Alignements et recherche de similarités

Analyse de séquences génomiques et phylogénie

Guy Perrière

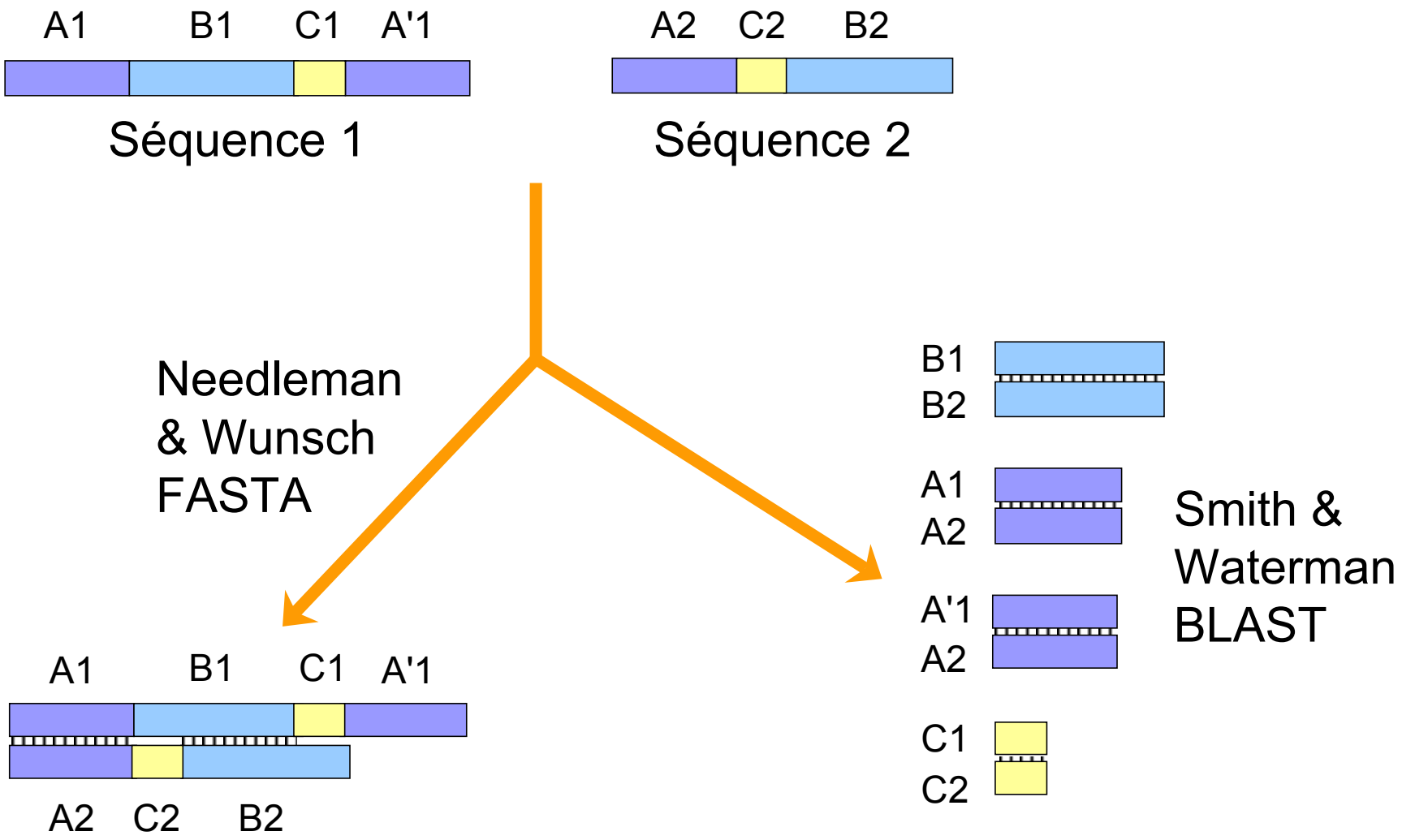
Laboratoire de Biométrie et Biologie Évolutive
UMR CNRS n° 5558
Université Claude Bernard – Lyon 1

3-6 avril 2017

Objectifs poursuivis

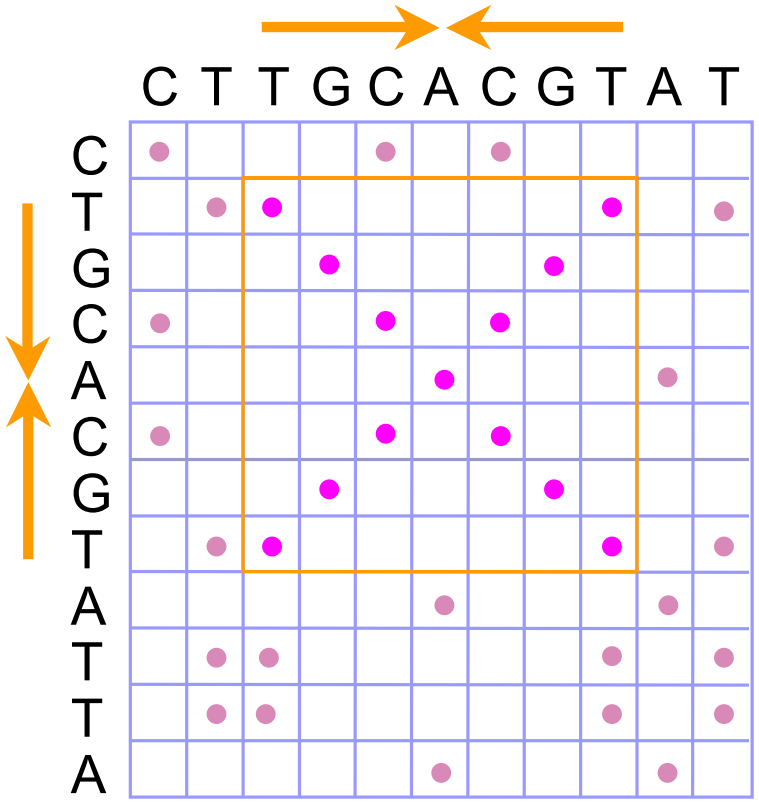
- Identification d'homologues.
- Recherche de contraintes fonctionnelles.
- Prédiction de structure (ARN, protéine).
- Prédiction de fonction.
- Reconstitution des relations évolutives entre séquences (phylogénie).
- Assemblage de lectures (séquençage).

Alignement global et local



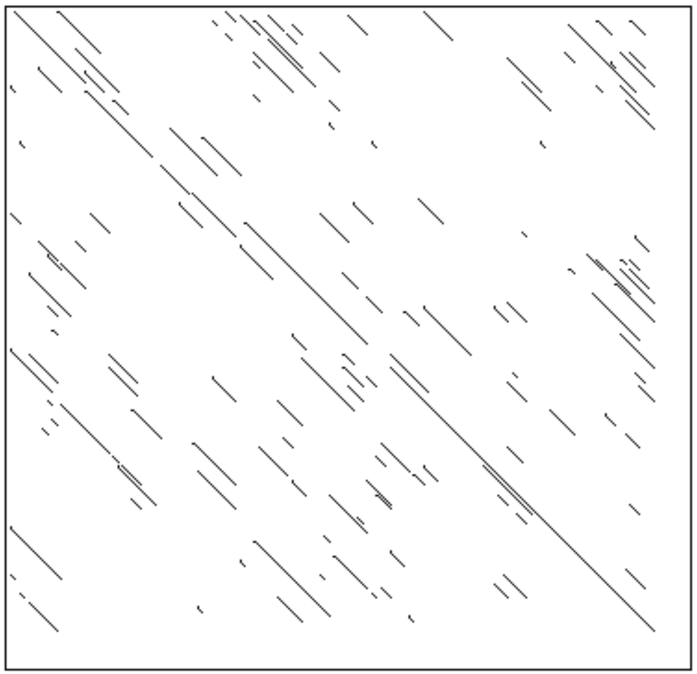
Matrices de points

- Comparaison visuelle de deux séquences :
  - Une diagonale indique une similarité locale.
  - Une croix indique une répétition inverse.
  - Méthode simple et rapide :
    - Algorithme en $O(nm)$.
  - Pas d'alignement ni de score global.

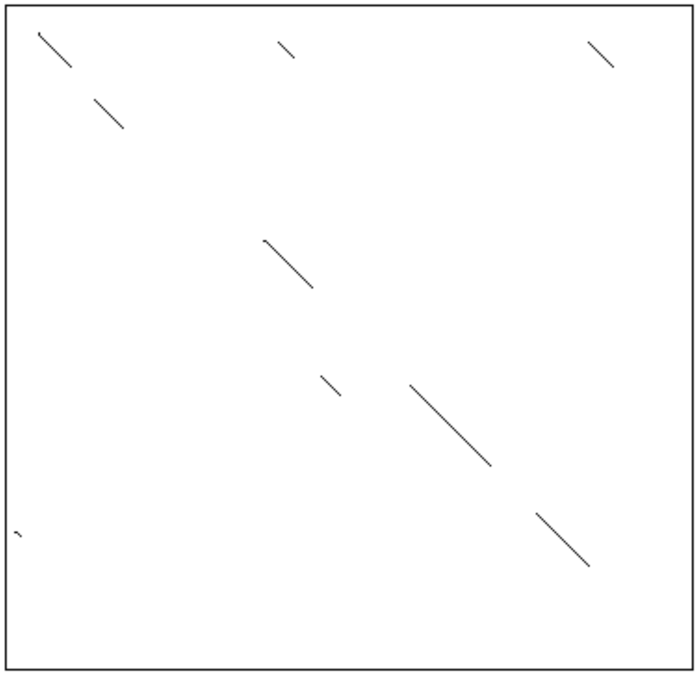


Élimination du bruit de fond

- Filtrage en affichant un point uniquement si plusieurs résidus successifs correspondent :
  - Exemple des hémoglobines α et β humaines :



Identités = 3/10



Identités = 5/10

Représentation

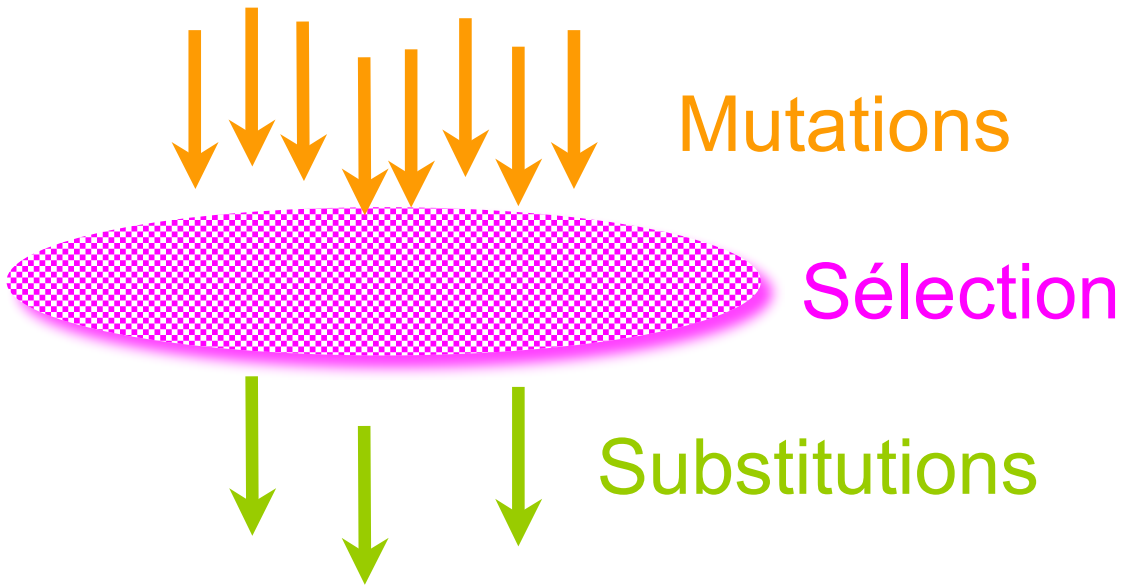
- Les résidus (nucléotides, acides aminés) sont superposés de façon à maximiser les identités entre les séquences :

```
G T T A A G G C G - G G A A A
G T T - - - G C G A G G A C A
* * *       * * *   * * *   *
```

- Il existe deux sortes de différences :
  - Substitutions (*mismatches*).
  - Insertions et délétions (*indels* ou *gaps*).

Mutations et substitutions

- Les différences observées dans un alignement correspondent aux substitutions :
  - Mutations ayant passé le filtre de la sélection :
    - Mutations neutres (*i.e.*, sans effet sur le phénotype) ou avantageuses du point de vue sélectif.



Quel est le bon alignement ?

```
G T T A C G A
G T T - G G A
* * *     * *
```

ou

```
G T T A C - G A
G T T - - G G A
* * *       * *
```

```
G T T A C G A
G T T G - G A
* * *     * *
```

- Pour le biologiste, le bon alignement est celui qui représente le scénario évolutif le plus probable :
  - Définition d'une *fonction de score* appropriée.
- Autres choix possibles (*e.g.*, assemblage de lectures).

