Analyse de Séquences

M1 BIBS

Représentation et recherche de motifs

http://rna.igmors.u-psud.fr/gautheret/cours/

V. 2011.1

Plan

- Représenter les motifs
- Estimer la performance d'une recherche de motifs
- Application à la détection de gènes
- Découverte de motifs inconnus
- Logiciels

Représenter un motif = représenter ensemble de séquences

- Identifier les résidus essentiels,
- Identifier les domaines fonctionnels
- Etablir la signature fonctionnelle
- Mais comment représenter ce qui est important?

eukaryotic TATA-box promoter sequences:

TCTATACAATGGC
ACTATATAATGGA
TGAATACATTGGG
TCTATACAATGCT
ACTATAATATTGC
TCTATATAATAGC

Consensus

- A partir d'un alignement, on détermine les résidus les plus fréquents à chaque position. Si la fréquence dépasse un certain seuil: séquence inclue dans le consensus. P. ex. consensus 90%:
- Représentation peu performante: faible spécificité / sensibilité

Position	1	2	3	4	5	6	7	8	9	10	11	12
% A	19.4	23.4	5.0	83.5	4.4	89.2	71.0	84.8	45.0	35.7	15.5	18.5
% C	22.7	34.0	11.0	1.3	3.3	0.8	0.8	2.9	3.4	14.0	36.5	37.0
% G	26.5	30.8	4.5	1.4	0.9	1.7	0.5	9.5	16.4	38.4	36.3	30.4
% T	31.4	11.7	79.5	13.9	91.4	8.4	27.7	2.8	35.2	11.8	11.7	14.1
Consensus			T	A	T	A	W	A	D	R		

Expressions régulières

- Un chaine de caractères décrivant un ensemble des séquences, avec des alternatives possibles à chaque position. c'est la méthode utilisée dans PROSITE. Exemple de descripteur PROSITE:
- `[AC]-x-V-x(4)-{ED}`
- x: N'importe quel aa
  []: choix entre plusieurs aa
  {}: Tous, sauf les aa mentionnés
  (x,y): Répétition x à y fois

★ Semblables en principe, les langages utilisés dans Prosite et dans les expressions régulières Unix diffèrent dans les détails.

Expressions régulières Unix:

^ Le début d'une ligne
. Tout caractère (sauf newline)
$ La fin d'une ligne
| Choix. A|B: A ou B
() groupement
[] Classe de caracteres. [AGUC]: A,G,U ou C
\ Avant un caractère spécial
* 0 fois ou plus
+ une fois ou plus
? une fois ou zero
{n} exactement n fois
{n,} au moins n fois
{n,m} de n a m fois

Profil ou Matrice poids-position (Position Weight Matrix)

- Plus subtil que les consensus: Pour chaque position de l'alignement, on détermine la fréquence d'observation des différents résidus.
- Ceci est résumé dans un tableau qui donne pour chaque position les fréquences des 20 a.a. (ou 4 bases)
- Une matrice de score est calculée à partir du tableau, selon la formule:
  $S_{b,i} = log(F_{b,i} / F_b)$ (Fb est la fréquence observée dans le génome analysé)
- La recherche est effectuée en faisant glisser une fenêtre sur la séquence à analyser et en calculant le score total à chaque position de la fenêtre.

Exemple de PWM

	A	G	G	A	T	C	T	C	T
	A	A	C	C	A	C	G	G	A
	A	A	C	G	T	C	G	C	A
	A	G	G	T	A	C	T	G	T
	A	A	C	A	T	C	A	A	T
	A	A	G	T	T	C	T	C	T
A	6	4	0	2	2	0	1	1	2
C	0	0	3	1	0	6	0	3	0
G	0	2	3	1	0	0	2	2	0
T	0	0	0	2	4	0	3	0	4

log(0): remplacé par pénalité fixe ou selon modèle

log(Fb,i /Fb)

0,6	0,43	####	0,12	0,12	####	-0,18	-0,18	0,125
####	####	0,3	-0,18	####	0,6	####	0,3	######
####	0,12	0,3	-0,18	####	####	0,12	0,12	######
####	####	####	0,12	0,43	####	0,3	####	0,426

A A C C A C G G A A A C C A C G G A A A C C

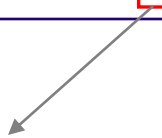
+0,6 -10 +0,3 -0,18 +0,12 -10 -0,18 +0,3 -10

Score=-29

Problème avec les jeux de séquences trop petits

Un jeu de séquences d'entrainement:

```
TC t GGCTGGT caaac- GGA a CCAA gtccgtcttcctgagaggt--- TTGG TCC CCTTCA ACCAGCT a CA
TG t GGCTGGT caaac- GGA a CCAA gtcaggtgtttctgtgaggt-- TTGG TCC CCTTCA ACCAGAC t AT
TG t GGCTGGT aaaac- GGA a CCAA gtcaggtgtttttgtgaggt-- TTGG TCC CCTTCA ACCAGCT a TG
TG c GGCTGGT gaaaa- GGA a CCAC atcaacccagaaaaaggat--- TTGG TCC CCTTCA ACCAGCC g CA
TA t GGCTGGT caaac- GGA a CCAA gtccgtcttccttagaggt--- TTGG TCC CCTTCA ACCAGCT a TT
AG t TGCTGGT aaaac- GGA a CCAA gtcgggtgtttgcgagaggt-- TTGG TCC CTTTCA ACCAGCT a CT
TG t GGCTGGT caaat- GGA a CCAA gtcaggtgtttctgcgaggt-- TTGG TCC CCTTCA ACCAGCT a CT
```



100% C

Autres scores = log (obs/expected) = valeur arbitraire!

Que faire si on trouve un G ?

Pseudocomptes

- Principe: remplir les colonnes avec des comptages attendus, selon un modèle raisonnable
- Exemple: la colonne *c* contient 7 C, on sait que T remplace souvent C. Insérons quelques T.
- Il nous faut des matrices de substitution!

Henikoff & Henikoff pseudocounts

$$b_{ca} = B_c + \sum_{i=1}^{16} \text{Probability}(\text{i}|\text{column c}) * \text{Probability}(\text{a}|\text{i})$$

Total # pseudocounts in column c

a substituted by i

Counts of base a in column c

Avec exemple précédent:

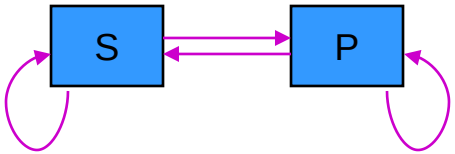
La colonne *c* est 100% C
Probabilité(C)=1, autres = 0

Nb de A = Bc * 1 * Probabilité (C | A)
Nb de T = Bc * 1 * Probabilité (C | T), etc.

Chaînes de Markov (MM: Markov Models)

- **Imaginons un climat à deux états:**
  - **P=Pluie**
  - **S=Soleil**

- **S P S S P P P S P P P ?**
  - **Quel est le temps le plus probable demain?**
  - **Solution: mesurer les probabilités de transition sur un ensemble d'entraînement, puis les appliquer à la séquence observée**
  - **Ordre 1: P -> ?**
  - **Ordre 5: P S P P P -> ?**



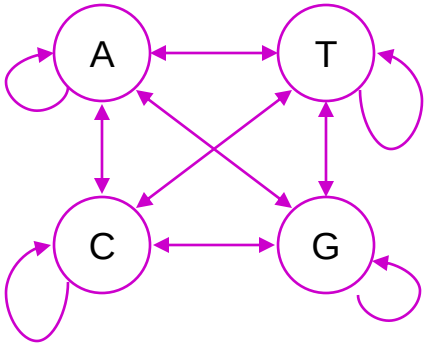
- Une *chaine de Markov* est une collection d'ETATS correspondant chacun à une observation, où le passage d'un état à l'autre (flèches) est associé à une probabilité.
- les probabilités de passage d'un état à l'autre sont appelées *probabilités de transition*.
- Le système a besoin d'une phase d'*entrainement* pour déterminer les probabilités de transition.

