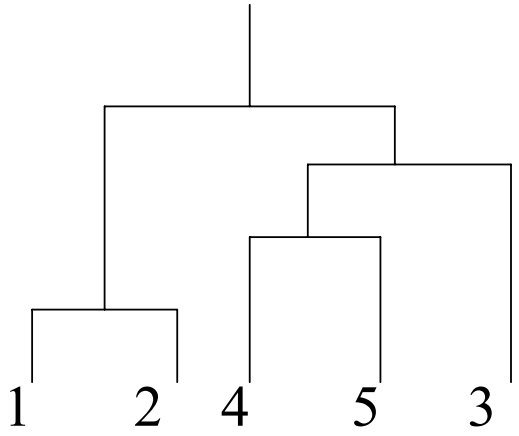
Idée: la paire de séquences de distance minimale a été obtenue, avec forte probabilité, d'une paire de séquences qui ont divergés le plus récemment.

L'algorithme de Feng-Doolittle (1987):

1. Calculer les $\binom{k}{2}$ scores des alignement par paires, et les convertir dans des distances.
2. Utilise un algorithme de clustering incrémentale pour construire un arbre a partir des distances.
3. Traverser les nœuds dans l'ordre de construction de l'arbre, en alignant progressivement les séquences. Noter que la paire la plus similaire est alignée par première, et elle sera suivie par la deuxième plus similaire et ainsi de suite.

Exemple de clustering incrémentale





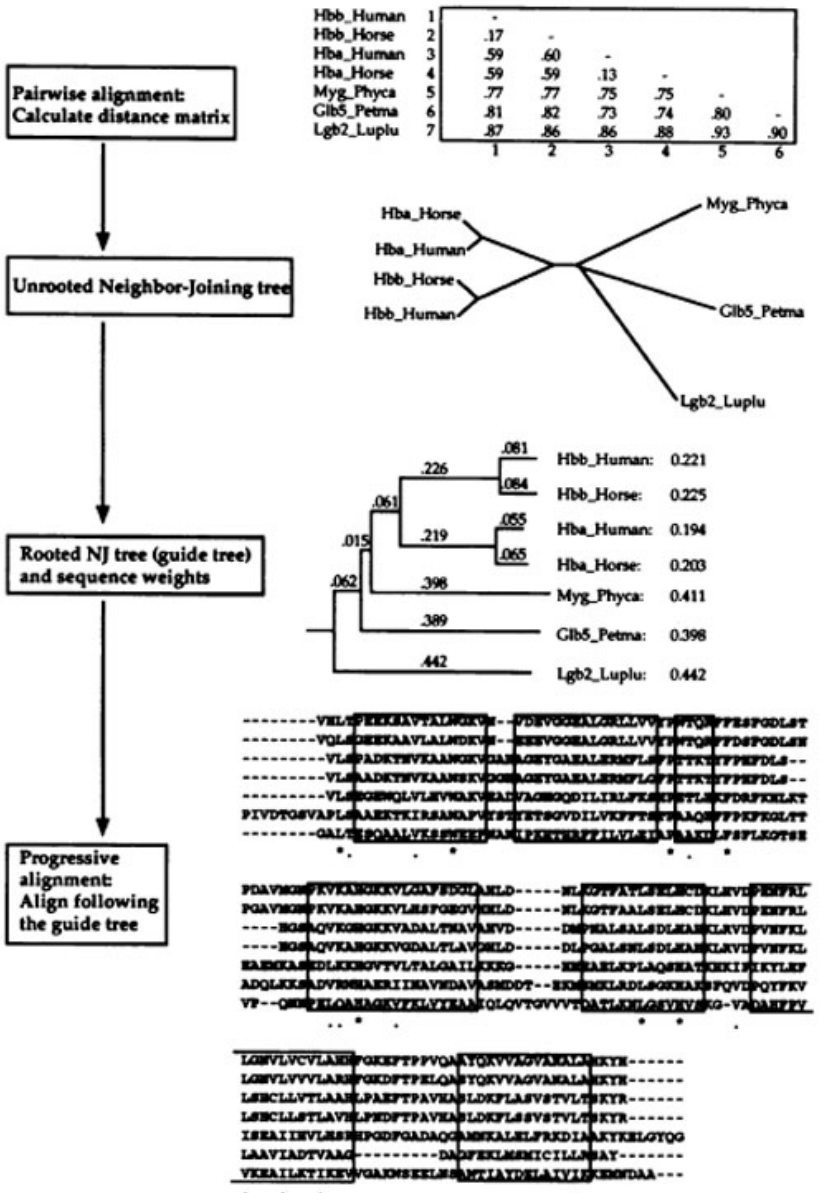
ClustalW

L'algorithme de Thompson, Higgins, Gibson (1994):

