

Modélisation et recherche de motifs biologiques

Hélène Touzet

Équipe Bioinfo — LIFL — USTL

Licence – Master

www.lifl.fr/~touzet/M1

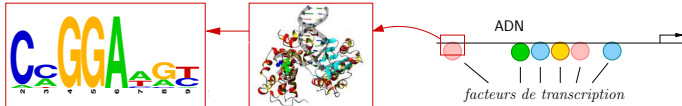
Motifs biologiques

- ▶ **Motifs approchés** – décrits par un modèle
 - ▶ ADN : sites de fixation de facteurs de transcription, ...
 - ▶ Protéines : domaines enzymatiques, sites d'interaction ADN-protéines, protéines-protéines, ...

- ▶ **Motifs exacts** – décrits par un mot
 - ▶ ADN : codons START, STOP, sites de clivage d'enzymes de restriction, ...
 - ▶ Protéines : signatures (moins courant)

Motifs approchés

► Transfac – motifs ADN



700 motifs, construits à partir de plus de 15 000 séquences

► Prosite – motifs protéiques

Créé en 1993

Environ 1400 motifs

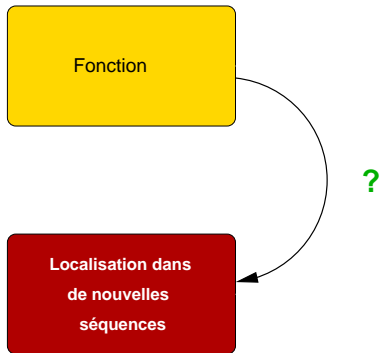


► Pfam – motifs protéiques

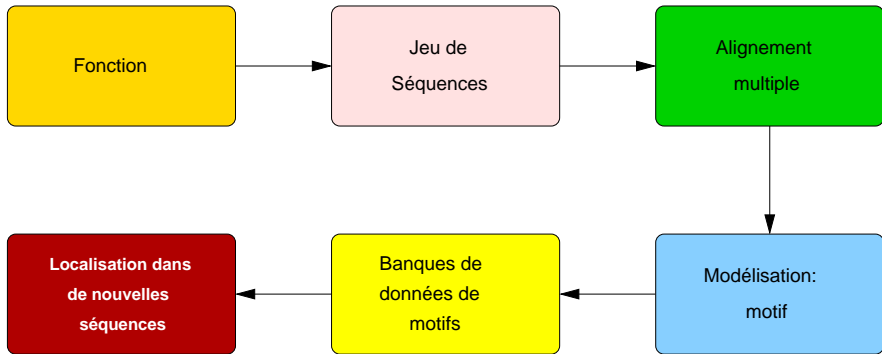


8153 familles de protéines (décembre 2005)
Taux de couverture Pfma-A: 75% (vérifiés)
Taux de couverture Pfam-B:19%

Motifs approchés



Motifs approchés

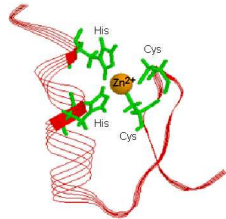


- ▶ **Problème 1:** trouver une représentation des motifs à partir des alignements multiples
- ▶ **Problème 2:** concevoir des algorithmes pour localiser les occurrences des motifs dans une nouvelle séquence

Motifs Prosite

► Exemple : doigt de zinc

```
YKCT--VCR---KDISSESRLRTHMFKQ-HH  
LKCSVPGCK---RSFRKKRALRIHVSE---H  
FECN--MCG---YHSQDRYEFSSHITRG-EH  
YTCG--YCTEDSPSPFPRPSLLESHISL--MH  
YKCEFADCE---KAFSNASDRAKHQNR--TH  
FVCHWQDCSRELRPFKAQYMLVVMRR---H  
FRCS--ECS---RSFTHNSDLTAHMRK---H  
CKCETENCN---LAFTTASNMRHLHFKR--AH  
YRCSYEDCQ---TVSPTWTALQTHLKK---H  
FRCV--WCK---QSFPPTLEALTTMVKDS-KH
```