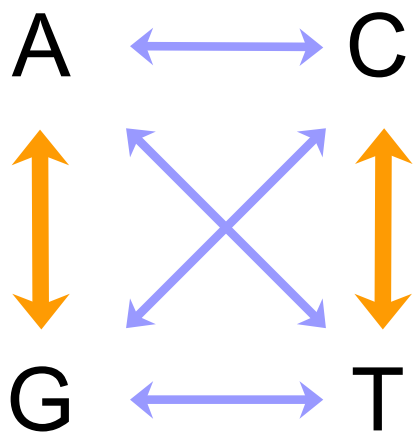
Fonction de score simple

```
G T T A A G G C G - G G A A A
G T T - - - G C G A G G A C A
* * *       * * *   * * *   *
```

$$\text{Score} = \sum \text{Identités} - \sum \text{Différences}$$

$$\left.\begin{array}{ll} \text{Identité} & = +1 \\ \text{Substitution} & = 0 \\ \text{Gap} & = -1 \end{array}\right\} \quad \Longrightarrow \quad \text{Score} = 10 - 4 = 6$$

Modèle d'évolution



$$\mathbb{P}(\text{transition}) > \mathbb{P}(\text{transversion})$$

```
G T T A C G A          G T T A C G A
G T T G - G A    >     G T T - G G A
* * * :   * *          * * *   * *
```

Matrice de substitution ADN

	A	C	G	T
A	1			
C	0	1		
G	0.5	0	1	
T	0	0.5	0	1

$\delta(\mathrm{A}, \mathrm{A}) = 1.0$
$\delta(\mathrm{A}, \mathrm{G}) = 0.5$
$\delta(\text{-}) = -1.0$

```
G T T A C G A          G T T A C G A
G T T G - G A    >     G T T - G G A
* * * :   * *          * * *   * *
```

Score = 4.5 > Score = 4.0

Le cas des acides aminés

- Plus complexe à modéliser que celui des séquences nucléotidiques :
  - Alphabet de vingt lettres au lieu de quatre.
  - Difficulté d'utiliser directement l'information portée par les séquences codantes :
    - Certaines substitutions peuvent avoir plus ou moins d'effet sur la fonction des protéines :

$$\mathbb{P}(\mathrm{AAU}_{\mathrm{Asn}} \rightarrow \mathrm{GAU}_{\mathrm{Asp}}) > \mathbb{P}(\mathrm{AAU}_{\mathrm{Asn}} \rightarrow \mathrm{CAU}_{\mathrm{His}})$$

  - Utilisation de modèles empiriques :
    - Alignement de séquences homologues avec la matrice identité.
    - Construction d'arbres phylogénétiques sur la base de ces alignements.
    - Construction des matrices sur la base du nombre de substitutions observées (ou inférées à partir des arbres).

BLOSUM (*Blocks Substitution Matrices*)

- Matrices fondées sur des alignements locaux de domaines conservés (Henikoff et Henikoff, 1992) :
  - Utilisation de ≈ 2000 domaines provenant de ≈ 500 familles de protéines.
  - Matrice identité.
  - Alignements sans *gaps*.
- Ensemble de quinze matrices créées à partir de domaines comprenant des séquences $\pm$ divergentes :
  - Toutes les paires ayant servi à construire une matrice BLOSUMk ($30 \leq k \leq 100$) ont une identité $\geq$ à $k\%$.
  - Bien adaptées pour l'alignement de séquences très divergentes.

Construction I

- Calcul de la fréquence *observée* de chaque paire d'acide aminé dans l'alignement :
  - Soit n_{ij} le nombre de paires (i, j) observées, $i, j \in \{\text{A, C, D}, \ldots, \text{W}\}$.
  - Soit w la largeur du bloc considéré et a le nombre de séquences alignées.
  - Dans ce cas, le nombre total de paires possibles q est tel que :

$$q = wa(a-1)/2$$

et f_{ij}, la fréquence observée de chaque paire (i, j) est égale à :

$$f_{ij} = n_{ij}/q$$

Construction II

- Calcul de la fréquence *attendue* de chaque paire :
  - Soit π_i la fréquence de l'acide aminé i dans l'alignement.
  - Dans ce cas, les fréquences attendues g_{ij} de chaque paire (i, j) sont égales à :

$$g_{ij} = 2\pi_i\pi_j \quad (i \neq j)$$
$$g_{ii} = \pi_i^2$$

- Calcul des éléments de la matrice de substitution $\mathbf{\Delta} = (\delta_{ij})$ par :

$$\delta_{ij} = 2\log_2(f_{ij}/g_{ij})$$

avec arrondissement à l'entier le plus proche :

  - $\delta_{ij} > 0 \Rightarrow$ substitution plus fréquente qu'attendue.
  - $\delta_{ij} < 0 \Rightarrow$ substitution moins fréquente qu'attendue.

Exemple simple

$$a = 6 \left\{ \overbrace{\begin{array}{cccc} \mathtt{E} & \mathtt{A} & \mathtt{E} & \mathtt{A} \\ \mathtt{A} & \mathtt{A} & \mathtt{A} & \mathtt{G} \\ \mathtt{A} & \mathtt{A} & \mathtt{G} & \mathtt{G} \\ \mathtt{A} & \mathtt{A} & \mathtt{E} & \mathtt{A} \\ \mathtt{A} & \mathtt{A} & \mathtt{G} & \mathtt{G} \\ \mathtt{A} & \mathtt{A} & \mathtt{E} & \mathtt{G} \end{array}}^{w=4} \right. \Rightarrow q = 60$$

i	π_i
A	14/24
E	4/24
G	6/24

(i, j)	f_{ij}	g_{ij}	δ_{ij}
(A, A)	26/60	196/576	0.70
(A, E)	8/60	112/576	−1.09
(A, G)	10/60	168/576	−1.61
(E, E)	3/60	16/576	1.70
(E, G)	6/60	48/576	0.53
(G, G)	7/60	36/576	1.80

	A	E	G
A	1		
E	−1	2	
G	−2	1	2

BLOSUM62

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

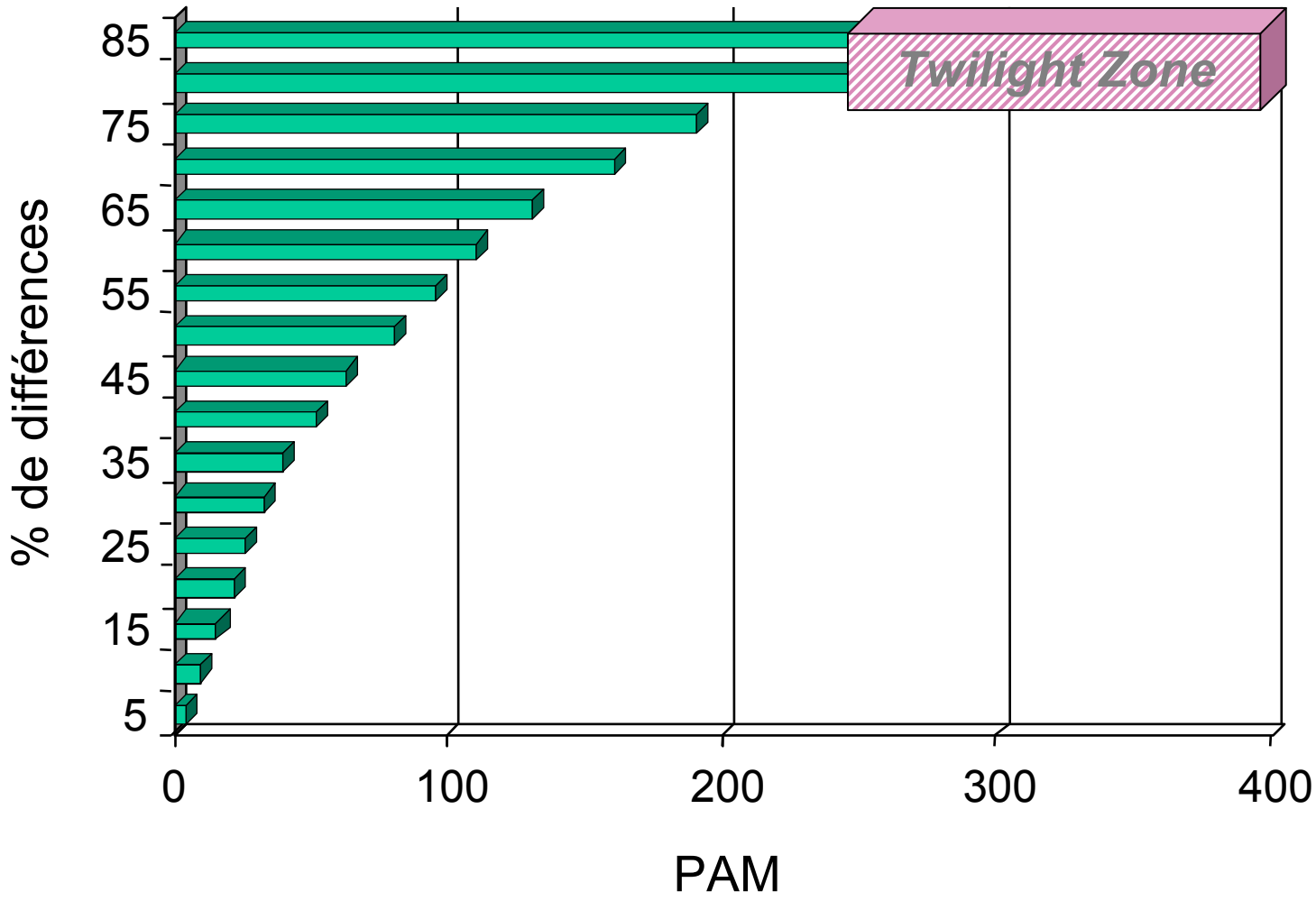
C	sulfhydryle
S	
T	
P	petit hydrophile
A	
G	
N	
D	acide, acide-amide
E	et hydrophile
Q	
H	
R	basique
K	
M	
I	petit hydrophobe
L	
V	
F	
Y	aromatique
W	

Les critères évolutifs recoupent les critères physico-chimiques

PAM (*Point Accepted Mutations*)

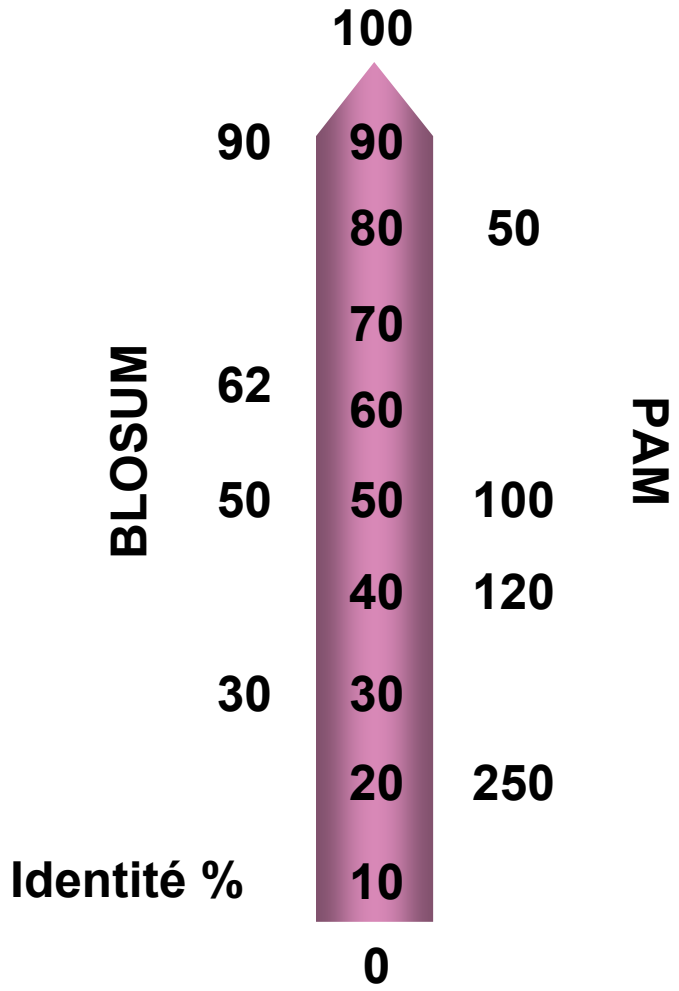
- Matrices construites sous l'hypothèse que les séquences évoluent selon un processus markovien (Dayhoff *et al.*, 1978).
- Alignements globaux de protéines conservées :
  - Utilisation de la matrice identité.
  - Pas de *gaps* dans les alignements.
  - 71 familles de gènes correspondant à 1300 séquences et contenant un total de 1572 substitutions :
    - Identité $\geq 85\%$ entre chaque paire possible au sein d'une famille.
- Matrices JTT (Jones, Taylor et Thornton, 1992) et Gonnet (Gonnet *et al.*, 1992) :
  - Construites avec une méthodologie comparable, mais sur des échantillons bien plus grands.
- Construction : *cf.* cours sur les modèles d'évolution.