1. Calculer les $\binom{k}{2}$ scores des alignement par paires, et les convertir dans des distances.

2. Utiliser l'algorithme de neighbor-joining pour construire un arbre a partir des distances. Cet arbre a des branches de longueurs différente, et la longueur est proportionnelle a la divergence estimée sur chaque branche.

3. Aligner séquence-séquence, séquence-profile, profile-profile dans un ordre de similarité décroissante représenté par l'arbre.


Pairwise alignment: Calculate distance matrix
Unrooted Neighbor-Joining tree
Rooted NJ tree (guide tree) and sequence weights
Progressive alignment: Align following the guide tree
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
Hba_Horse
Hba_Human
Hbb_Horse
Hbb_Human
Myg_Phyca
Glb5_Petma
Lgb2_Luplu
.081
.226
.084
.061
.055
.219
.015
.065
.062
.398
.389
.442
Hbb_Human: 0.221
Hbb_Horse: 0.225
Hba_Human: 0.194
Hba_Horse: 0.203
Myg_Phyca: 0.411
Glb5_Petma: 0.398
Lgb2_Luplu: 0.442

Etape 1: Alignement par paires

- Aligner chaque séquence contre les autres en donnant une matrice de similarité
- Similarité = appariements exactes / longueur des séquences (pourcentage d' identité)

	v_1	v_2	v_3	v_4
v_1	–			
v_2	.17	–		
v_3	.87	.28	–	
v_4	.59	.33	.62	–

(.17 means 17 % identical)

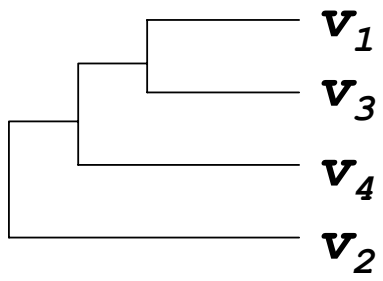
distance = 1 - similarité

Etape 2: “Arbre guide”

- Créer l’arbre guide en utilisant la matrice de similarité
  - ClustalW utilise la méthode neighbor-joining
  - Les arbres guide reflètent grosso modo les relations évolutionnaires

Etape 2: “Arbre guide”

	v_1	v_2	v_3	v_4
v_1	-			
v_2	.17	-		
v_3	.87	.28	-	
v_4	.59	.33	.62	-



Calculer:

$v_{1,3}$ = alignement (v_1, v_3)

$v_{1,3,4}$ = alignement$((v_{1,3}), v_4)$

$v_{1,2,3,4}$ = alignement$((v_{1,3,4}), v_2)$

Etape 3: Alignement progressif

- Commencer par aligner les paires de séquences les plus proches
- En suivant l'arbre guide, ajoute les nouvelles séquences, en alignant aux alignements existants.
- Insérer les gaps si nécessaire

```
FOS_RAT          PEEMSVTS-LDLTGGLPEATTPESEEAFTLPLLNDPEPK-PSLEPVKNISNMELKAEPFD
FOS_MOUSE        PEEMSVAS-LDLTGGLPEASTPESEEAFTLPLLNDPEPK-PSLEPVKSISNVELKAEPFD
FOS_CHICK        SEELAAATALDLG----APSPAAAEEAFALPLMTEAPPAVPPKEPSG--SGLELKAEPFD
FOSB_MOUSE       PGPGPLAEVRDLPG-----STSAKEDGFGWLLPPPPPPP-----------------LPFQ
FOSB_HUMAN       PGPGPLAEVRDLPG-----SAPAKEDGFSWLLPPPPPPP-----------------LPFQ
                 .   . :   ** .     :..  *:.*   *   . *                   **:
```

Points et étoiles montrent le niveau de conservation d'une colonne

Alignements multiples: les fonctions de score

- Nombre d'appariements (score de la sous-séquence plus longue du multiple)
- Score d'entropie
- Sum-of-pairs (SP-score)