Modèles Markoviens et séquences biologiques

- Lorsqu'on sait que la succession des nt est importantes (par ex. dinucléotides (CpG), trinucléotides (codons), etc.), on veut un *modèle* dans lequel la probabilité d'une base dépende des bases précédentes.
- La base d'entraînement est constituée d'un ensemble de séquences de la même famille à reconnaître (par exemple: exons).
- Pour calculer la probabilité qu'une séquence appartienne à cette famille, il suffit d'observer les transitions apparaissant dans cette séquence, puis de se reporter au MM pour obtenir les probas. La probabilité finale est le produit des probabilités de transition.

Ici, 4 états.

Dans une *Chaine d'ordre k* : l'état suivant dépend des k états précédents. Par exemple, ordre 5: probabilité d'observer un A après avoir vu AAUAA.



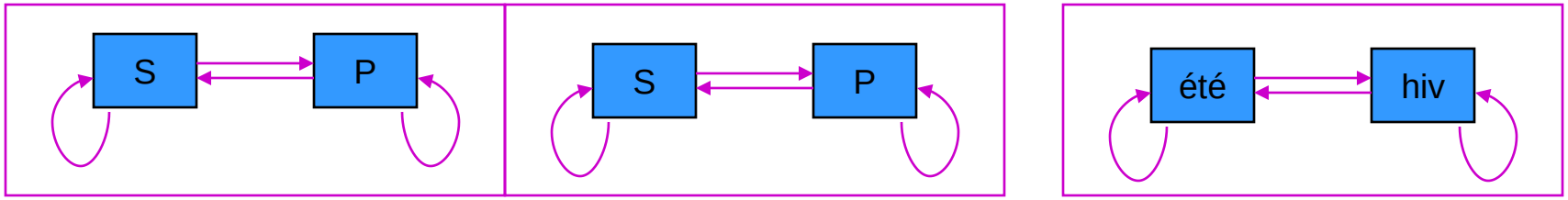
Illustr. adaptée de Durbin et al. Biological Sequence Analysis, Cambridge University Press.

Modèles de Markov cachés (HMM)

- S'il y a plusieurs type d'objets à identifier (cf été/hiver, ou intron/exon/promoteur, etc.) il faut employer *plusieurs MM.*
- Si l'on veut détecter le passage d'un modèle à l'autre, il faut ajouter à chaque état d'un modèle une probabilité de passer à un état de l'autre modèle. Il n'y a plus de correspondance directe entre les bases et les états. Par ex. la base G peut se trouver dans un modèle ou dans l'autre. On dit alors que le modèle est *caché*
- Exemple: modélisation d'un gène complet (programme Genscan): MM pour intron/exon/intergénique, et transitions d'un MM à l'autre.

Modèles de Markov cachés (HMM)

- L'information que l'on cherche n'est pas un évènement de la chaîne.
- Par exemple: S P S S P P P S P P P -> est-on en été ou en hiver?
- Dans ce cas, il faut entraîner deux MM (été et hiver) et évaluer en plus les transitions été/hiver:



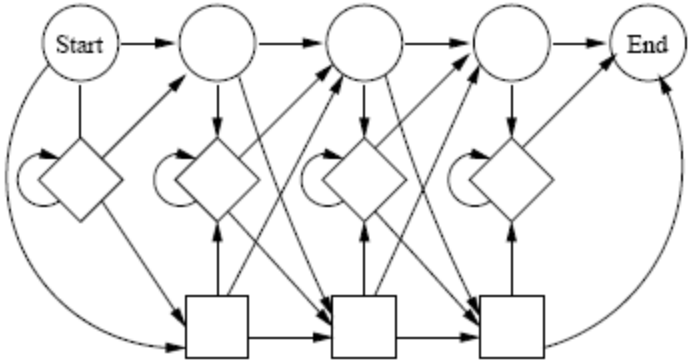
été hiver HMM

Recherche avec HMM

- L'algorithme de Viterbi est utilisé pour découvrir la suite d'état cachés la plus probable dans une séquence donnée.
- Algorithme de programmation dynamique
- Nécessité de tester un grand nombre de chemins possibles: lent!

Profile-HMM

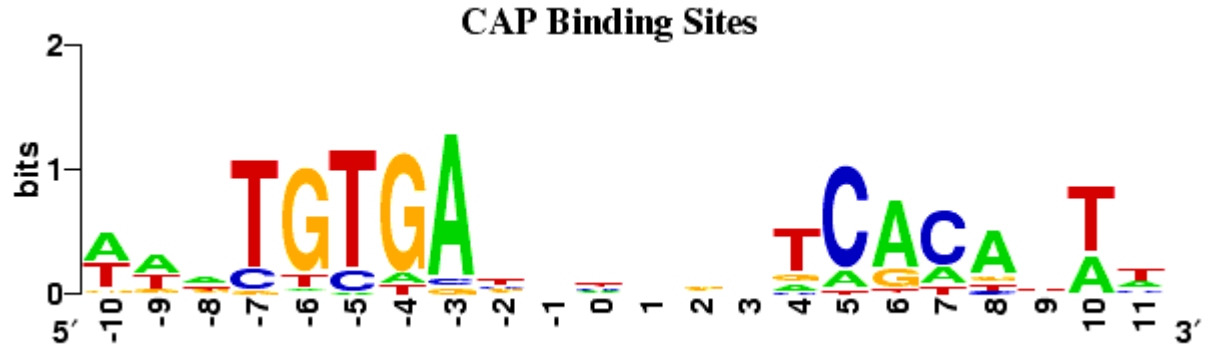
- Un HMM de taille définie où chaque position a ses propres probas de transition
- Typiquement produit à partir d'un alignement multiple
- Combine les avantages des PWM et de HMM: les positions importantes sont identifiées, les transitions sont prises en compte

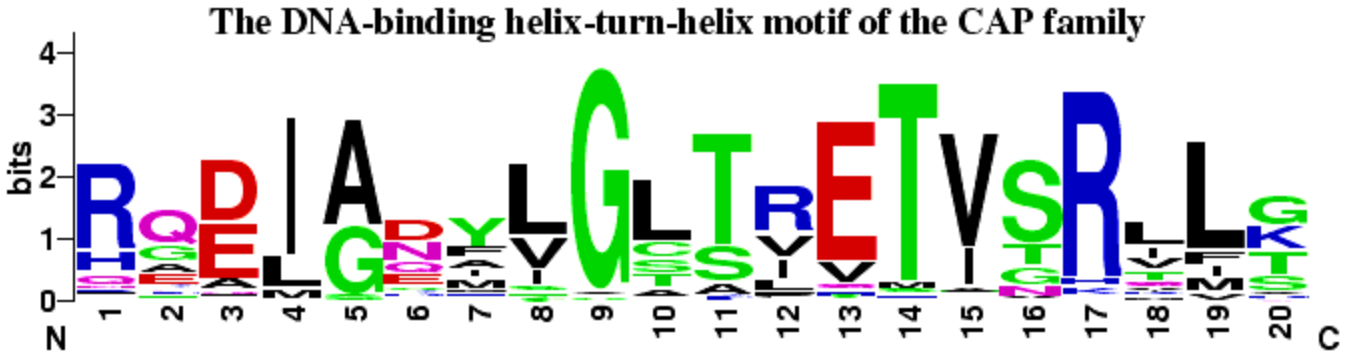


Structure d'un profile-HMM, Colin Cherry Univ. of Alberta

Sequence logos

(Schneider TD, Stephens RM. NAR. 1990)