C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H

► Syntaxe d'une expression Prosite

- : séparation des éléments
- x : n'importe quel acide aminé
- (3,5) : nombre d'occurrences (entre 3 et 5)
- [FY] : alternative (F ou Y)

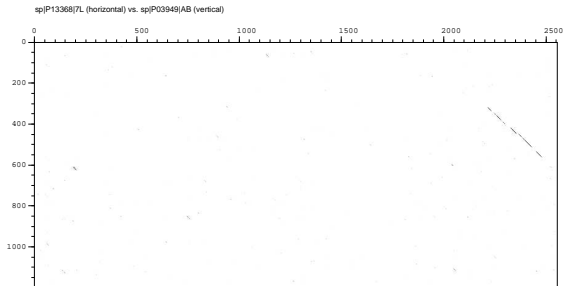
Motifs Prosite - rappel TP 1

- ▶ Comparaison de deux protéines kinases
- ▶ Apparition d'une région conservée : domaine kinase

2215-2242 : LGSGAFGEVYeGqlktedseepqr.....VAIK

862 - 891 : LGEGNFGQVWkAeaddlsghfgatri....VAVK

[LIV]-G-{P}-G-{P}-[FYWMGSTNH]-[SGA]-{PW}-[LIVCAT]-{PD}-x-[GSTACLIVMFY]
-x(5,18)-[LIVMFYWCSTAR]-[AIVP]-[LIVMFAGCKR]-K



Limite des motifs Prosite

- ▶ **Absence de prise quantitative en compte des erreurs**
 - ▶ Composition des colonnes
 - ▶ Probabilité d'une insertion
 - ▶ Compromis difficile entre la sensibilité et la sélectivité

- ▶ **Modèle enrichi, sous forme de **profil****
 - ▶ Modèle de base : automate
 - ▶ Ajout d'informations statistiques sur la composition
 - ▶ Fondement théorique : apprentissage, algorithme de Baum Welch