Multiple LCS Score

- Une colonne est un “match” si toutes les lettres dans la colonne sont les mêmes

AAA
AAA
AAT
ATC

- Bon score seulement pour des séquences très proches

Entropie

- Définie les fréquences pour l'occurrence de chaque lettre dans chaque colonne de l'alignement multiple
  - $p_A = 1, p_T=p_G=p_C=0$ (1^{re} colonne)
  - $p_A = 0.75, p_T = 0.25, p_G=p_C=0$ (2^{eme} colonne)
  - $p_A = 0.50, p_T = 0.25, p_C=0.25\ p_G=0$ (3^{eme} colonne)

- Calcule l'entropie de chaque colonne

$$-\sum_{X=A,T,G,C} p_X \log p_X$$

AAA
AAA
AAT
ATC

Entropie: exemple

$$entropy\begin{pmatrix} A \\ A \\ A \\ A \end{pmatrix} = 0$$ meilleur cas

pire cas

$$entropy\begin{pmatrix} A \\ T \\ G \\ C \end{pmatrix} = -\sum \frac{1}{4}\log\frac{1}{4} = -4(\frac{1}{4} * -2) = 2$$

Alignement multiple: Entropy Score

L'entropie d'un alignement multiple est la somme des entropies de chaque colonne :

$$\Sigma_{\text{sur toutes les colonnes}} \; \Sigma_{X=A,T,G,C} \; p_X \log p_X$$

Entropie d'un alignement: exemple

entropie d'alignement d'une colonne:

$$-(p_A \log p_A + p_C \log p_C + p_G \log p_G + p_T \log p_T)$$

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

- Colonne 1 = -[1*log(1) + 0*log0 + 0*log0 +0*log0]
  = 0
- Colonne 2 = -[($^1/_4$)*log($^1/_4$) + ($^3/_4$)*log($^3/_4$) + 0*log0 + 0*log0]
  = -[($^1/_4$)*(-2) + ($^3/_4$)*(-.415)] = +0.811
- Colonne 3 = -[($^1/_4$)*log($^1/_4$)+($^1/_4$)*log($^1/_4$)+($^1/_4$)*log($^1/_4$) +($^1/_4$)*log($^1/_4$)]
  = 4* -[($^1/_4$)*(-2)] = +2.0
- Entropie d'alignement = 0 + 0.811 + 2.0 = +2.811

Sum of Pairs Score (SP-Score)

- Considerer l'alignement par paires des séquences

  a_i and a_j

  induit par l'alignement multiple de k séquences

- Dénoter le score de l'alignement par paires sous-optimale (pas forcement optimale) comme

  $s^*(a_i, a_j)$

- Sommer les scores par paires pour l'alignement multiple:

  $s(a_1,\ldots,a_k) = \Sigma_{i,j}\, s^*(a_i, a_j)$

Calcule de l'SP-Score

Aligner 4 sequences: 6 alignements par paires

Etant donnees a_1,a_2,a_3,a_4:

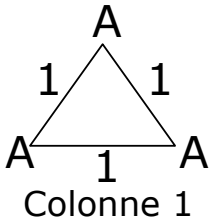
$$s(a_1 \ldots a_4) = \Sigma s^*(a_i, a_j) = s^*(a_1, a_2) + s^*(a_1, a_3) + s^*(a_1, a_4) + s^*(a_2, a_3) + s^*(a_2, a_4) + s^*(a_3, a_4)$$

SP-Score: exemple

a_1 ATG–C–AAT
. A–G–CATAT
a_k ATCCCATTT

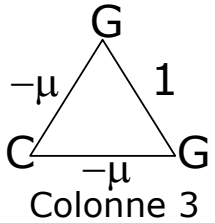
Pour calculer chaque colonne :

$$s'(a_1 \ldots a_k) = \sum_{i,j} s^*(a_i, a_j) \longleftarrow \binom{n}{2} \text{ Pairs of Sequences}$$



Colonne 1

Score=3



Colonne 3

Score = 1 – 2μ

La mesure somme-des-paires, qui combine la projection des alignements de paires pour toutes les paires dans l'alignement multiple, a été largement utilisée dans les méthodes d'alignement progressif. Mais ces méthodes d'alignement sont prônes a une **accumulation d'erreurs** des les premières étapes d'alignement. Pour surmonter ce problème, des étapes de post-processing comme le raffinement itératif sont d'habitude appliquées.