Limites de validité



Choix d'une matrice

- Pas de matrice idéale.
- Meilleurs résultats avec les matrices construites avec un plus grand nombre de séquences.
- Choix principalement en fonction du degré de similarité entre les séquences.
- Il est recommandé d'expérimenter !



Pondération des *gaps*

- Avec la fonction de score telle que présentée dans la Diapo 9, la pénalité w associée à un *gap* de longueur k est égale à :

$$w = k\delta(\text{-})$$

avec $\delta(\text{-})$, la pénalité d'un *gap* individuel.

- Amélioration par l'emploi de fonctions affines ou logarithmiques :

$$w = \delta_o(\text{-}) + (k-1)\delta_e(\text{-})$$
$$w = \delta_o(\text{-}) + \ln(k-1)\delta_e(\text{-})$$

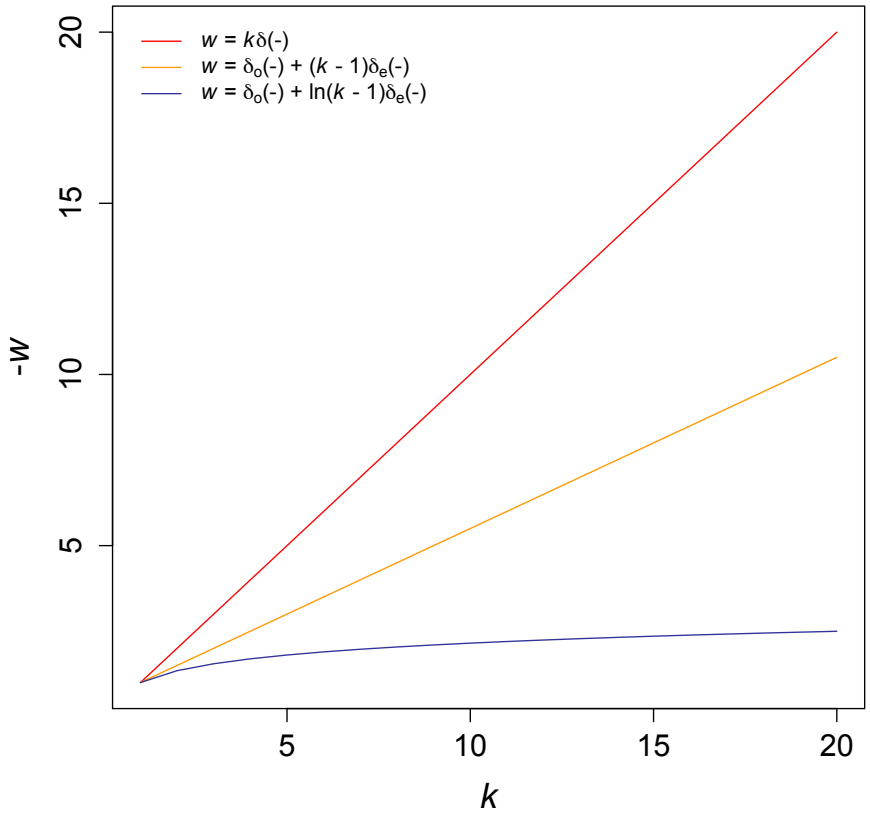
avec $\delta_o(\text{-})$ la pénalité associée à l'ouverture d'un *gap*, et $\delta_e(\text{-})$ la pénalité associée à l'extension de ce *gap*.

Influence sur les alignements

- Pénalités affines :
  - Évitement de *gaps* trop rapprochés :

```
TGATATCGCCA      TGATATCGCCA
TGAT---TCCA  >   TGAT-T--CCA
****    ***      **** *  ***
```

  - Alignements plus réalistes du point de vue évolutif.
- Pénalités logarithmiques :
  - Seulement dans le cas où il est nécessaire d'avoir de très longs *gaps*.



Nombre d'alignements

- Le nombre d'alignements possibles entre deux séquences de longueur m et n est égal à (Waterman, 1984) :

$$f(m,n) = f(m-1,n) + f(m-1,n-1) + f(m,n-1)$$

avec $f(0,j) = f(i,0) = 1 \; \forall i,j$.

- Par ailleurs, ce nombre croît de manière exponentielle (Torres *et al.*, 2003) :

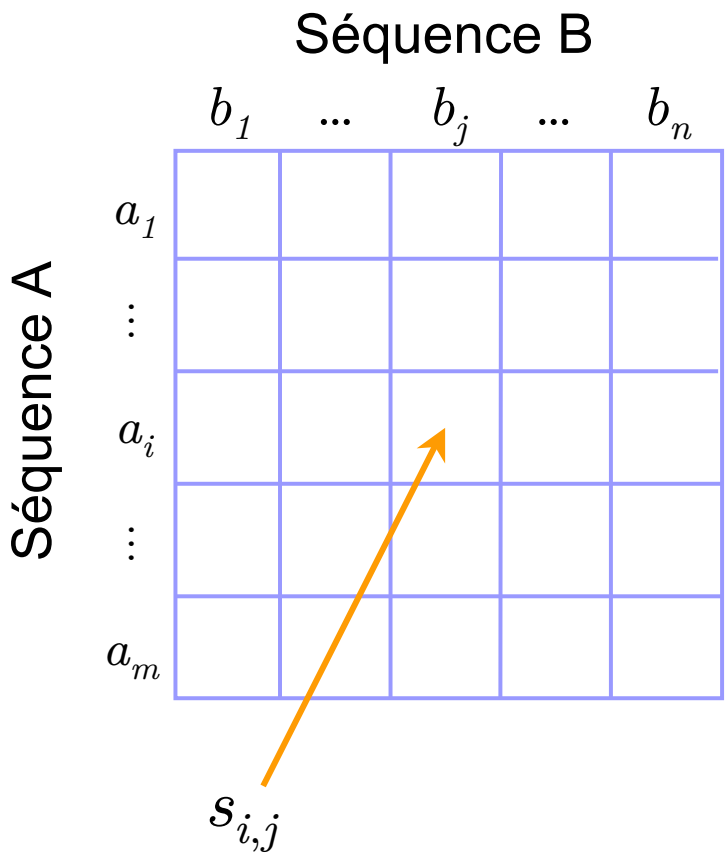
$$f(m,n) = \sum_{k=0}^{\min(m,n)} 2^k \binom{m}{k} \binom{n}{k}$$

Trouver le bon alignement

- Calcul de tous les alignements possibles :
  - Trop long (croissance exponentielle du nombre d'alignements).
  - Peu efficace (recalcul des mêmes valeurs).
- Utilisation d'un algorithme de *programmation dynamique* :
  - Recherche du meilleur alignement possible sur une fraction de la longueur des séquences :
    - Construction progressive de l'alignement.
  - Needleman et Wunsch (1970) : alignement global.
  - Smith et Waterman (1981) : alignement local.

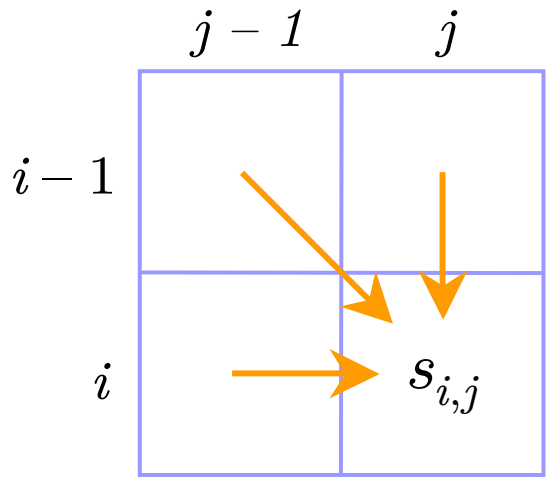
Needleman et Wunsch

- Soit deux séquences A et B de longueurs m et n.
- Soit $\mathbf{S} = (s_{i,j})$, la matrice de chemin associée à ces deux séquences.
- Stockage dans chaque case de $\mathbf{S}$ du score du chemin menant à cette case :
  - Utilisation d'une fonction de score donnée.
  - Le score de l'alignement est la valeur en $s_{m,n}$.



Construction de la matrice

- Soit $s_{i,j}$ la valeur du score optimum dans la case de coordonnées (i, j) :
  - Calcul au moyen des scores dans les trois cases adjacentes $(i-1, j)$, $(i-1, j-1)$ et $(i, j-1)$.
  - Si pas de pénalités affines pour les *gaps* :

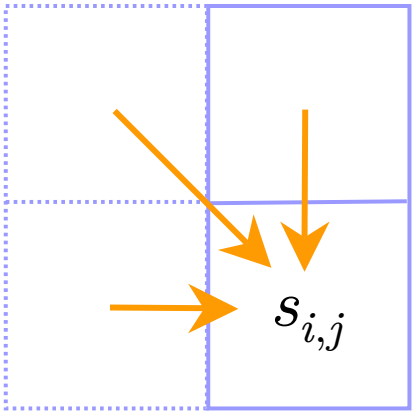


$$s_{i,j} = \max \begin{cases} s_{i-1,j} + \delta(\text{-}) \\ s_{i-1,j-1} + \delta(a_i, b_j) \\ s_{i,j-1} + \delta(\text{-}) \end{cases}$$

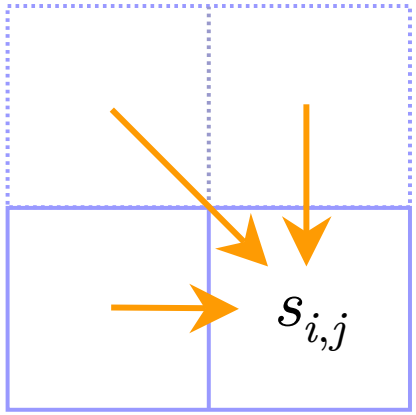
Bords de la matrice

- Les cases situées sur le bord du haut ou le bord gauche ne possèdent pas le total requis de trois cases adjacentes :
  - Ajout d'une ligne et d'une colonne afin d'initialiser la matrice :
    - Le balayage ne se faisant plus qu'avec des indices ≥ 1, on ne rencontre plus de cases nécessitant un traitement particulier.

Bord gauche



Bord du haut



Initialisation de la matrice

- Pénalisation des *gaps* terminaux :

$$
\begin{aligned}
s_{0,0} &= 0 \\
s_{i,0} &= s_{i-1,0} + \delta(\text{-}) \quad \forall i \in [1, m] \\
s_{0,j} &= s_{0,j-1} + \delta(\text{-}) \quad \forall j \in [1, n]
\end{aligned}
$$

- Pas de pénalisation des *gaps* terminaux :

$$s_{0,0} = s_{i,0} = s_{0,j} = 0 \quad \forall i \in [1, m],\ \forall j \in [1, n]$$

Option par défaut de beaucoup de programmes d'alignement (*e.g.*, ClustalW).