Description par profil

V E D - - L I R Y

V E D - - L R R Y

P N E - - L R R F

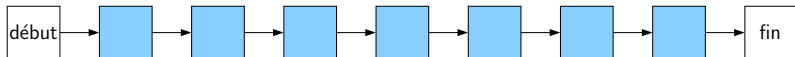
D N K A A L R R F

A E E - - L A - -

Description par profil

V	E	D	-	-	L	I	R	Y
V	E	D	-	-	L	R	R	Y
P	N	E	-	-	L	R	R	F
D	N	K	A	A	L	R	R	F
A	E	E	-	-	L	A	-	-

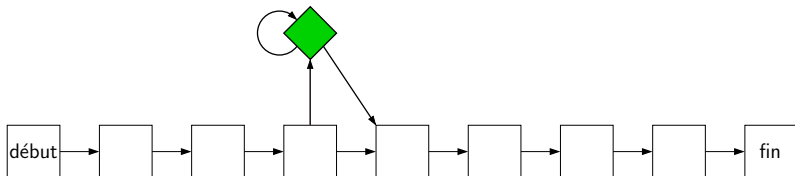
Création d'un état par colonne



Description par profil

V	E	D	-	-	L	I	R	Y
V	E	D	-	-	L	R	R	Y
P	N	E	-	-	L	R	R	F
D	N	K	A	A	L	R	R	F
A	E	E	-	-	L	A	-	-

Prise en compte des insertions



Description par profil

V E D - - L I R Y

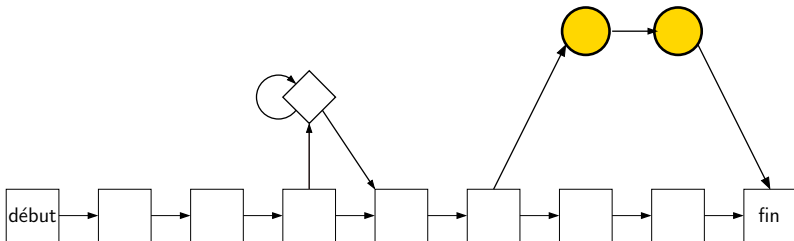
V E D - - L R R Y

P N E - - L R R F

D N K A A L R R F

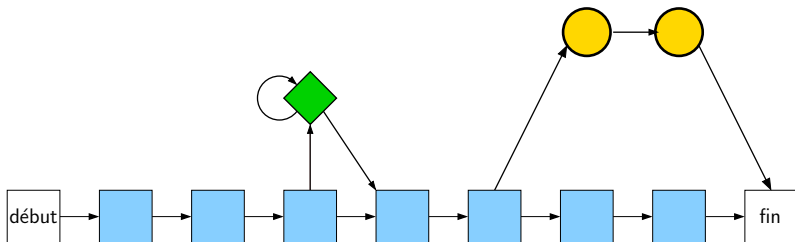
A E E - - L A - -

Prise en compte des délétions



Description par profil

V	E	D	-	-	L	I	R	Y
V	E	D	-	-	L	R	R	Y
P	N	E	-	-	L	R	R	F
D	N	K	A	A	L	R	R	F
A	E	E	-	-	L	A	-	-



Description par profil

V E D - - L I R Y

V E D - - L R R Y

P N E - - L R R F

D N K A A L R R F

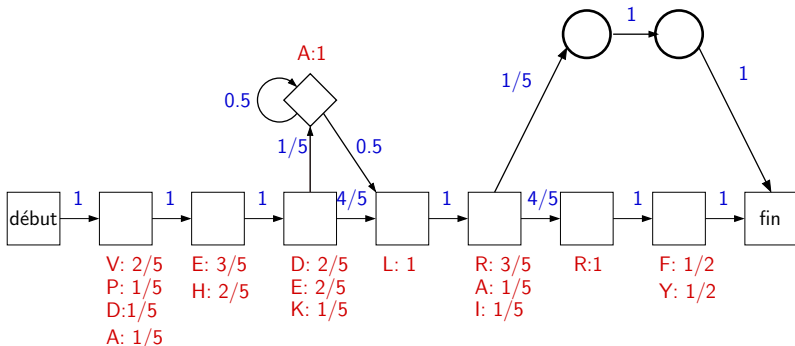
A E E - - L A - -

■ **émissions**

fréquences des acides aminés

■ **transitions**

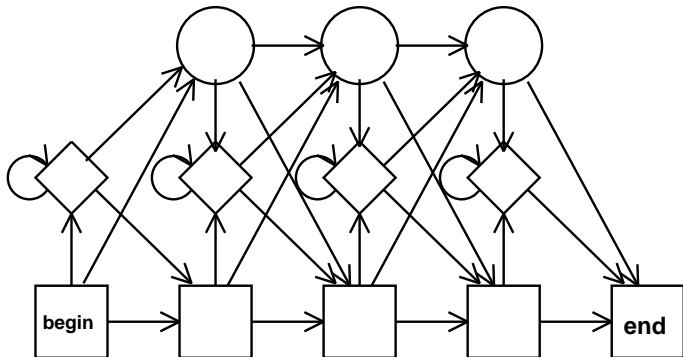
circulation dans le modèle
indels



▶ **En résumé**

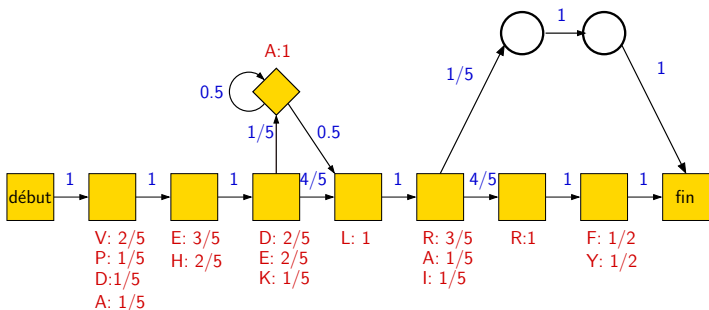
- ▶ Etats matchants : colonnes avec moins de 50% de -
- ▶ Etats d'insertion : majorité de -
- ▶ Etats de délétion : minorité de -
- ▶ Probabilités d'émission : on compte le nombre d'occurrences de chaque acide aminé
- ▶ Probabilités de transition : on compte le nombre de séquences empruntant la transition
- ▶ Correction avec les pseudo-poids : +1 à chaque compte (loi de Laplace – apprentissage)