Raffinement itératif: partage de façon aléatoire les séquences dans l'alignement multiple courant en deux groupes et les re-aligner. Répéter itérativement cette étapes 100 fois.

Alignement basé sur la consistance

La prévention est le meilleur médicament

Pour chaque alignement multiple, les alignements par paires induits sont nécessairement consistant, cad que étant donné un alignement multiple de x,y,z, si la position x_i aligne avec la position z_k et z_k aligne avec y_j dans les projections x-z et z-y des alignements en multiple, alors x_i doit aligner avec y_j dans l'alignement x-y projeté.

Les alignements basés sur la consistance appliquent ce principe mais renversé, en utilisant des alignements a des séquences intermédiaires comme évidence pour guider l'alignement des paires x et y.

L'outil PROBCONS

- Disponible sur le Web et en local
- Input :
  - Plusieurs protéines au format MFA dans un seul fichier
  - Exemple , 5 protéines:

  - >plas_horvu
  - DVLLGANGGVLVFEPNDFSVKAGETITFKNNAGYPHNVVFDEDAVPSGVDVSKISQEEYL
  - TAPGETFSVTLTVPGTYGFYCEPHAGAGMVGKVTV
  - >plas_chlre
  - VKLGADSGALEFVPKTLTIKSGETVNFVNNAGFPHNIVFDEDAIPSGVNADAISRDDYLN
  - APGETYSVKLTAAGEYGYYCEPHQGAGMVGKIIV
  - >plas_anava
  - VKLGSDKGLLVFEPAKLTIKPGDTVEFLNNKVPPHNVVFDAALNPAKSADLAKSLSHKQL
  - LMSPGQSTSTTFPADAPAGEYTFYCEPHRGAGMVGKITV
  - >plas_proho
  - VQIKMGTDKYAPLYEPKALSISAGDTVEFVMNKVGPHNVIFDKVPAGESAPALSNTKLRI
  - APGSFYSVTLGTPGTYSFYCTPHRGAGMVGTITV
  - >azup_achcy
  - VHMLNKGKDGAMVFEPASLKVAPGDTVTFIPTDKGHNVETIKGMIPDGAEAFKSKINENY
  - KVTFTAPGVYGVKCTPHYGMGMVGVVEV

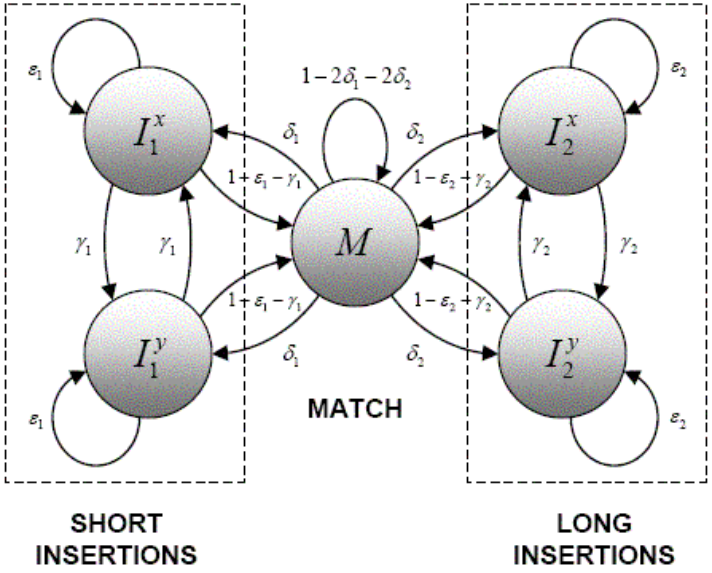
- Output :
  - Un fichier MFA ou CLUSTALW contenant le résultat de l'alignement
  - MFA:
  - >plas_horvu
  - -DVLLGANGGVLVFEPNDFSVKAGETITFKNNAGYPHNVVFDEDAVPS--GVDVSKISQE
  - EYLTAPGETFSVTLTV---PGTYGFYCEPHAGAGMVGKVTV
  - >plas_chlre
  - --VKLGADSGALEFVPKTLTIKSGETVNFVNNAGFPHNIVFDEDAIPS--GVNADAISRD
  - DYLNAPGETYSVKLTA---AGEYGYYCEPHQGAGMVGKIIV
  - >plas_anava
  - --VKLGSDKGLLVFEPAKLTIKPGDTVEFLNNKVPPHNVVFDAALNPAKSADLAKSLSHK
  - QLLMSPGQSTSTTFPADAPAGEYTFYCEPHRGAGMVGKITV
  - >plas_proho
  - VQIKMGTDKYAPLYEPKALSISAGDTVEFVMNKVGPHNVIFDK--VPA--GESAPALSNT
  - KLRIAPGSFYSVTLGT---PGTYSFYCTPHRGAGMVGTITV
  - >azup_achcy
  - VHMLNKGKDGAMVFEPASLKVAPGDTVTFIPTDK-GHNVETIKGMIPD--GAEA------
  - -FKSKINENYKVTFTA---PGVYGVKCTPHYGMGMVGVVEV