Exemple de calcul

		A	G	C	T	A
	0	-2	-4	-6	-8	-10
A	-2	-4	-6	-8	-10	-12
		+1	-2	-4	-6	-7
		-4	-1	-3	-5	-7
T	-4	-1	-3	-5	-7	-9
		-2	+1	-1	-2	-5
		-6	-3	-1	-3	-4
T	-6	-3	-1	-3	-4	-6
		-4	-1	+1	0	-2
		-8	-5	-3	-1	-2
A	-8	-5	-3	-1	-2	-4
		-5	-3	-1	+1	+1
		-10	-7	-5	-3	-1

Identité : +1
Substitution : 0
Gap : -2

```
 A  G  C  T  A
 A  -  T  T  A
+1 -2 +0 +1 +1
```

$s = +1$

```
 A  G  C  T  A
 A  T  -  T  A
+1 +0 -2 +1 +1
```

$s = +1$

Pénalités affines pour les gaps

- Soit $u_{i,j}$ et $v_{i,j}$, les scores associés à l'alignement entre les positions i et j et se terminant par un *gap* dans B ou dans A :

$$u_{i,j} = \max \begin{cases} u_{i-1,j} + \delta_e(\text{-}) \\ s_{i-1,j} + \delta_o(\text{-}) + \delta_e(\text{-}) \end{cases}$$

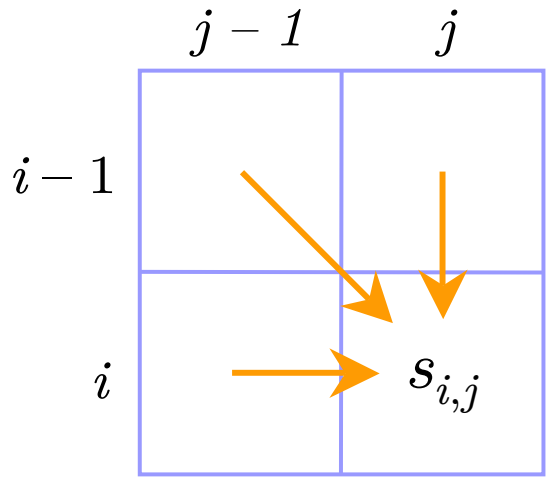
$$v_{i,j} = \max \begin{cases} v_{i,j-1} + \delta_e(\text{-}) \\ s_{i,j-1} + \delta_o(\text{-}) + \delta_e(\text{-}) \end{cases}$$

- Dans ces conditions, le calcul de $s_{i,j}$ est donné par :

$$s_{i,j} = \max \begin{cases} u_{i,j} \\ v_{i,j} \\ s_{i-1,j-1} + \delta(a_i, b_j) \end{cases}$$

Smith et Waterman

- Algorithme dérivé de Needleman et Wunsch :
  - Initialisation des bords à 0.
  - N'importe quelle case de la matrice peut être considérée comme point de départ pour le calcul du score.



$$s_{i,j} = \max \begin{cases} s_{i-1,j} + \delta(\text{-}) \\ s_{i-1,j-1} + \delta(a_i, b_j) \\ s_{i,j-1} + \delta(\text{-}) \\ 0 \end{cases}$$

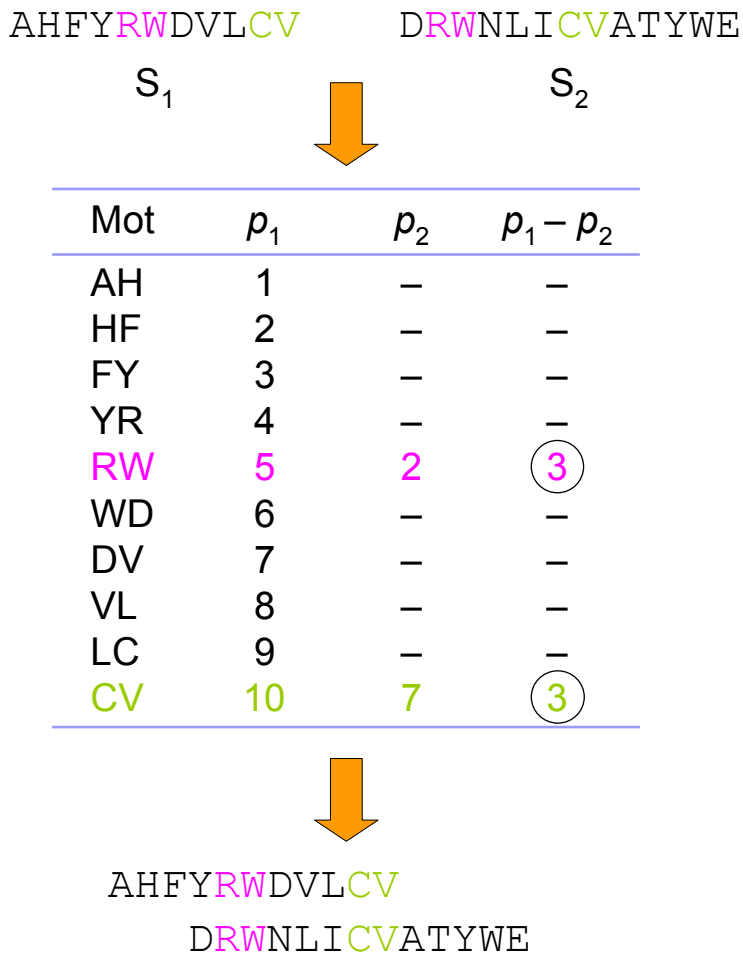
$$s_{i,j} < 0 \Rightarrow s_{i,j} = 0$$

Recherche dans les banques

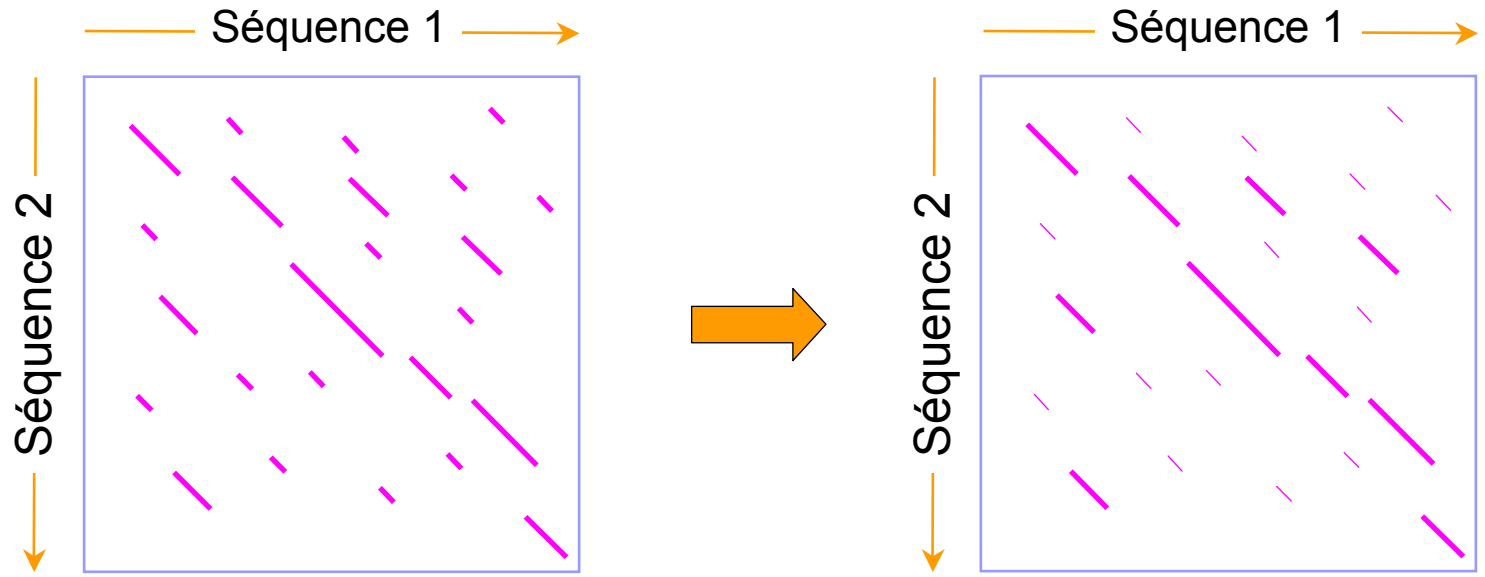
- Algorithme exact (Smith-Waterman) :
  - SSEARCH :
    - Alignements locaux optimaux.
    - Trop lent en pratique et nécessitant beaucoup de mémoire vive.
- Algorithmes fondés sur des heuristiques :
  - FASTA :
    - Recherche de mots *identiques*.
    - Alignement global, ancré sur des régions similaires.
  - BLAST :
    - Recherche de mots *similaires*.
    - Alignements locaux par extension autour de ces mots.

Recherche des mots

- Mots communs entre la séquence requête et une séquence de la banque :
  - Construction d'une table contenant les positions de chaque mot.
  - Détection des mots communs séparés par un même intervalle.
- Construction d'un alignement local à partir de ces mots communs.

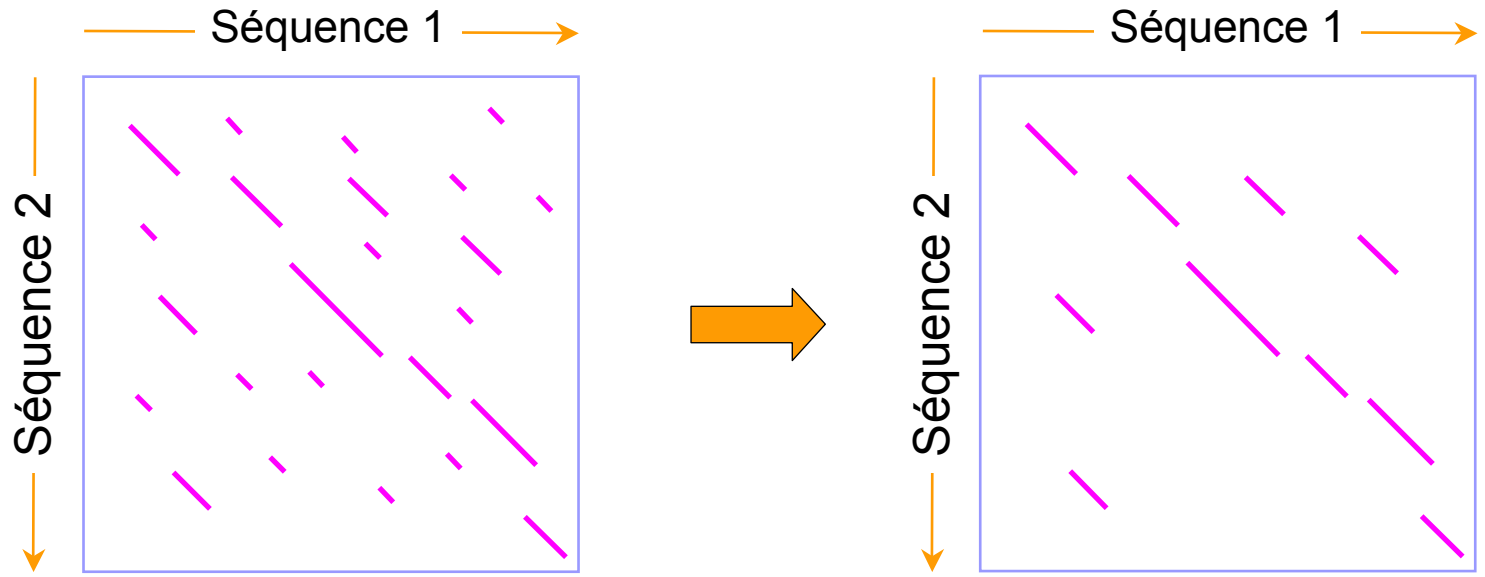


Procédure d'alignement I



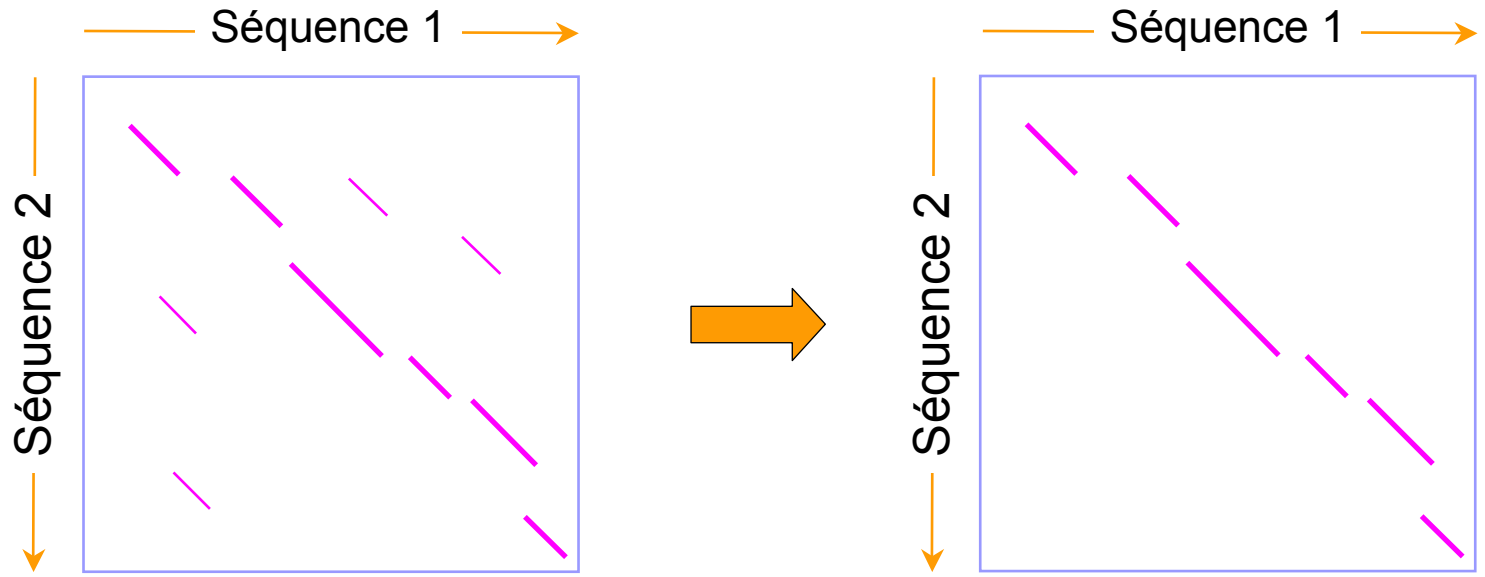
- Calcul de l'ensemble des alignements locaux en partant des mots communs détectés.
- Calcul des scores pour chaque alignement local puis élimination des scores les plus faibles.