► **Modèle complet**



Recherche avec un profil

- **Score** : probabilité maximale d'un mot dans le modèle



Score de VHKALARY

$$1 \times \frac{2}{5} \times 1 \times \frac{2}{5} \times 1 \times \frac{1}{5} \times \frac{1}{5} \times 1 \times 0.5 \times 1 \times 1 \times \frac{1}{5} \times 1 \times 1 \times \frac{1}{2} \times 1$$

- **Algorithme de Viterbi** : trouver le chemin de probabilité maximale
Recherche d'un chemin optimal dans un graphe

Motifs exacts

- ▶ **Données**

- ▶ un texte T de longueur n : $T(0..n - 1)$
- ▶ un motif M de longueur m : $M(0..m - 1)$

- ▶ **Problème**

Trouver toutes les occurrences de M dans T

Algorithme par force brute



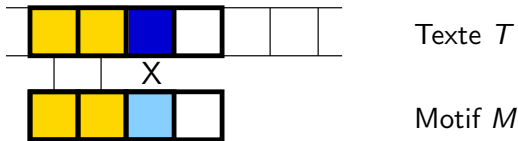
Texte T



Motif M

- ▶ **Balayage** du texte, du début à la fin

Algorithme par force brute



- ▶ **B**alayage du texte, du début à la fin
- ▶ **À** chaque position du texte, comparaison du motif et du texte, jusqu'à rencontrer un mismatch, ou épuiser le motif

Algorithme par force brute



Texte T



Motif M

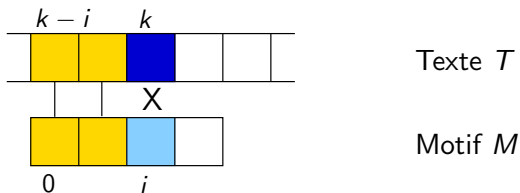


- ▶ **B**alayage du texte, du début à la fin
- ▶ **À** chaque position du texte, comparaison du motif et du texte, jusqu'à rencontrer un mismatch, ou épuiser le motif
- ▶ **D**écalage du motif vers la droite, de position en position

Amélioration de l'approche par force brute

- ▶ Faire des décalages de plus de une position
- ▶ Condition pour que les décalages restent corrects
ne pas rater une occurrence du motif
- ▶ Pré-traitement du motif M

Algorithme de Knuth, Moris et Pratt

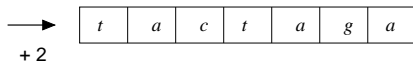
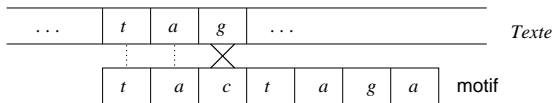
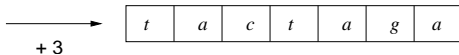
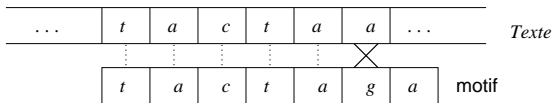


- ▶ Deux informations sont exploitées

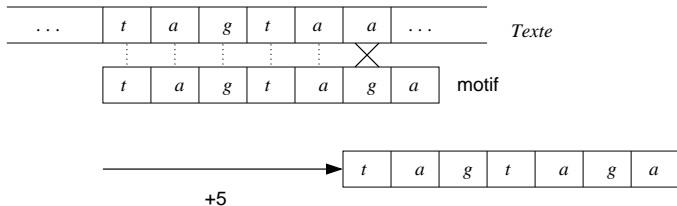
$$\begin{aligned}T(k - i..k - 1) &= M(0..i - 1) \\T(k) &\neq M(i)\end{aligned}$$

- ▶ Quand un mismatch intervient en position i , on décale le motif le long du texte directement à la prochaine position où peut démarrer M

► Exemple 1: le motif *tactaga*



► Exemple 2: le motif *tagtaga*



Mise en œuvre

► Calcul du tableau **Next**

- i** : position dans le motif
Next(i) : longueur du plus long mot u tel que
 u est un préfixe de M
 u est un suffixe de $M(0..i - 1)$
 $uM(i)$ n'est pas un préfixe de M
ou **-1** si u n'existe pas