- Output:
  - Format ClustalW:

```
plas_horvu    -DVLLGANGGVLVFEPNDFSVKAGETITFKNNAGYPHNVVFDEDAVPS--GVDVSKISQE
plas_chlre    --VKLGADSGALEFVPKTLTIKSGETVNFVNNAGFPHNIVFDEDAIPS--GVNADAISRD
plas_anava    --VKLGSDKGLLVFEPAKLTIKPGDTVEFLNNKVPPHNVVFDAALNPAKSADLAKSLSHK
plas_proho    VQIKMGTDKYAPLYEPKALSISAGDTVEFVMNKVGPHNVIFDK--VPA--GESAPALSNT
azup_achcy    VHMLNKGKDGAMVFEPASLKVAPGDTVTFIPTDK-GHNVETIKGMIPD--GAEA------
              *.:    .     : *  :.: .*:*: *  .    **:       *   .  .  :*.

plas_horvu    EYLTAPGETFSVTLTV---PGTYGFYCEPHAGAGMVGKVTV
plas_chlre    DYLNAPGETYSVKLTA---AGEYGYYCEPHQGAGMVGKIIV
plas_anava    QLLMSPGQSTSTTFPADAPAGEYTFYCEPHRGAGMVGKITV
plas_proho    KLRIAPGSFYSVTLGT---PGTYSFYCTPHRGAGMVGTITV
azup_achcy    -FKSKINENYKVTFTA---PGVYGVKCTPHYGMGMVGVVEV
              .     ..  ...: .   .* *   * ** * **** : *
```

PROBCONS

1. (Alignement initiale) Pour chaque paire de séquences *x* et *y*,
   a. Calcule la table des probabilités a posteriori a l'aide des HMM (spécifiés dans le diagramme), contenant les probabilités a posteriori $P(xi \sim yj \mid x, y)$ pour l'appariement de chaque lettre *xi* d'une séquence avec chaque lettre *yj* de l'autre.
   b. Calcule la fiabilité attendue de l'alignement, $E(x, y)$, définie comme la somme des probabilités d'appariement a posteriori le long du chemin ayant la plus grande somme divisée par la plus petite longueur.



2. (Transformation de consistance) mettre a jours simultanément toutes les matrices de probabilités a posteriori en utilisant la transformation:

$$P'(x_i \sim y_j \mid x,y) = \frac{1}{N} \sum_{z \in S} \sum_{k} P(x_i \sim z_k \mid x,z) P(z_k \sim y_j \mid z,y)$$

Répéter cette étape pour 2 itérations.

3. (Arbre guide dans l'alignement) Etant donne les valeurs de fiabilité attendue pour chaque alignement par paires, calculer un arbre guide *T* a l'aide de l'algorithme gourmand de clustering hiérarchique :
   a. Initialement, placer chaque séquence dans son propre cluster.
   b. Fusionner les clusters *x* et *y* avec la valeur de fiabilité attendue maximale. Quand le nouveau cluster *xy* est formé, définir sa valeur de fiabilité attendue avec tout autre cluster *z* comme *E*(*x*, *y*)(*E*(*x*, *z*) + *E*(*y*, *z*)) / 2.
   c. Répéter b jusqu'à quand un seul cluster est obtenu.