Procédure d'alignement I



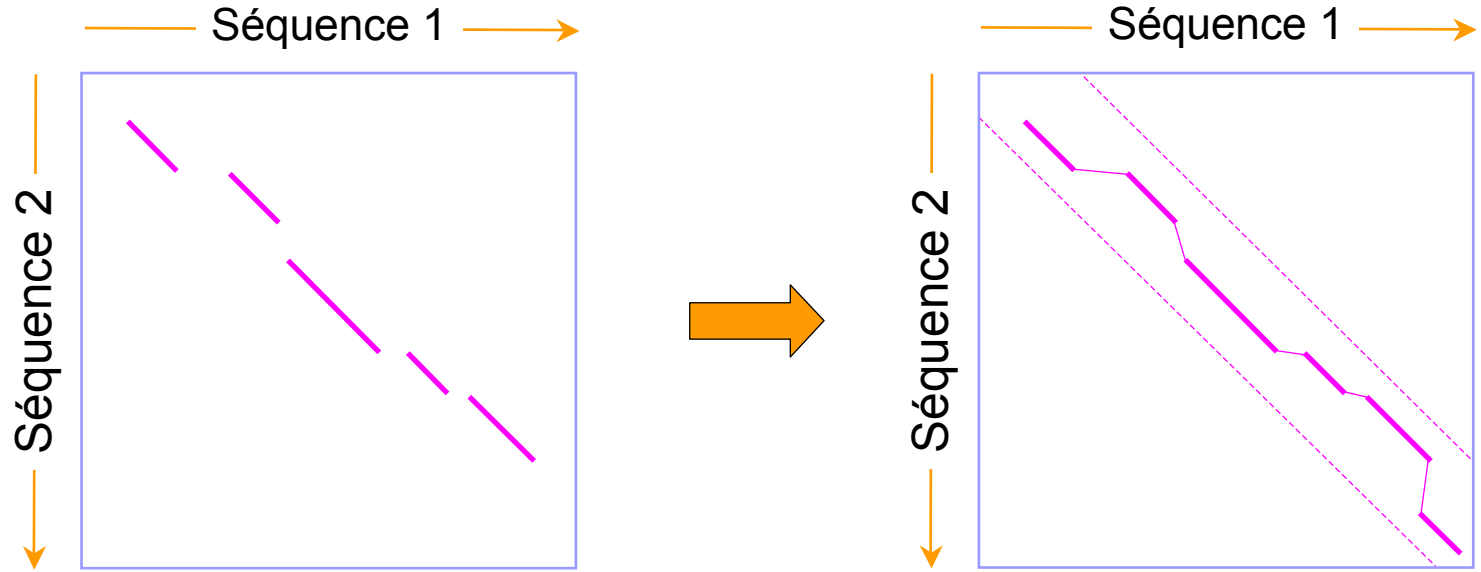
- Calcul de l'ensemble des alignements locaux en partant des mots communs détectés.
- Calcul des scores pour chaque alignement local puis élimination des scores les plus faibles.

Procédure d'alignement II



- Élimination des segments incompatibles avec le segment de score le plus élevé.
- Algorithme de programmation dynamique afin de joindre les segments entre eux, à l'intérieur d'une diagonale.

Procédure d'alignement II

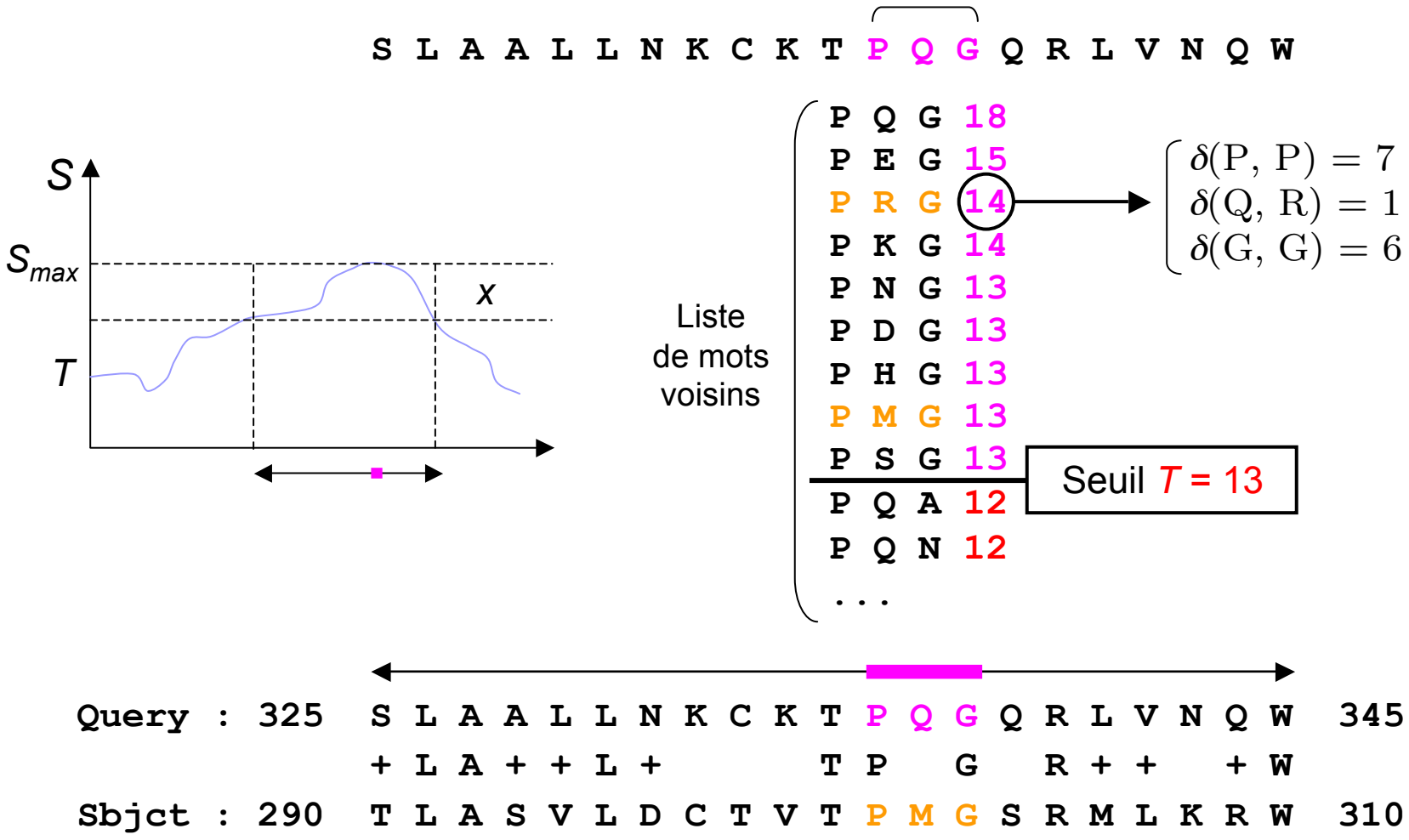


- Élimination des segments incompatibles avec le segment de score le plus élevé.
- Algorithme de programmation dynamique afin de joindre les segments entre eux, à l'intérieur d'une diagonale.

Principe général

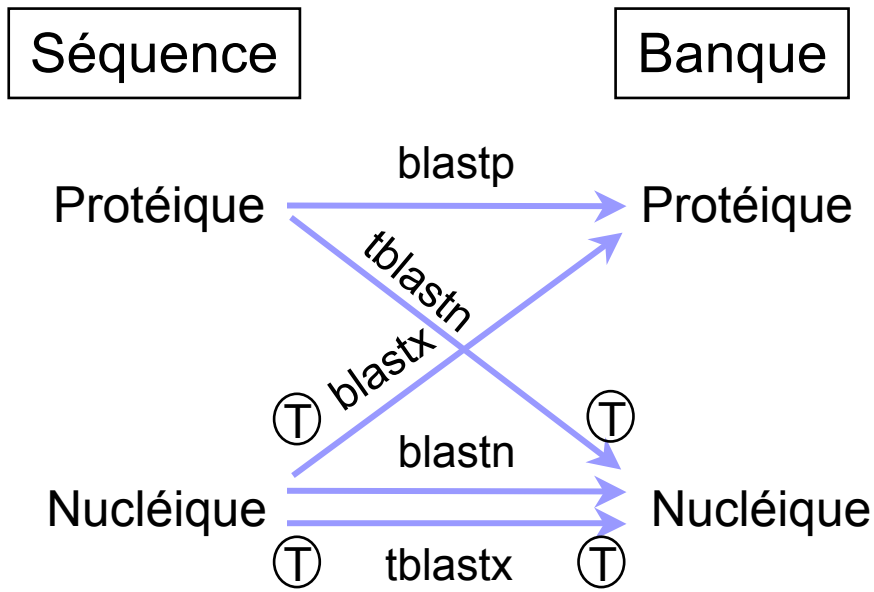
- Recherche de tous les mots de longueur w similaires entre la séquence requête et une séquence de la banque.
- Soit S, le score de ce mot au sens de la matrice de substitution utilisée (BLOSUM62 par défaut) :
  - Si $S \geq T$, extension du segment similaire (*High-scoring Segment Pair* – HSP) à droite et à gauche du mot.
  - Extension stoppée quand :
    - La fin d'une des deux séquences est atteinte.
    - $S \leq 0$.
    - $S \leq S_{\max} - x$.
  - Les valeurs de w, T, et x sont paramétrables par l'utilisateur :
    - Par défaut, $w = 3$ pour les séquences protéiques et $w = 7$ pour les séquences nucléotidiques.

Exemple


S L A A L L N K C K T P Q G Q R L V N Q W
S
S_{max}
T
X
Liste de mots voisins
P Q G 18
P E G 15
P R G 14
P K G 14
P N G 13
P D G 13
P H G 13
P M G 13
P S G 13
P Q A 12
P Q N 12
...
$\delta(P, P) = 7$
$\delta(Q, R) = 1$
$\delta(G, G) = 6$
Seuil T = 13
Query : 325 S L A A L L N K C K T P Q G Q R L V N Q W 345
+ L A + + L + T P G R + + + W
Sbjct : 290 T L A S V L D C T V T P M G S R M L K R W 310

Versions

- blastp : protéine *vs.* protéine.
- blastn : utile pour le non-codant.
- blastx : identification de séquences codantes.
- tblastn : homologues dans un génome non complètement annoté.



Évaluation statistique

- Similarités détectées :
  - Distinguer les relations significatives des similarités dues au hasard.
- Évaluation de la significativité fondée sur :
  - Le score brut d'alignement observé (S).
  - Distribution de probabilité de ce score (loi de Gumbel).
- Mesure sous la forme :
  - Valeur en bits (S').
  - Espérance mathématique (*E-value*).
  - Probabilité (*P-value*).

Équivalences entre les scores I

- Tout d'abord le score en bits S' est donné par :

$$S' = (\lambda S - \ln K)/\ln 2$$

avec K et λ deux paramètres *d'échelle* dépendant de la matrice de substitution utilisée et des pénalités associées aux *gaps*.

- L'espérance mathématique E d'avoir par hasard une HSP dont le score d'alignement serait $\geq$ au score brut observé est égale à :

$$E = Km'n'\mathrm{e}^{-\lambda S}$$

avec m' et n' les tailles *effectives* de la séquence requête et de la banque, calculées à partir des valeurs réelles m et n.