Entropie et contenu en information

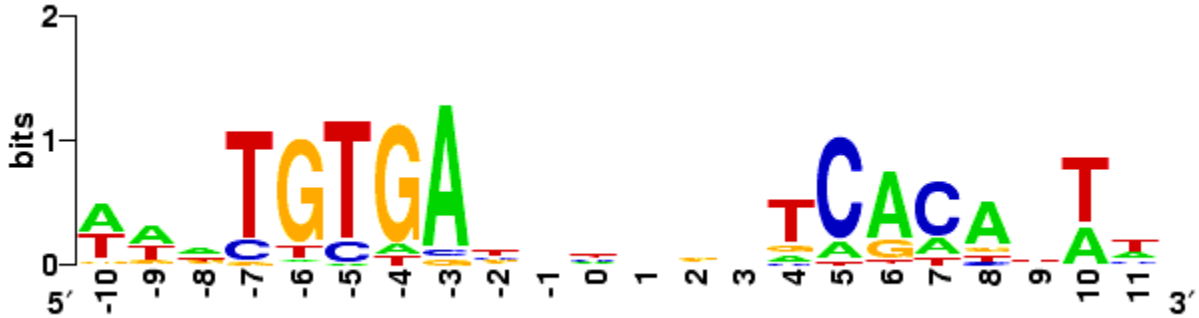
- Entropie de Shannon à la position *i*:

$$H_i = - \sum_{a=A,T,G,C} f_{a,i} \log(f_{a,i}).$$

$f_{a,i}$: fréquence lettre *a* à la position *i*.

- Hauteur des lettres proportionnelle à:

$$f_{a,i} * H_i$$



Estimer la performance des recherches de motifs

Sensibilité et Spécificité

- ***Sensibilité*: La capacité à détecter les vraies instances de l'objet recherché (« vrais positifs »).**
- ***Spécificité*: La capacité à rejeter les fausses instances (« faux positifs »).**

Sensibilité: $SN = \frac{TP}{TP+FN}$

Total « vrais » objets

Specificité : $SP = \frac{TP}{TP+FP}$

Total prédictions

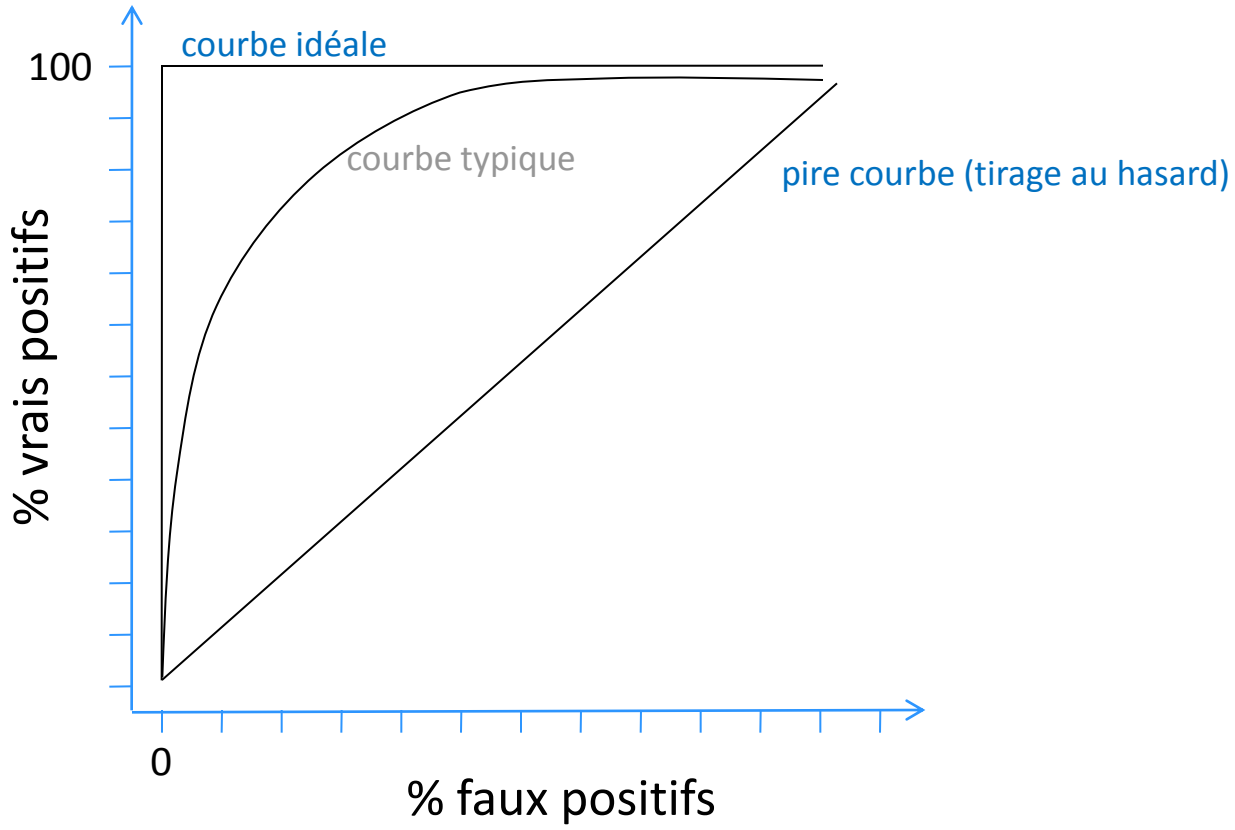
TP: “Vrai positif”

FP: “Faux positif”

FN: “Faux négatif”