4. (Alignement progressif) Réaliser de l'alignement progressif multiple a l'aide de l'arbre guide *T* et en utilisant comme fonction objectif la somme-des-paires, défini comme la somme des probabilités *P*(*xi* ~ *yj* | *x*, *y*) re-estimées pour tous les paires de résidus alignés; comme a l'étape 2, pénalité d'insertion ne sont pas utilisées pour le calcul du chemin ayant somme maximale.

5. (Raffinement itératif) Partager de façon aléatoire les séquences dans l'alignement multiple courant en deux groupes et les re-aligner. Répéter itérativement cette étapes 100 fois.

Méthodes de test

- Bases de données utilisées:
  - BAliBASE 2.01 (Thompson et al. 1999a)
    - 141 alignements de protéines
    - Tests effectués sur les *core blocks,* régions pour lesquelles on connaît les alignements
  - PREFAB 3.0 (Edgar 2004),
    - Base générée automatiquement
    - 1932 alignements sur 49 protéines de 250 a.a
  - SABmark 1.63 (Van Walle et al. 2004), 2 sets:
    - « twilight zone » : séquences de moins de 25% d'identité
    - « superfamily » : séquences de moins de 50% d'identité
- Machine de test:
  - 3.3-Ghz Pentium IV avec 2 Gb Ram

4. Les méthodes de test

- Critères de score d'alignement
  - BaliBASE
    - SP = sum-of-pairs =
      - nb de paires de résidus correctement alignées / nb total de paires de résidus du *core block*
    - CS = columns-score =
      - nb de colonnes correctement alignées / nb total de colonnes du *core block*
  - PREFAB
    - Q = comme SP
  - SABmark
    - Fd = comme SP

5. Comparaison avec d'autres algorithmes d'alignement de séquences

Table 1. Performance of aligners on the BAliBASE benchmark alignments database

Aligner	Ref 1 (82)		Ref 2 (23)		Ref 3 (12)		Ref 4 (12)		Ref 5 (12)		Overall (141)		Time (mm:ss)
	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	
Align-m	76.6	n/a	88.4	n/a	68.4	n/a	91.1	n/a	91.7	n/a	80.4	n/a	19:25
DIALIGN	81.1	70.9	89.3	35.9	68.4	34.4	89.7	76.2	94.0	84.3	83.2	63.7	2:53
CLUSTALW	86.1	77.3	93.2	56.8	75.3	46.0	83.4	52.2	85.9	63.8	86.1	68.0	1:07
MAFFT	86.7	78.1	92.4	50.2	78.8	50.4	91.6	72.7	96.3	85.9	88.2	71.4	1:18
T-Coffee	86.6	77.4	93.4	56.1	78.5	48.7	91.8	73.0	95.8	90.3	88.3	72.2	21:31
MUSCLE	88.7	80.8	93.5	56.3	82.5	56.4	87.6	60.9	96.8	90.2	89.6	73.9	**1:05**
ProbCons	**90.1**	**82.6**	**94.4**	**61.3**	84.1	**61.3**	90.1	72.3	97.9	91.9	91.0	77.2	5:32
ProbCons-ext	90.0	82.5	94.2	59.1	**84.3**	61.1	**93.8**	**81.0**	**98.1**	**92.2**	**91.2**	**77.6**	8:02

On constate que

- ProbCons et sa version étendue ont les meilleurs scores
- Mais ne sont pas très rapide

5. Comparaison avec d'autres algorithmes d'alignement de séquences

Table 3. Performance of aligners on the PREFAB protein reference alignment benchmark

Aligner	Overall (1927)	Time
DIALIGN	57.2	12 h, 25 min
CLUSTALW	58.9	2 h, 57 min
T-Coffee	63.6	144 h, 51 min
MUSCLE	64.8	3 h, 11 min
MAFFT	64.8	**2 h, 36 min**
ProbCons	66.9	19 h, 41 min
ProbCons-ext	**68.0**	37 h, 46 min

- Les entrées montrent la moyenne Q de score atteint sur les 1927 alignements de la base PREFAB
- Même bilan que précédemment

5. Comparaison avec d'autres algorithmes d'alignement de séquences

Table 5. Performance of aligners on the SABmark sequence and structure alignment benchmark