Équivalences entre les scores II

- La relation entre E et S' est donc telle que :

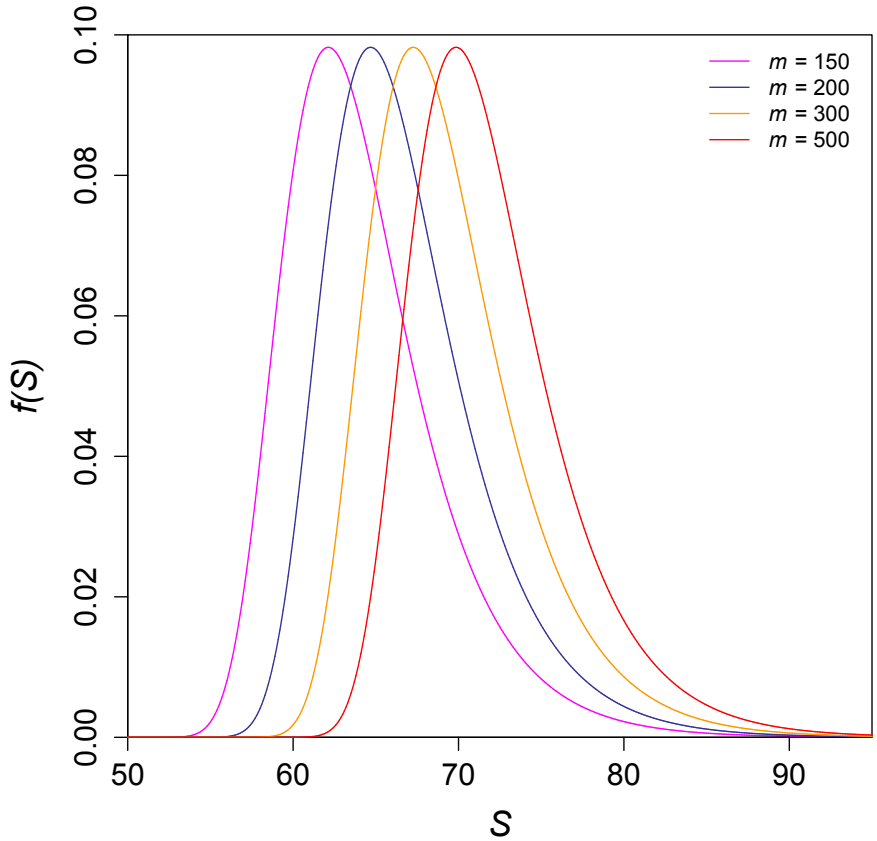
$$E = m'n'2^{-S'}$$

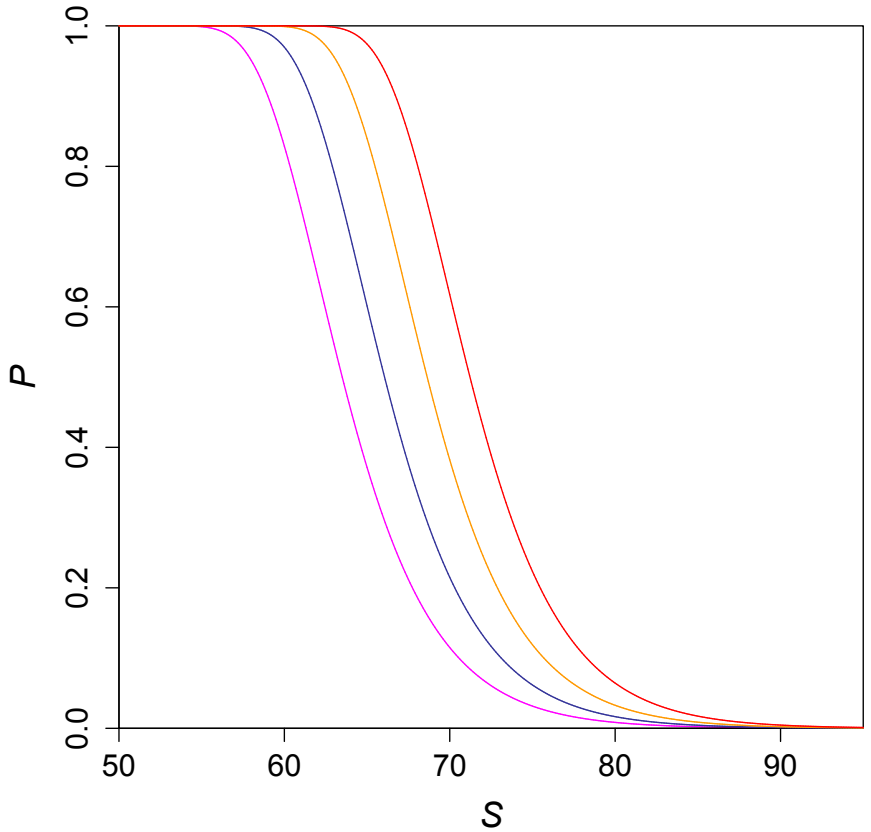
- Enfin, la probabilité P d'avoir par hasard une HSP dont le score d'alignement serait $\geq$ au score brut observé est telle que :

$$P \simeq 1 - \mathrm{e}^{-E}$$

- Utilisation de E plutôt que de P dans les sorties de BLAST :
  - Différences plus directement compréhensibles (*e.g.*, $E = 5$ ou 10 au lieu de $P = 0.9933$ ou 0.99995).
  - A noter que $P \simeq E$ dès que $E \leq 0.01$.

Distribution des valeurs de S

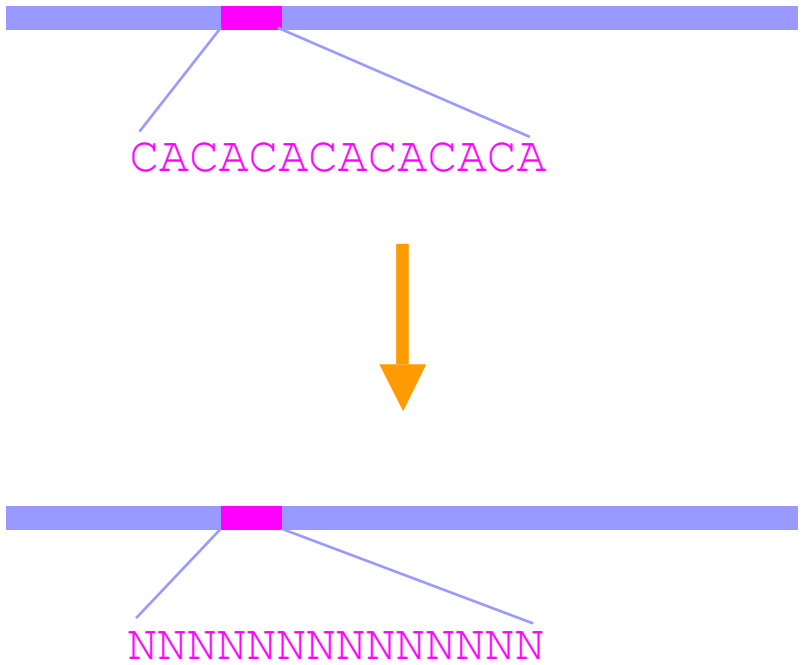




$n = 9418064,\ \delta_o(\text{-}) = -11,\ \delta_e(\text{-}) = -1$, BLOSUM62

Séquences abondantes

- Immunoglobulines :
  - $>$ 127000 séquences dans GenBank.
- Séquences répétées :
  - 10^6 Alu et 10^5 L1 dans le génome humain.
- Programmes de masquage pour BLAST :
  - DUST, XNU, SEG, RepeatMasker.



Serveurs

- Il existe un grand nombre de serveurs permettant d'effectuer des recherches BLAST mais...
  - Toutes les options ne sont pas toujours accessibles.
  - Peu sont exhaustifs du point de vue des banques de données accessibles.
  - Tous ne permettent pas d'accéder à des banques mises à jour quotidiennement.
  - Les possibilités de filtrage pré- ou post-recherche sont rares et limitées.
  - Généralement pas de lien directs avec d'autres applications (*e.g.*, alignements multiples).

BLAST au NCBI

- Répond à (quasiment) toutes les questions précédentes :
  - Accès aux options.
  - Mises à jour en continu.
  - Filtrage taxonomique pré- et post-recherche.
  - Alignements multiples.
- Est particulièrement rapide.
- Bénéficie d'une interface graphique de visualisation des résultats.
- Est très sollicité !
  - Utilisation en matinée recommandée pour les européens si calculs lourds (*e.g.*, PSI-BLAST).

Quelle approche adopter ?

- Stratégie de recherche (nucléique ou protéique).
- Choix d'un algorithme.
- Banque sur laquelle effectuer la recherche.
- Choix de la matrice de substitution.
- Choix des paramètres (pondération des *gaps*, longueurs des mots, seuils, etc.)
- Traitement du bruit de fond.
- Répétition de la recherche.

Du bon usage de BLAST

- L'annotation par similarité peut conduire à certains abus...

```
MZEORFG     ILNSPDRACNLAKQAFDEAISELDSLGEESYKDSTLIMQLLXDNLTLWTSDTNEDGGDE
BOV1433P    IQNAPEQACLLAKQAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDQQDEEAGE
            * * *:.** *******:** ***.* *:************ ********* .::   *

Score = 87.4 bits (213), Expect = 1e-17
Identities = 41/59 (69%), Positives = 50/59 (84%)

LOCUS       BOV1433P     1696 bp    mRNA            MAM       26-APR-1993
DEFINITION  Bovine brain-specific 14-3-3 protein eta chain mRNA, complete
            cds.

LOCUS       MZEORFG       187 bp    mRNA            PLN       31-MAY-1994
DEFINITION  Zea mays putative brain specific 14-3-3 protein, tau protein
            homolog  mRNA, partial cds.
```

Similarités faibles

```
SéqA      CGRRLILFMLATCGECDTDSSE-...-HICCIKQCDVQDIIRVCC
          ||   |        |||                ||   |         |
SéqB      CGSHLVEALYLVCGERGFFYTP-...-EQCCTSICSLYQLENYCN
            |||    |      |   |                ||   |   |
SéqC      YQSHLLIVLLAITLECFFSDRK-...-KRQWISIFDLQTLRPMTA
```

- Les comparaison par paires présentent des limitations dans le cas de similarités limitées à quelques acides aminés :
  - Paire (A, B) : 25% d'identité.
  - Paire (B, C) : 25% d'identité.
  - Triplet (A, B, C) : $< 5\%$ d'identité.

Exemple : famille de l'insuline

Recherche par profils

- Recherche d'un ensemble d'homologues proches.
- Alignement de ces homologues entre eux.
- Calcul d'une matrice de score position-spécifique (ou *profil*) à partir de l'alignement entier ou d'une région définie :
- Recherche dans la banque en utilisant la matrice au lieu de la séquence :
  - Sélection des *hits* ayant un score supérieur à un seuil.
- Éventuellement, modification itérative de la matrice en incorporant les séquences détectées.

Construction de la matrice

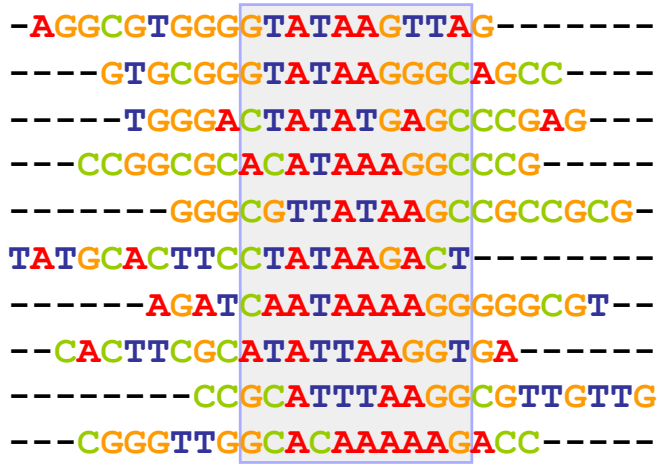
- Soit π_i la fréquence du résidu i dans l'ensemble des séquences utilisées pour construire le profil.
- Soit n le nombre de séquences dans l'alignement et n_{ik} le nombre de résidus i à la position k de l'alignement.
- Positions contenant un ou plusieurs *gaps* :
  - Soit ignorées, soit prises en compte en considérant qu'un *gap* constitue un état de caractère supplémentaire.
- Calcul du score δ_{ik} associé au résidu i à la position k :

$$\delta_{ik} = 2 \log_2 \left[\frac{n_{ik} + 1}{\pi_i (n + 1)} \right]$$

avec arrondissement à l'entier le plus proche.