La courbe ROC*

*Receiver Operator Characteristic

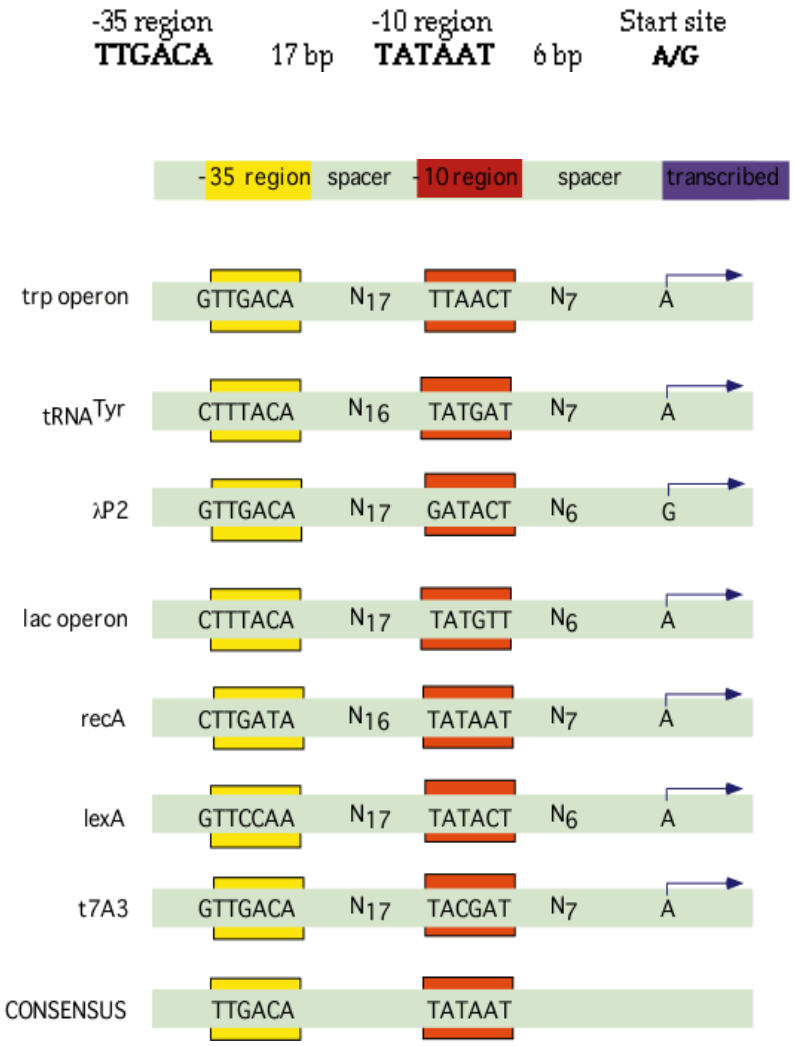


Application à la détection des gènes

Les gènes microbiens

Le promoteur microbien

- la région reconnue par la polymérase à ARN. Juste en amont du site d'inititiation de la transcription.
- Comprend trois éléments: la boite de Pribnow (TTGACa) vers -35, la boite TATA (TAtAAT) vers -10 et le site d'initiation de la transcription
- Pribnow E.coli (%)= T82 T84 G78 A65 C54 a45
- TATA E coli (%) = T80 A95 T45 A60 a50 T96
- Ce sont ces 3 séquences que les facteurs sigma reconnaissent (lient à la fois le promoteur et l'ARN polymérase).

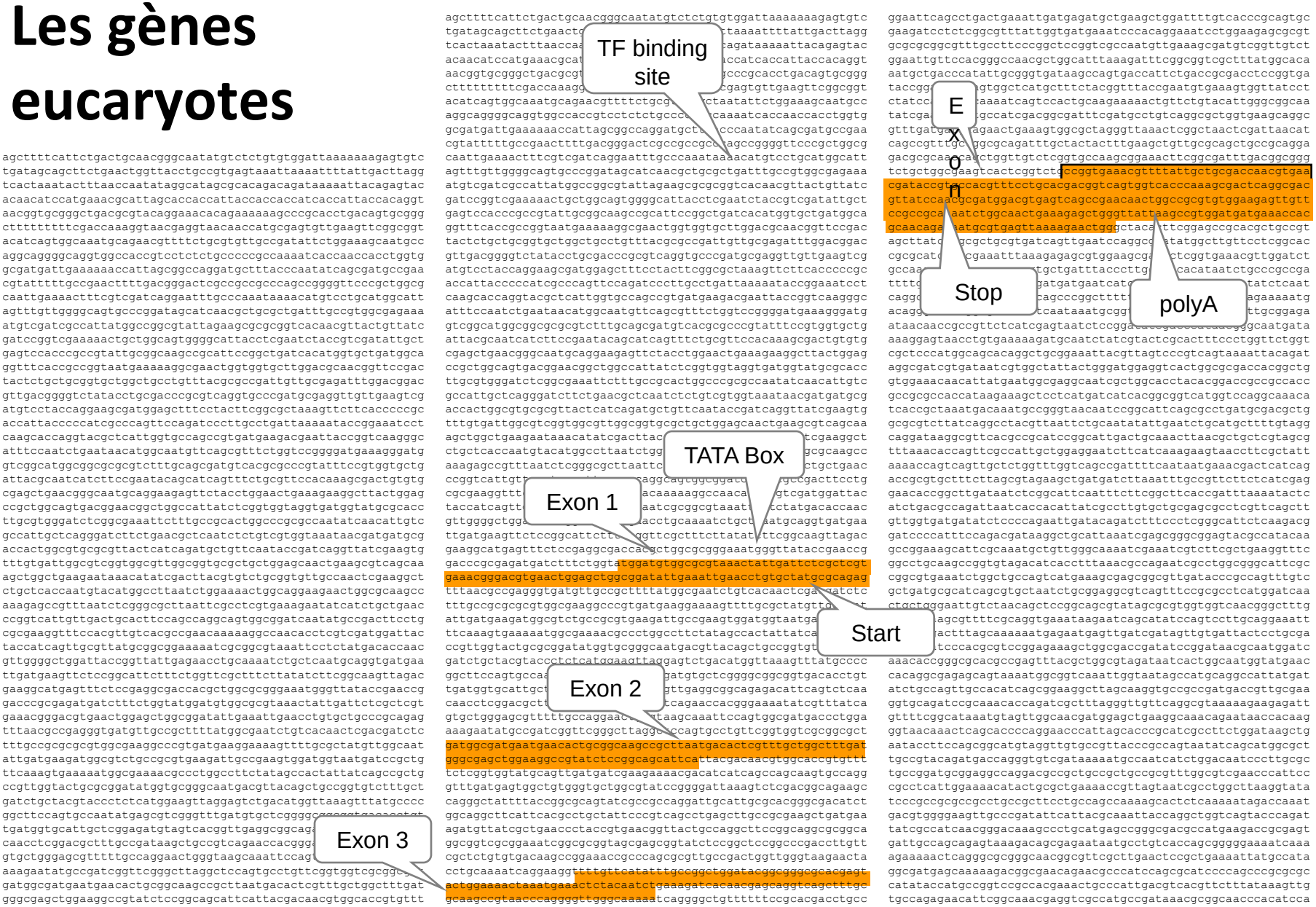


Crédit:
http://www.blc.arizona.edu/marty/411/Modules/prokrom.html

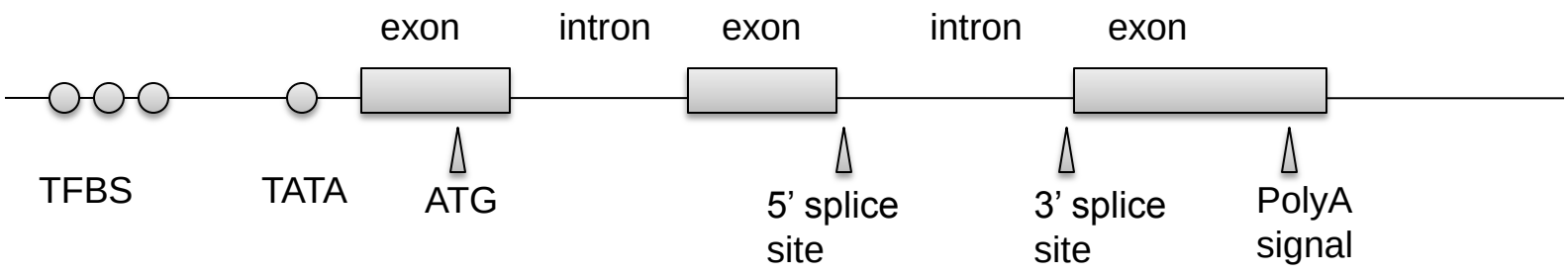
La séquence de Shine-Dalgarno (SD), ou Ribosome Binding Site (RBS)

- (aGGAGGu) environ 10 nt avant codon ATG.
- Région riche en purine de 3 à 10 nt qui permet au ribosome de distinguer le véritable codon initiation d'autres AUG fortuits.
- S'apparie avec une région très conservée et riche en pyrimidine de l' ARNr 16S. Cet appariement aligne le codon AUG au site P de l'ARNr.