► Règle d'utilisation

- Quand un mismatch intervient à la position i du motif, on décale de $i - \text{Next}(i)$ vers la droite
- La comparaison reprend à partir de la position i du motif.

tactaga

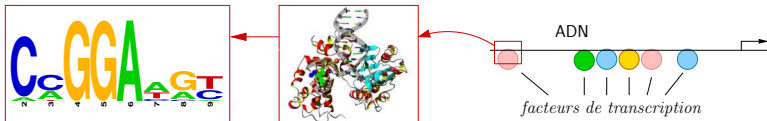
i	0	1	2	3	4	5	6	7
Next(i)	-1	0	0	-1	0	2	0	0

tagtaga

i	0	1	2	3	4	5	6	7
Next(i)	-1	0	0	-1	0	0	3	0

Comptage des mots

- ▶ Application 1: Comment les gènes sont-ils régulés ?



- ▶ Facteurs de transcription (FT): protéines régulatrices
- ▶ Premiers acteurs de la régulation transcriptionnelle
- ▶ Domaine de liaison à l'ADN : court motif nucléique conservé (site de fixation)

▶ **Recherche d'oligonucléotides sur-représentés**

- ▶ les mots qui apparaissent plus que ce qui est attendu par hasard ont un rôle biologique
- ▶ ce sont des sites de fixation potentiels

▶ **Exemple:** 12 gènes de la levure, régulés par la méthionine

SAM2, MET6, MIUP3, MET30, MET3, MET14, MET1, SAM1, MET17, ZWF1, MET2

- ▶ analyse de la région $[-800, -1]$ en amont du site d'initiation de la transcription
- ▶ comptage de tous les mots de longueur 6
- ▶ modèle de fond : %GC, régions intergéniques de la levure
- ▶ calcul de la probabilité du comptage de chaque mot

cacgtg	cacgtg cacgtg	13	1.26	1.00e-9
ccacag	ccacag ctgtgg	11	2.22	2.10e-5
acgtga	acgtga tcacgt	13	3.1	2.20e-5
aactgt	aactgt acagtt	17	5.28	3.90e-5
actgtg	actgtg cacagt	12	3.16	0.00011
gccaca	gccaca tgtggc	10	2.59	0.00037
gcttcc	gcttcc ggaagc	12	6.6	0.00037
séquence	2 brins	n.obs.	n.att.	P-value

n.obs. : nombre d'occurrences observé

n.att. : nombre d'occurrences attendu

P-value : probabilité du nombre d'occurrences observé

motif 1

```
tcacgt..  ..acgtga
.cacgtg.  .cacgtg
..acgtga  tcacgt..
```

```
tcacgtga  tcacgtga
```

motif 2

```
aactgt..  ..acagtt
.actgtg.  .cacagt.
..ctgtgg  ccacag..
```

```
aactgtgg  ccacagtt
```

complexe Met4p/Cbfl/Met28

Met31p

Comptage des mots

- ▶ **Application 2** : mots exceptionnellement rares
- ▶ **Recensement** de tous les mots de longueur 6 dans le génome de la bactérie de *Escherichia coli*

	comptage observé	comptage attendu
ggcggc	96	2041,0
gccggc	294	1764,0
ctgcag	958	1981,1
tccgaa	906	1708,6

- ▶ **Remarque ? Explication ?**

Comptage des mots

Problème

- ▶ **Données** : un texte T , un entier k
- ▶ **Question** : compter le nombre d'occurrence de chaque mot de longueur k dans T

Implémentation

- ▶ **Utilisation** d'une structure d'index V : table de hachage sans collision
- ▶ **Taille** de la table : $4^k - 1$
- ▶ **Chaque case** de la table stocke le nombre d'occurrences d'un mot