Exemple de construction

Famille de séquences



Alignement

A	2	1	9	0	8	7	6	6	1	1
C	4	3	0	1	0	0	0	0	1	4
G	4	1	0	0	0	0	4	3	7	3
T	0	5	1	9	2	3	0	1	1	2

Matrice des fréquences

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

Matrice de score

Exemples de recherche

Base	Position									
	1	2	3	4	5	6	7	8	9	10
A	-1	-2	2	-4	2	2	1	1	-2	-2
C	4	3	-1	1	-1	-1	-1	-1	1	4
G	2	-1	-3	-3	-3	-3	2	1	3	1
T	-3	2	-1	4	0	1	-3	-1	-1	0

Matrice de pondération

$S_{\max} = 27$

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

Fenètre glissante

```
G C A G T A T A A G                 S=+2+3+2-3+0+2-3+1-2+1= +3
. C A G T A T A A G G               S=+4-2-3+4+2+1+1+1+3+1=+12
. . A G T A T A A G G G             S=-1-1-1-4+0+2+1+1+3+1= +1
. . . G T A T A A G G G G           S=+2+2+2+4+2+2+2+1+3+1=+21
. . . . T A T A A G G G G A         S=-3-2-1-4+2-3+2+1+3-2= -7
```

Exemples de recherche

Base	Position									
	1	2	3	4	5	6	7	8	9	10
A	-1	-2	2	-4	2	2	1	1	-2	-2
C	4	3	-1	1	-1	-1	-1	-1	1	4
G	2	-1	-3	-3	-3	-3	2	1	3	1
T	-3	2	-1	4	0	1	-3	-1	-1	0

Matrice de pondération

$S_{\max} = 27$

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

Fenètre glissante

```
G A A A G G T G A G                S=+2-2+2-4-3-3-3+1-2+1=-11
. A A A G G T G A G T              S=-1-2+2-3-3+1+2+1+3+0=  0
. . A A G G T G A G T C            S=-1-2-3-3+0-3+1+1-1+4= -7
. . . A G G T G A G T C A          S=-1-1-3+4-3+2+2-1+1-2= -2
. . . . G G T G A G T C A T        S=+2-1-1-3+2-3-3-1-2+0=-10
```

Test de significativité

- Soit S_{obs}, le meilleur score observé sur une séquence requête :
  - Dans notre exemple $S_{\text{obs}} = 21$.
- Génération de B séquences aléatoires ($B \geq 100$) :
  - Même longueur et même composition que la séquence requête.
  - Soit b, le nombre de ces séquences où l'on observe un score $S \geq S_{\text{obs}}$.
  - Dans ce cas la mesure de la significativité est donnée par :

$$\mathbb{P}(S \geq S_{\text{obs}}) = b/B$$

- Dans notre exemple, pour $B = 1000$, une seule séquence générée aléatoirement avait un score $S \geq 21$:

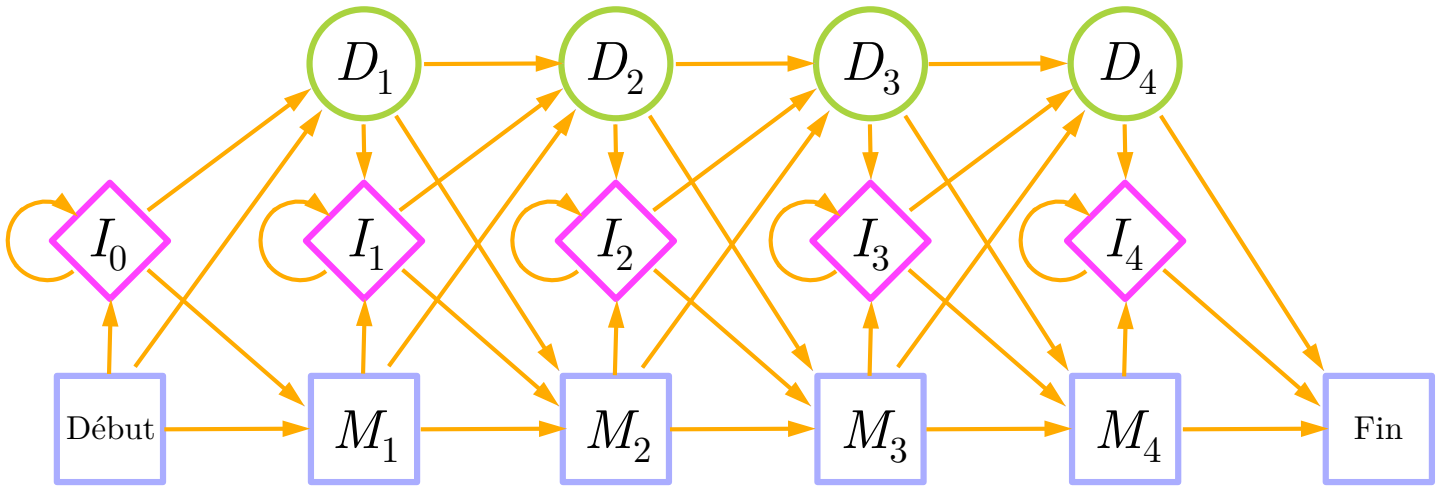
$$\mathbb{P}(S \geq 21) = 1/1000$$

PSI-BLAST

- *Position-Specific Iterated* BLAST :
  1. Recherche BLAST classique.
  2. Calcul d'un profil au moyen des *hits* significatifs sélectionnés par l'utilisateur (utilisation de la E-value).
  3. Nouvelle recherche en utilisant le profil.
  4. Répétition des étapes 2-3 jusqu'à convergence.

Profils généralisés

- Intègrent des modèles de Markov cachés (*Hidden Markov Models* – HMM) au cours de la construction du profil :
  - Meilleure prise en compte des *indels*.
- Programmes disponibles :
  - PFSEARCH (Bücher *et al.*, 1996) :
  - HMMER (Eddy, 1998) :



Alignements multiples

- Généralisation de Needleman et Wunsch ?
  - Pour n séquences de longueur ℓ :
    - Complexité en $O(\ell^n)$.
    - Croissance exponentielle du temps de calcul et de l'espace mémoire.
- Utilisation d'heuristiques :
  - Approximation de l'alignement optimal.

n	t	t
2	0.01 sec	10^{-4} sec
3	1.5 sec	0.015 sec
4	225 sec	2.25 sec
5	9h22 min	5.6 min
6	58 jours	14.06 h
7	24 ans	87.9 jours
8	3612 ans	36 ans

Temps de calcul pour aligner n séquences de longueur $\ell = 150$

Principe de l'alignement progressif

- Construction itérative par ajout progressif de séquences :
  - Établissement d'une *matrice de distances* entre toutes les paires possibles de séquences.
  - Construction d'un *arbre guide* à partir de cette matrice.
  - Ajout de paires et/ou de séquences individuelles dans l'alignement en remontant dans la structure de cet arbre.
- Nombreuses implémentations disponibles :
  - Clustal (Higgins *et al.*, 1988).
  - T-Coffee (Notredame *et al.*, 2000).
  - Mafft (Katoh *et al.*, 2002).
  - Muscle (Edgar, 2004).
  - ProbCons (Do *et al.*, 2005).

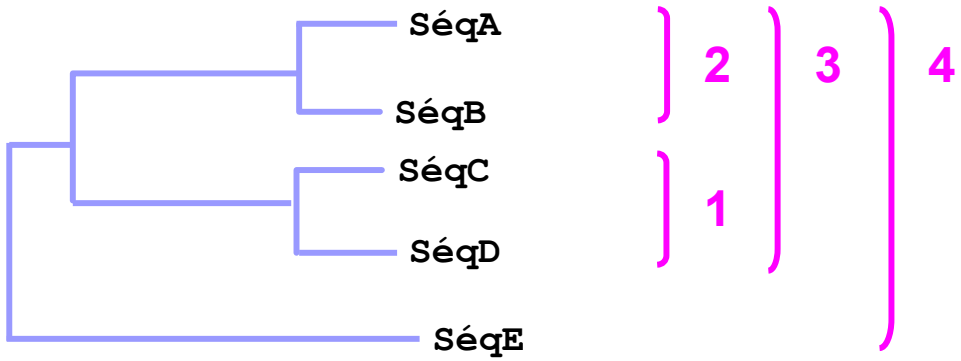
Procédure de Clustal

- Alignement de toutes les paires possibles au moyen d'une heuristique de l'algorithme de Needleman et Wunsch.
- Construction d'une matrice de distance en utilisant les scores d'alignement par paires.
- Construction de l'arbre guide en utilisant cette matrice :
  - La 1ère paire regroupe les deux séquences présentant la distance la plus faible.
  - L'alignement en question est « fixé » et ne changera plus par la suite :
    - Si un *gap* doit être introduit, il le sera à la même position dans les deux séquences.
    - Application du même principe pour toutes les séquences ajoutées par la suite.

Illustration

SéqB	0.17			
SéqC	0.59	0.60		
SéqD	0.59	0.59	0.13	
SéqE	0.77	0.77	0.75	0.75
	SéqA	**SéqB**	**SéqC**	**SéqD**

Calcul de tous les alignements simples et construction de la matrice de distance



Arbre guide par *Neighbour-joining*

Hélices α

Alignement progressif

Pondérations initiale des *gaps*

- Fonction affine pour les *gaps* dans les alignements par paires.
- Correction des valeurs de δ_o(-) et δ_e(-) en fonction de différents facteurs :
  - Le degré de similarité entre les séquences :

$$\delta_o(\text{-}) \propto \%\ \text{identité(A, B)}$$

  - La longueur des séquences :

$$\delta_o(\text{-}) \propto \log[\min(m, n)]$$

  - La différence de longueur entre les séquences :

$$\delta_e(\text{-}) \propto 1.0 + |\log(n/m)|$$

Pondérations supplémentaires

- Prises en compte au moment du groupement des alignements :
  - Diminution de la pénalité à l'emplacement de *gaps* préexistants.
  - Augmentation de la pénalité au voisinage (8 résidus) de *gaps* préexistants.
  - Réduction de la pénalité au niveau de régions contenant une suite d'acides aminés hydrophiles (≥ 5 résidus) :
    - Lien avec la structure 3D des protéines globulaires.
  - Modification spécifiques en fonction des acides aminés présents (*e.g.*, la pénalité est plus faible avec Gly, Asn, Pro).

Le succès de Clustal

- Fut l'un des tous premiers programmes réellement utilisable disponibles (1ère version en 1988).
- Temps de calcul raisonnable pour des jeux de données de taille importante ($n \leq 500$).
- Fonctionalité de calcul d'arbres phylogénétiques (méthode du *Neighbour-Joining*) avec *bootstrap*.
- Utilisable sur la quasi-totalité des architectures disponibles à l'époque :
  - Windows, Unix/Linux, MacOS, VMS.
- Interface graphique (ClustalX).

Limitations

- Perte de l'optimalité au moment du regroupement des paires :
  - Existence de minima locaux.
  - Importance de l'ordre dans lequel sont regroupées les séquences.
  - Impossibilité de corriger ces erreurs par la suite.
- Pas de fonction objective :
  - Impossibilité de déterminer la qualité de l'alignement.
- Désormais beaucoup moins performant que ses concurrents directs :
  - Développement d'une nouvelle version appelée ClustalΩ (Sievers *et al.*, 2011).

Importance de l'arbre guide

```
SéqA  GARFIELD THE LAST FAT CAT
SéqB  GARFIELD THE FAST CAT ---

SéqA  GARFIELD THE LAST FA-T CAT
SéqC  GARFIELD THE VERY FAST CAT

SéqA  GARFIELD THE LAST FAT CAT
SéqD  -------- THE ---- FAT CAT
```

```
SéqB  GARFIELD THE ---- FAST CAT
SéqC  GARFIELD THE VERY FAST CAT

SéqB  GARFIELD THE FAST CAT
SéqD  -------- THE FA-T CAT

SéqC  GARFIELD THE VERY FAST CAT
SéqD  -------- THE ---- FA-T CAT
```

Alignement par paires

Arbre guide

```
SéqA  GARFIELD THE LAST FA-T CAT
SéqB  GARFIELD THE FAST CA-T ---
SéqC  GARFIELD THE VERY FAST CAT
SéqD  -------- THE ---- FA-T CAT
```

Alignement multiple

Alignement progressif

Contraintes de consistance

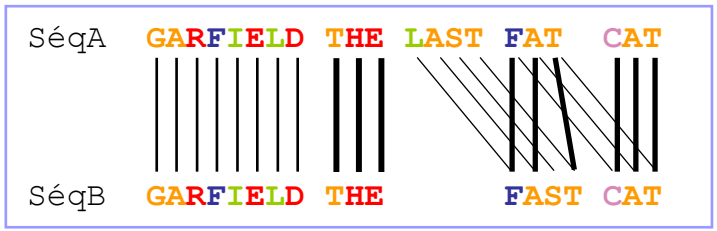
- Utilisées pour améliorer les méthodes d'alignement progressif.
- Prise en compte de combinaisons consistantes de séquences :
  - Données intrinsèques (alignements par paires et par triplets).
  - Données extrinsèques (*e.g.*, structures 3D).
- Introduites dans différents programmes :
  - T-Coffee (Notredame *et al.*, 2000).
  - ProbCons (Do *et al.*, 2005).
  - ProbAlign (Roshan et Livesay, 2006).