Les gènes eucaryotes



Motifs des gènes eucaryotes



Sites de liaison aux facteurs de transcription

- Séquences courtes de 6-20 nt affectant généralement l' efficacité de l'initiation de la transcription.
  - Boîte CCAAT
  - Sp1 box
  - CRE
  - AP2 box
    etc..

La boîte TATA eucaryote

- La sequence d'environ 8 paires de bases contient pratiquement que des adenines et thymines, et tend a être encadrée par des séquences riches en guanine et cytosine, ces dernières pouvant participer à la fonction du promoteur.
- TATA consensus: *GTATAAAAGGCGGGG* (mais beaucoup de variation)
  Le consensus de la TATA box est faible et cet élément est même absent dans de nombreux promoteurs.

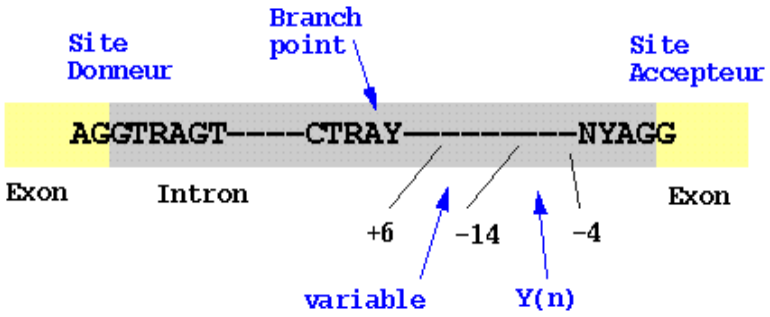
General eukaryotic TATA-box model derived from 860 unrelated promoter sequences:

Position	1	2	3	4	5	6	7	8	9	10	11	12
% A	19.4	23.4	5.0	83.5	4.4	89.2	71.0	84.8	45.0	35.7	15.5	18.5
% C	22.7	34.0	11.0	1.3	3.3	0.8	0.8	2.9	3.4	14.0	36.5	37.0
% G	26.5	30.8	4.5	1.4	0.9	1.7	0.5	9.5	16.4	38.4	36.3	30.4
% T	31.4	11.7	79.5	13.9	91.4	8.4	27.7	2.8	35.2	11.8	11.7	14.1
Consensus			T	A	T	A	W	A	D	R		

Les jonctions intron-exon

- Motifs utiles: jonctions, point de branchement et région riche en pyrimidines
- Le plus conservé: GT et AG en 5' et 3' de l'intron
- 98.1% des introns humains possèdent les GT et AG. 0.76% ont GC-AG. 0.1% ont AT-AC.

Signaux jonction intron-exon vertébré



Fin du gène: le signal de polyadénylation

- Dans l'exon terminal: signaux nécessaires à la maturation de pré- mRNA en mRNA: clivage et polyadénylation.
  1. Un héxamère AAUAAA ou AUUAAA (et parfois des variants présentant une mutation sur une base: AGUAAA, UAUAAA, CAUAAA, etc.), 10 à 30 bases en amont du site de clivage (en moyenne 17 bases). L'un ou l'autre des variants est observé dans plus de 90% des gènes.
  2. Au site de clivage: un dinucléotide CA, assez mal conservé.
  3. 20 à 40 bases après le site de clivage (donc toujours sur le pré-mRNA): une région riche en GU, de séquence variable.

A la recherche des motifs dans les génomes

Quels sont les signaux les plus forts?

Peut-on identifier les gènes par leurs motifs?

La recherche de motifs seuls est insuffisante

Nombreux faux positifs: spécificité faible

- La plupart des auteurs ont tenté d'exploiter à la fois la présence d'un cadre de lecture et des autres motifs: promoteur (TATA, ilots CpG), jonction intron-exon (donneur, accepteur), signal de polyadenylation, etc.
- Même si les motifs étaient parfaitement conservés, ils sont peu spécifiques.
- Les boîtes TATA et autres éléments des promoteurs, ainsi que les signaux d'épissage sont également peu spécifiques: le motif donneur AxGT(A/G)xG est observé 559 fois dans un contig humain de 67kb ne contenant que 7 exons.
- La reconstruction du gène complet ajoute encore une source d'erreur: risque d'oublier des exons ou de mélanger ceux provenant de deux gènes.

Les motifs de composition

- De nombreux motifs ne s'expriment pas par une séquence consensus spécifique, mais par un biais de séquence.
  - Biais de GC
  - Biais de codon etc.

Les ilôts CpG

- Les ilôts CpG sont des zones riches en dinucléotide CG, fréquemment associées aux régions 5' des gènes de vertébrés
- L'ilôt s'étend sur le promoteur et l'exon 1 (ou 1 et 2)
- Fréquence attendue du dinucléotide CpG = 4% (0.21x0.21), mais fréquence observée: un cinquième de cette valeur. Pourquoi?
  - Méthylation naturelle des CpG et réparation en TpG par déamination
  - Au niveau du promoteur: protection des CpG. Donc Fréquence normale.
- Typiquement 1-2kb de longueur. Environ 70% G+C (contre 40% dans le reste du génome humain)
- Les îlots CpG sont associés à tous les gènes housekeeping (constitutifs) et à 40% des gènes tissu-spécifiques

Open Reading frames

- On trouve en moyenne un ORF de 150 nt (la taille typique d'un ORF) tous les kilobases, alors qu'il n'en existe en fait qu'un tous les 10kb dans les génomes de vertébrés!
- 9 FP pour un TP!

Biais d'usage de codons

- **l'abondance et l'utilisation différente des acides aminés entraine naturellement des fréquences différentes pour chaque codon**
- **Mais lorsque les codons synonymes ne sont pas employés avec la même fréquence, on parle de biais d'usage des codon (codon usage bias). Découvert initialement chez la levure et coli.**
- **Les biais de codon diffèrent d'une espère à l'autre, selon les contraintes propres à chaque espèce:**
  - **la nature du code génétique**
  - **les ARNt disponibles**
  - **D'éventuelles contraintes évolutives (GC content, taux de mutation..)**
  - **Une préférence pour les séquences purine-N-pyrimidines**
  - **Une préférence pour les codon/anticodons riches en GC qui minimiserait les erreurs de traduction**
- **A coté du biais propre à l'espèce, il existe des biais propres à certains gènes. Généralement les gènes les plus exprimés sont les plus biaisés, les codons les plus utilisés étant ceux pour lesquels les ARNt sont les plus nombreux (codons sélectionnés pour une plus grande vitesse de traduction)**

Annotation de gènes microbiens: Genemark

Utilise les modeles markoviens dans sa version de base (recherche des regions codantes seulement). Puis l'idee a ete integrée dans un modèle markovien cache (HMM) plus complexe tenant compte des informations du promoteur et du RBS.