- ▶ **Fonction de hachage, pour l'ADN:**

$$e : \{A, C, G, T\} \rightarrow \{0, 1, 2, 3\}$$

$$\mathcal{H}(u_0 \dots u_k) = \sum_{j=0}^k e(u_j) 4^j$$

- ▶ **Exemple : mots de longueur 6**

$$\mathcal{H}(AAAAAA) = 0 \quad \dots \quad \mathcal{H}(GTTTTT) = 6^4 - 2$$

$$\mathcal{H}(CAAAAA) = 1 \quad \mathcal{H}(TTTTTT) = 4^6 - 1$$

- ▶ **Remplissage de la table : balayage du texte**

$$h = 4 \times h + e(x) \pmod{4^k}$$

$$V(h) = V(h) + 1$$

x est la lettre courante

- ▶ **Mise à jour de la clé de hachage h en temps constant**

Le rêve d'un biologiste

“Avec les algorithmes vus précédemment, je sais qu'on est capable de comparer deux séquences, et de regarder si elles ont une fonction commune.”

“Que faire si je ne dispose que d'une séquence ?”

- ▶ Comparaison avec toutes les autres séquences existantes sur terre, disponibles dans les banques de données (Genbank, Swissprot, . . .)
- ▶ Besoin d'algorithmes d'alignements locaux extrêmement efficaces

Le programme Blast

Basic Local Alignment Search Tool - Altschul *et al.* - 1997

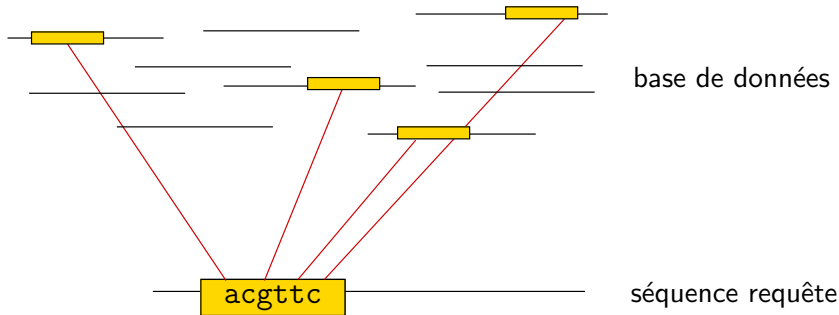


Ne pas construire d'alignement avec toutes les séquences de la banque de données, mais seulement avec un petit nombre de séquences candidates qui peuvent être sélectionnées de manière très efficace

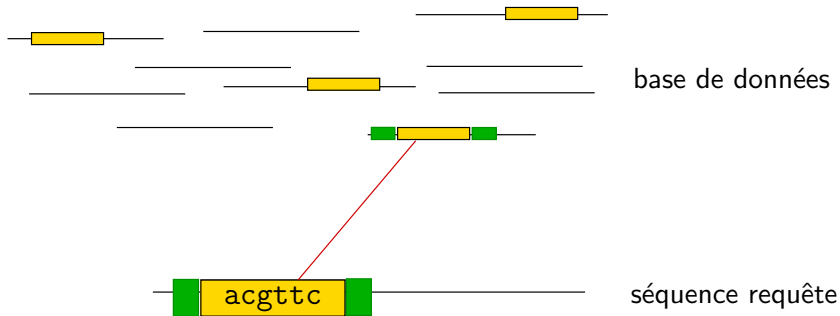
- ▶ Heuristique très rapide pour l'alignement local
- ▶ Recherche de mots exacts communs entre la séquence et la banque de données

Algorithme de Blast

- ▶ **Étape 1:** pré-traitement de la banque de données
 - ▶ Choix d'une taille de mots: k
 - ▶ Indexation de tous les mots de longueur k dans une table de hachage, avec leurs positions
- ▶ **Étape 2:** traitement de la séquence requête : recensement de tous les mots de longueur k et recherche dans la table de hachage



- ▶ **Étape 3** : extension de ces points d'ancrage de proche en proche, pour avoir un score significatif.



Performances

- ▶ **Comparaison exhaustive** entre le génome humain (3 milliards de bases) et le génome de la souris (3 milliards de bases)
 - ▶ **Temps CPU** estimés pour un Pentium III 700MH, 1GB :
 - ▶ **Alignement local exact** (Smith&Waterman) : 100 siècles
 - ▶ **BLAST** : 19 années
- ▶ **Plus de 100 000 citations** scientifiques
- ▶ **Plus de 150 000 requêtes** quotidiennes (NCBI)