Aligner	Superfamily (462)		Twilight zone (236)		Overall (698)		Time (mm:ss)
	f_D	f_M	f_D	f_M	f_D	f_M	
Align-m	44.4	**58.9**	17.1	**43.0**	35.2	**53.5**	56:44
DIALIGN	50.3	42.5	22.5	19.2	41.0	34.6	8:28
CLUSTALW	53.7	38.7	24.8	15.2	43.9	30.8	**2:16**
MAFFT	54.1	40.0	24.8	16.0	44.2	31.9	7:33
T-Coffee	55.4	41.8	26.4	18.0	45.6	33.7	59:10
MUSCLE	55.9	40.1	27.6	17.5	46.4	33.0	20:42
ProbCons	**59.9**	45.0	**32.1**	21.7	**50.5**	37.1	17:20
ProbCons-ext	**59.9**	45.3	32.0	22.1	**50.5**	37.5	23:10

- Nettes améliorations des scores sur les protéines à faible et fort taux d'identité de la base SABmark

Références bibliographiques

D.S. Hirschberg, Algorithms for the longest common subsequence problem, *J. ACM*, 24:664-675, 1977.

D. Gusfield, *Algorithms on strings, Trees and Sequences*, Cambridge University Press, 1997.

FASTA algorithm:

D. Lipman, W. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435-1441, 1985

D. Lipman, W. Pearson, Improved tools for biological sequence comparison. *PNAS USA*, 85:2444-2448, 1988.

BLAST algorithm:

S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J.Lipman. Basic local alignment search tool, *Journal Mol Biol*, 215:403-410, 1990

S.F. Altschul et al. Gapped BLAST and Psi-Blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389-3402, 1997.

Matrices d'alignement :

M. Dayhoff, R. Schwartz. Matrices for detecting distant relationship. *Atlas of Protein Sequences*, 353-358,1979.

S. Henikoff, J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences USA*, 89:10915-10919, 1992.

G.Gonnet, M.A.Cohen, S.A.Benner, Exhaustive matching of the entire protein sequence database. *Science*, 256, 1433-1445.

Tests d'evaluation des alignements

C.Lambert, J.M. vanCampenhout, X.DeBolle, E.Depiereux, Review of common sequence alignment methods: clues to enhance reliability. *Current Genomics*, 4, 131-146, 2003.

Alignement multiple

H. Carrillo and D. Lipmann. The multiple sequence alignment problem in biology. *SIAM J. Appl. Math*, 48:1073–1082, 1988.

D. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 25:351–360, 1987.

W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *science*, 15:279–284, 1967.

T. Jiang, L. Wang, and E. L. Lawler. Approximation algorithms for tree alignment with a given phylogeny. *Algorithmica*, 16:302–315, 1996.

D. J. Lipman, S. Altshul, and J. Kececiogly. A tool for multiple sequence alignment. *Proc. Natl. Academy Science*, 86:4412–4415, 1989.

J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specificgap penalties and weight matrix choice. *Nucleic Acids Res*, 22:4673–80, 1994.

L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Computational Biology*, 1:337–348, 1994.

Do, C.B., Mahabhashyam, M.S.P., Brudno, M., and Batzoglou, S. 2005. PROBCONS: Probabilistic Consistency-based Multiple Sequence Alignment. *Genome Research* 15: 330-340.

Bibliographie PROBCONS et HMM

- ProbCons paper :
  - Do, C.B., Mahabhashyam, M.S.P., Brudno, M., and Batzoglou, S. 2005. PROBCONS: Probabilistic Consistency-based Multiple Sequence Alignment. *Genome Research* 15: 330-340.
- Plus sur les HMMs :
  - Transparents du cours ASB 2005-06
    - http://www.ihes.fr/~carbone/L4_ABS_Recherche_Motifs_HMM.pdf
  - Fabien Campillo :
    - http://www.lirmm.fr/%7Erivals/DEA/DOC/HMMBio.pdf
  - Laurent Bréhelin :
    - http://www.lirmm.fr/%7Erivals/DEA/DOC/HMM-Brehelin-tutorial.pdf
- ProbCons l'outil :
  - http://probcons.stanford.edu/
- Bases de données: MUSCLE: multiple sequence alignment with high accuracy and high throughput :
  - http://nar.oxfordjournals.org/cgi/content/full/32/5/1792?ijkey=48Nmt1tta0fMg&keytype=ref

exercises

Exercise: construct a simple symmetric PAM matrix as in Ewens/Grant pp210.

Preuve de complexite pour alignement remonte