Results of GeneMark.hmm predictions for 10 complete bacterial genomes*

Genome	Genes annotated	Genes predicted	Annotated genes predicted by GeneMark.hm GeneMark (%)	Correct 5' end diction of annotated genes (%)	Potential new genes (%)
A.fulgidus	2407	2530	98.0	73.1	15.1
B.subtilis	4101	4384	97.2	77.5	9.8
E.coli	4288	4440	97.3	75.4	8.2
H.influenzae	1718	1840	96.2	86.7	10.2
H.pylori	1566	1612	95.6	79.7	8.7
M.genitalium	467	509	98.3	78.4	17.3
M.jannaschii	1680	1841	99.2	72.7	12.9
M.pneumoniae	678	734	95.9	70.1	13.6
M.thermoauto	1869	1944	97.5	70.9	8.6
Synechocystis	3169	3360	98.5	89.6	9.4
Average	**21943**	**23194**	**97.3**	**78.1**	**10.4**

Sensibilité

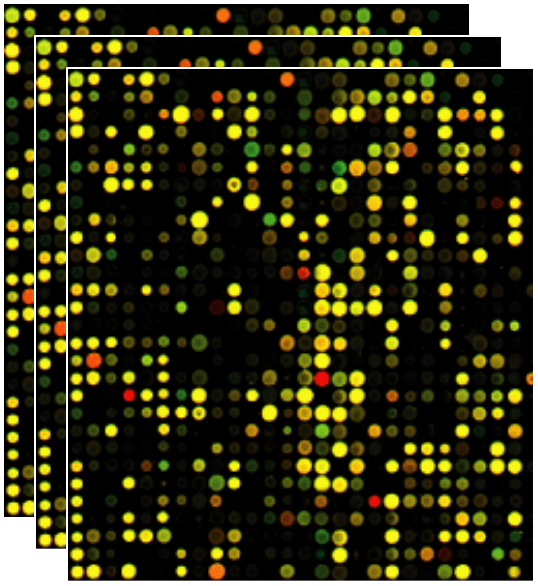
The second and third columns show the number of genes annotated in GenBank and the number of genes predicted, respectively.
The "Annotated genes predicted" column presents the percentage of annotated genes which were predicted by GeneMark and GeneMark.hmm
The "Correct 5' end prediction of annotated genes" column shows the percentage of genes whose starts were predicted exactly.
"Potential new genes" is the fraction of predicted genes for which no annotated analog was found. All measures are expressed in percent.
* **Reference:** A. Lukashin and M. Borodovsky, GeneMark.hmm: new solutions for gene finding, **NAR**, 1998, Vol. 26, No.4, pp 1107-1115.

Gènes eucaryotes

- GENSCAN (Burge & Karlin, J. Mol. Biol. 268, 1-17, 1998.) Utilise les HMM. Plusieurs modèles sont employés pour exons, introns, promoteurs, etc. Sensibilité et Spécificité autour de 80% pour les exons correctement prédits. Beaucoup moins bon pour la prédiction de gènes complets.

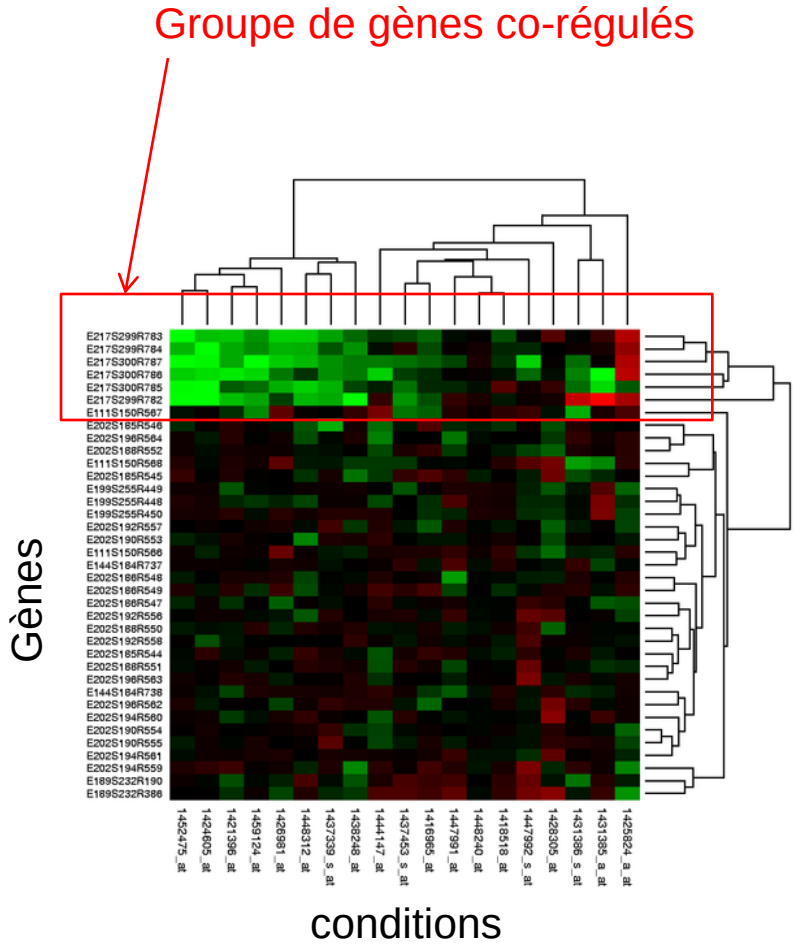
Découvrir des motifs inconnus

Exemple: groupes de gènes co-régulés identifiés par analyse de transcriptome

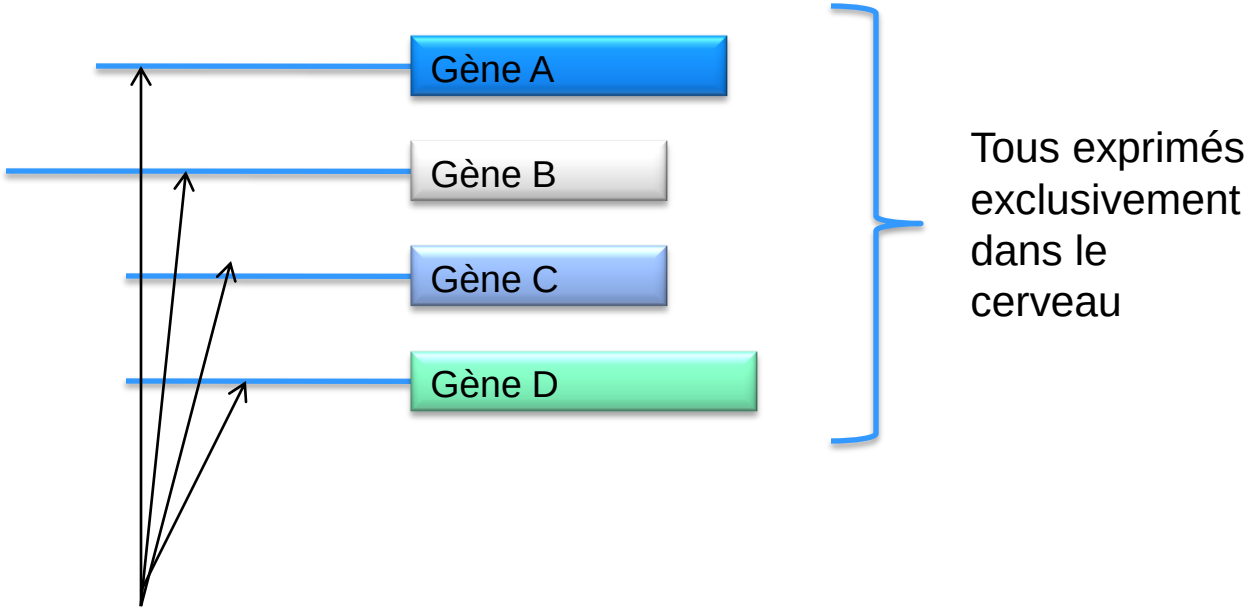


Puces à ADN ->
expression des gènes
dans différentes conditions

Clustering



Découverte de motifs régulateurs

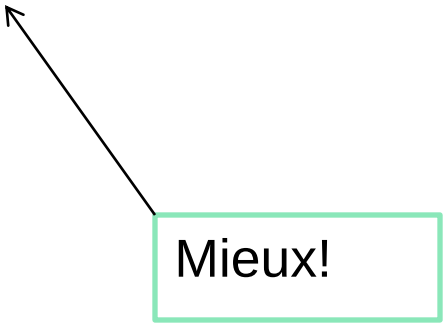


Séquence régulatrice commune?