Stratégie de T-Coffee

```
SéqA  GARFIELD THE LAST FAT CAT
SéqB  GARFIELD THE FAST CAT ---

SéqA  GARFIELD THE LAST FA-T CAT
SéqC  GARFIELD THE VERY FAST CAT

SéqA  GARFIELD THE LAST FAT CAT
SéqD  -------- THE ---- FAT CAT

SéqB  GARFIELD THE ---- FAST CAT
SéqC  GARFIELD THE VERY FAST CAT

SéqB  GARFIELD THE FAST CAT
SéqD  -------- THE FA-T CAT

SéqC  GARFIELD THE VERY FAST CAT
SéqD  -------- THE ---- FA-T CAT
```

Alignements par paires de départ

```
SéqA  GARFIELD THE LAST FAT CAT
      |||||||| ||| |||| |||
SéqB  GARFIELD THE FAST CAT

SéqA  GARFIELD THE LAST FAT CAT
      |||||||| ||| |||| || \ \\\
SéqC  GARFIELD THE VERY FAST CAT
      |||||||| |||      |||| |||
SéqB  GARFIELD THE       FAST CAT

SéqA  GARFIELD THE LAST FAT CAT
                |||      ||| |||
SéqD            THE      FAT-CAT
                |||      || \ \\\
SéqB  GARFIELD THE       FAST CAT
```



```
SéqA  GARFIELD THE LAST FAT CAT
SéqB  GARFIELD THE FAST --- CAT
```

Alignement A-B retenu

Aminoacyl-ARNt synthétases d'*E. coli*

```
SYL_ECOLI   HMGHVRNYTIGDVIARYQRMLGKNVLQPIGWDAFGLPAEGAAVKNNTAPA----------/…/
SYV_ECOLI   HMGHAFQQTIMDTMIRYQRMQGKNTLWQVGTDHAGIATQMVVERKIAAEEGKTRHDYGRE/…/
SYI_ECOLI   HIGHSVNKILKDIIVKSKGLSGYDSPYVPGWDCHGLPIELKVEQEYGKPG----EKFTAA/…/
SYM_ECOLI   HLGHMLEHIQADVWVRYQRMRGHEVNFICADDAHGTPIMLKAQQLGITPE----Q-----/…/
            *:**  :    *   : : :  * :    . *   * .    . :

SYL_ECOLI   LVYTGMSKMSKSKNNGIDPQVMVER-----------------------------------
SYV_ECOLI   -------KMSKSKGNVIDPLDMVDGISLPELLEKRTGNMMQPQLADKIRKRTEKQFPNGI
SYI_ECOLI   -------KMSKSIGNTVSPQDVMNK-----------------------------------
SYM_ECOLI   -------KMSKSRGTFIKASTWLNH-----------------------------------
                  ***** .. :..   ::
```

Alignement de T-Coffee

```
SYL_ECOLI   HMGHVRNYTIGDVIARYQRMLGKNVLQPIGWDAFGLPAEGAAVKNNTAPAP---------/…/
SYV_ECOLI   HMGHAFQQTIMDTMIRYQRMQGKNTLWQVGTDHAGIATQMVVERKIAAEEGKTRHDYGRE/…/
SYI_ECOLI   HIGHSVNKILKDIIVKSKGLSGYDSPYVPGWDCHGLPIELKVEQEYGKPGEK----FTAA/…/
SYM_ECOLI   HLGHMLEHIQADVWVRYQRMRGHEVNFICADDAHGTPIMLKAQQLGITPEQMIG------/…/
            * :*:**  :   *   : : : * :    . *  * .    . :

SYL_ECOLI   MHLLYFRFFHKLMRDAGMVNSDEPAKQLLCQG--MVLADAFYYVGENGERNWVS------
SYV_ECOLI   EGQKMSKSKGNVIDPLDMVDGISLPELLEKRTGNMMQPQLADKIRKRTEKQFPNGIEPHG
SYI_ECOLI   APYRQVLTHGFTVDGQGRKMSKSIGNTVSPQD-------VMNKLGADILRLWVASTDYTG
SYM_ECOLI   ------------------------------------------------------------
```

Alignement de Clustal

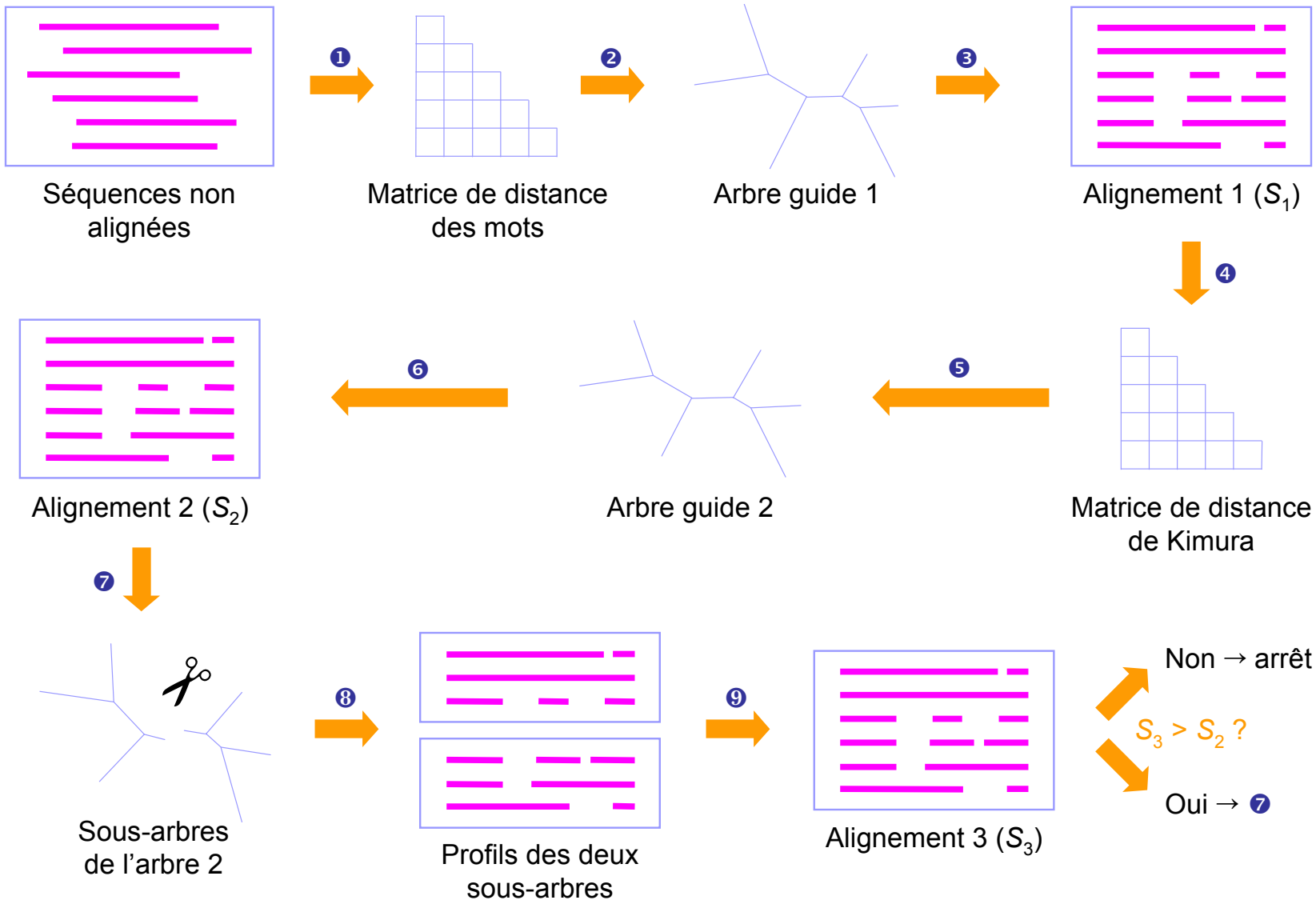
Interêt et limitations

- Résultats très supérieurs à ceux de ClustalW.
- Possibilité d'améliorer les alignements en utilisant des données structurales.
- Intègre une fonction objective d'évaluation de la qualité des alignements.
- Très gourmand en mémoire et en temps de calcul :
  - Jeux de données de taille limitée ($n \leq 100$).

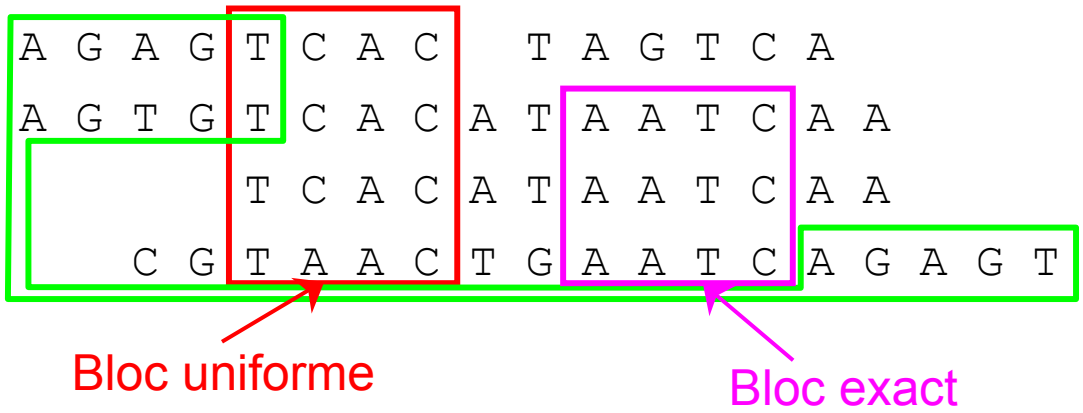
Amélioration par itération

- Premier alignement par la méthode progressive.
- Raffinements successifs de l'alignement de départ jusqu'à convergence.
- Implémentation la plus performante réalisée dans le programme Muscle :
  - Rapide.
  - Peut travailler sur des jeux de données de très grande taille (plusieurs milliers de séquences).
  - Bon résultats en terme de qualité des alignements.

Procédure de Muscle



Alignement par blocs

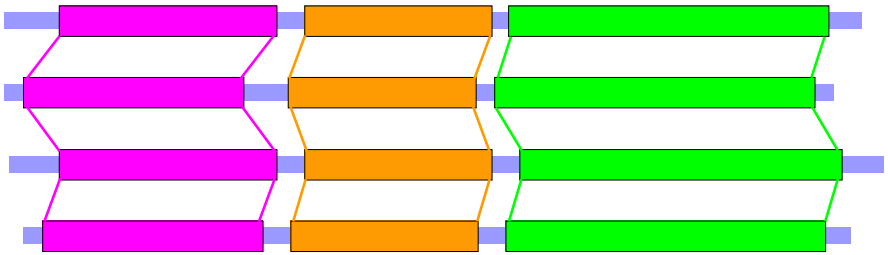


- Recherche des blocs similaires sans *gaps*.
- Sélection de la meilleure combinaison de blocs compatibles entre eux (heuristique).
- Plus lent que les alignements progressifs.
- Implémentation disponible :
  - Dialign2 (Subramanian *et al.*, 2008).

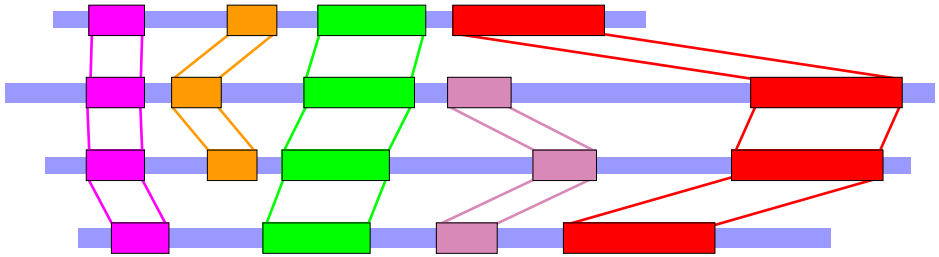
Jeux de données de référence

- Banques créées manuellement, souvent à partir d'alignements structuraux :
  - BAliBASE (Thompson *et al.*, 1999).
  - PALI (Balaji *et al.*, 2001).
  - OXBench (Raghava *et al.*, 2003).
  - PREFAB (Edgar, 2004).
  - SABmark (Van Walle *et al.*, 2005).
  - HomFam (Blackshields *et al.*, 2010).
- Utilisées pour évaluer la qualité des algorithmes disponibles :
  - Certains programmes « passent » mieux sur certaines références que sur d'autres.

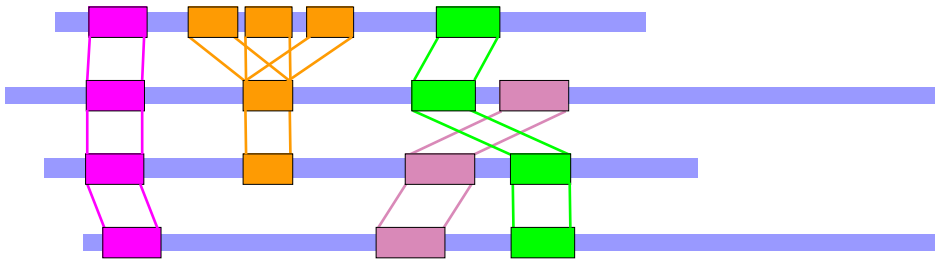
Programmes recommandés



Clustal
Multalin



Dialign2
Muscle
T-Coffee



Match-Box
MEME
PIMA

Cas particuliers

- Alignement de séquences codantes :
  - SeaView (Gouy *et al.*, 2010) :
    - Alignement des séquences protéiques.
    - Calage du nucléique sur l'alignement protéique.
- Alignement ADNc / ADN génomique :
  - SIM4 (Florea *et al.*, 1998).
- Alignement protéine / ADN :
  - GeneWise (Birney *et al.*, 2004).

Quelques conseils

- Considérez les résultats avec recul.
- Pour la phylogénie, n'utilisez que les sites pour lesquels l'hypothèse d'homologie est vraisemblable.
- Essayez d'identifier les régions dont vous pensez qu'elles sont correctement alignées :
  - Un alignement multiple peut (parfois) être amélioré « à la main » à l'aide d'un éditeur :
    - SeaView, STRAP, CINEMA, etc.
  - Il existe des programmes permettant de filtrer automatiquement les régions mal alignées :
    - Gblocks, TrimAl, BMGE, Guidance, etc.