(= Site de fixation pour un Facteur spécifique du cerveau)

Recherche par comptage de mots

- Idée: rechercher des mots de k lettres surreprésentés dans la région d'intérêt
- Mais surreprésentés par rapport à quoi?
  - Une composition uniforme 25%A/25%T/25%G/25%C?
  - La composition du génome étudié?
  - La composition des régions étudiées?



Diapos sur comptage de mot inspirée du cours de Jacques van Helden:
http://rsat.scmbb.ulb.ac.be/rsat/course/

Combien de fois s'attend-on à trouver un mot *w* donné?

- $Ew = Pw \times T$

  Ew: nombre d'occurrences attendu

  Pw: probabilité d'observer w à chaque position

  T: nombre de positions

T dépend des recouvrements

Avec recouvrements

Sans recouvrements

Pw dépend de l’autocorrélation du mot

CCTAA

Sans autocorrélation: Pw = produit des fréquences des bases

Autocorrélation: Pw nécessite correction

Comment donner un score au nombre d'occurrences observées?

- Ratio
- Log de vraissemblance
- Z-score / distribution normale
- Distribution binomiale

Ratio

- r = Cw / Ew

> *Cw:* nombre d'occurrence attendu de *w*
> *Ew:* nombre d'occurrences observé de *w*

- Facile, mais surestime l'importance des mots rarement attendus

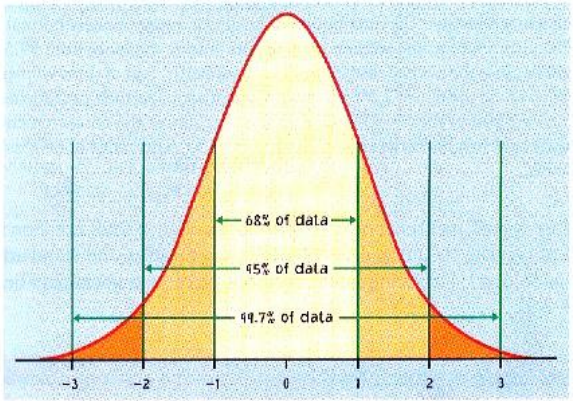
Log de vraissemblance

- *K = Fw log (Fw / Pw)*

Fw: fréquence observée de *w*
Pw: fréquence attendue de *w*

- Mieux, mais donne un score « brut », pas une probabilité

Z-score



From: http://www.sci.sdsu.edu/class/psychology/psy271/Weeks/psy271week06.htm

- *Z = (Cw – Ew) / Sw*

Cw: nombre d'occurrence attendu de *w*
Ew: nombre d'occurrences observé de *w*
Sw: variance

- Peut se traduire en P-value grâce à la distribution normale, mais:
- Nécessite un grand jeu de données
- La distribution normale/gaussienne est approximative, surtout dans les valeurs extrêmes

Loi binomiale et P-value

- Donne une probabilité
- Faiblesse: ne gère pas les mots chevauchants (autocorrélés)

Exactement *Cw* fois

$$P(X = C_w) = \frac{T!}{C_w!(T - C_w)!} p^{C_w} (1-p)^{T-C_w} = C_T^{C_w} p^{C_w} (1-p)^{T-C_w}$$

Plus de *Cw* fois

$$P(X \geq C_w) = \sum_{i=C_w}^{T} \frac{T!}{i!(T-i)!} p^i (1-p)^{T-i}$$

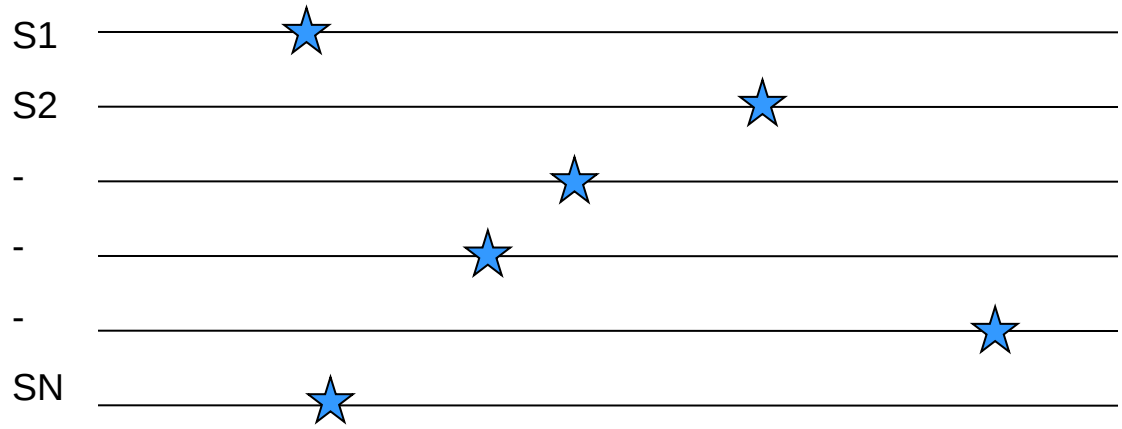
Moins de *Cw* fois

$$P(X \leq C_w) = \sum_{i=0}^{C_w} \frac{T!}{i!(T-i)!} p^i (1-p)^{T-i}$$

Gibbs sampling

(CE Lawrence et al. Science, vol. 262, 1993)

- Principe:
  - N sequences



But: Découvrir ★

Gibbs sampling: outils

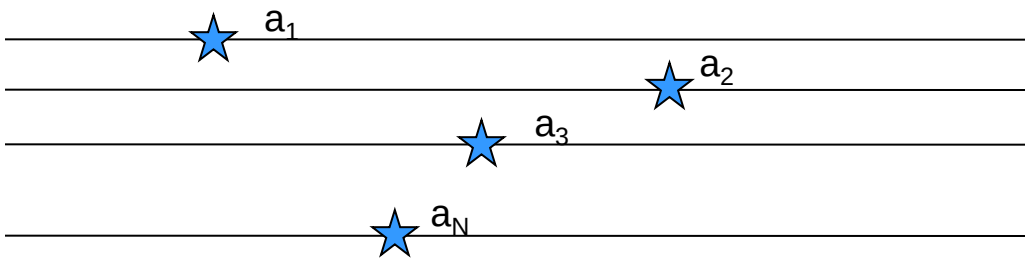
PWM de taille W pour stocker le motif en cours de recherche

	1	2							W
A									
C									
G									
T									

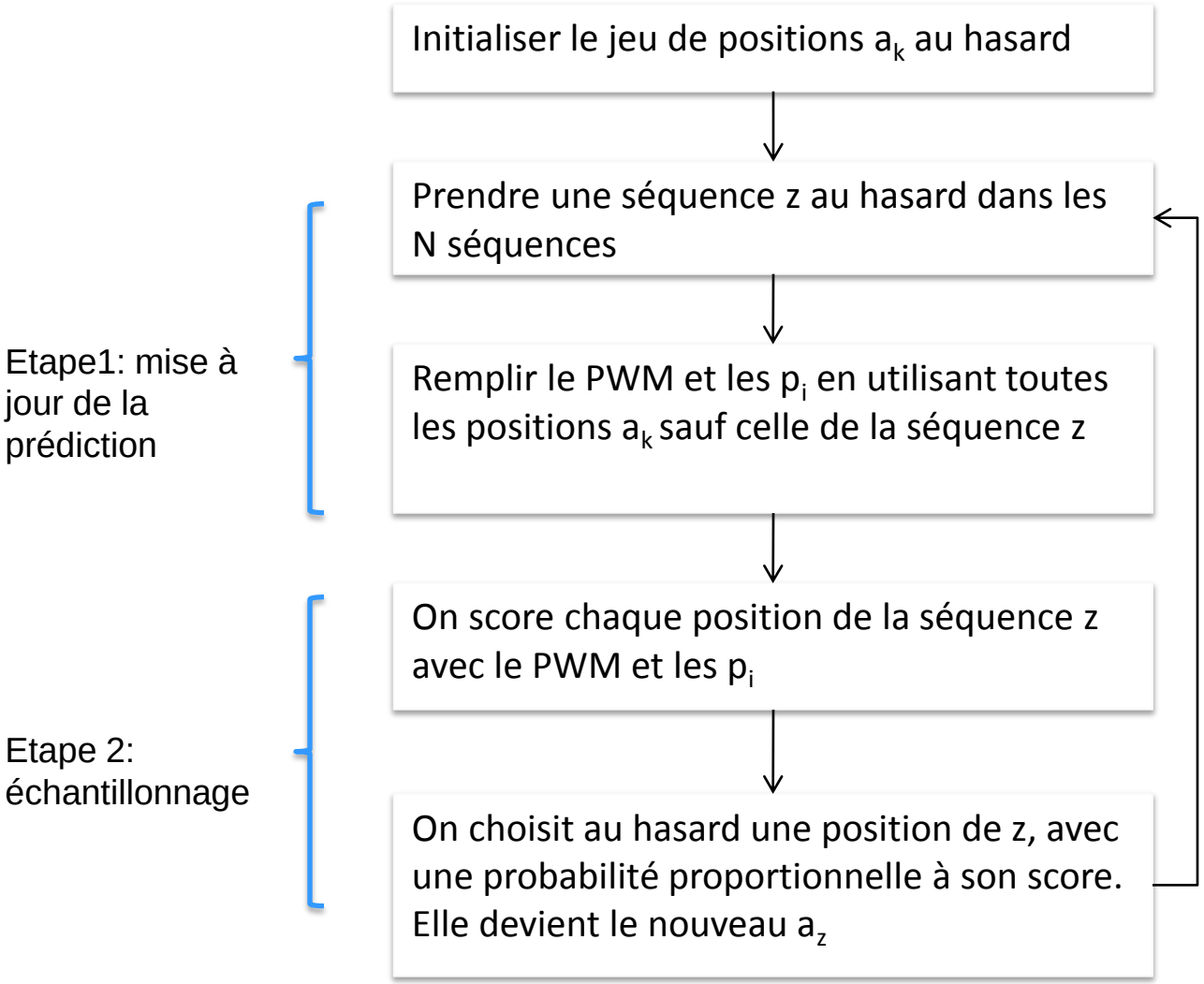
Tableau pour stocker les fréquences de fond des 4 résidus

p_1	p_2	p_3	p_4

Un jeu de positions $a_1..a_N$ indiquant la position du motif dans chaque séquence



Gibbs sampling: algorithme



Logiciels

HMMER

S. Eddy ; hmmer.janelia.org

Extrait de la doc:

Let's assume you have a multiple sequence alignment of a protein domain or protein sequence family. To use HMMER to search for additional remote homologues of the family, you want to first build a profile HMM from the alignment. The following command builds a profile HMM from the alignment of 50 globin sequences in **globins50.msf:**

> hmmbuild globin.hmm globins50.msf

As an example of searching for new homologues using a profile HMM, we'll use the globin model to search for globin domains in the example *Artemia globin sequence in **Artemia.fa:***

> hmmsearch globin.hmm Artemia.fa

HMMER output

```
Parsed for domains:
Sequence Domain  seq-f seq-t    hmm-f hmm-t       score  E-value
-------- ------- ----- -----    ----- -----       -----  -------
S13421     7/9     932  1075 ..     1   143 []    76.9  7.3e-24
S13421     2/9     153   293 ..     1   143 []    63.7  6.8e-20
S13421     3/9     307   450 ..     1   143 []    59.8  9.8e-19
S13421     8/9    1089  1234 ..     1   143 []    57.6  4.5e-18
S13421     9/9    1248  1390 ..     1   143 []    52.3  1.8e-16
S13421     1/9       1   143 [.     1   143 []    51.2    4e-16
S13421     4/9     464   607 ..     1   143 []    46.7  8.6e-15
S13421     6/9     775   918 ..     1   143 []    42.2    2e-13
S13421     5/9     623   762 ..     1   143 []    23.9  6.6e-08
```

```
Alignments of top-scoring domains:
S13421: domain 7 of 9, from 932 to 1075: score 76.9, E = 7.3e-24
                  *->eekalvksvwgkveknveevGaeaLerllvvyPetkryFpkFkdLss
                     +e a vk+ w+ v+ ++  vG  +++ l++ +P+ +++FpkF d+
      S13421   932    REVAVVKQTWNLVKPDLMGVGMRIFKSLFEAFPAYQAVFPKFSDVPL 978

                  adavkgsakvkahgkkVltalgdavkkldd...lkgalakLselHaqklr
                   d++++++ v +h   V t+l++ ++ ld++ +l+   ++L+e H+  lr
      S13421   979 -DKLEDTPAVGKHSISVTTKLDELIQTLDEpanLALLARQLGEDHIV-LR 1026

                  vdpenfkllsevllvvlaeklgkeftpevqaalekllaavataLaakYk<
                  v+   fk +++vl+  l++ lg+ f+  ++ +++k+++++++ +++  +
      S13421  1027 VNKPMFKSPGKVLVRLLENDLGQRPSSFASRSWHKAYDVIVEYIEEGLQ  1075
```

RSA-tools

J. Van Helden: rsat.bigre.ulb.ac.be/rsat/

Extrait de la doc:

5.2.1 Counting word occurrences and frequencies

Try the following command:

oligo-analysis -v 1 -i Escherichia_coli_K12_start_codons.wc \

-format wc -l 3 -1str

Output:

```
;seq identifier exp_freq occ exp_occ occ_P occ_E occ_sig rank ovl_occ forbocc
acgtgc acgtgc|gcacgt 0.0002182431087 16 2.46 8.4e-09 1.7e-05 4.76 1 2 76
cccacg cccacg|cgtggg 0.0001528559297 11 1.72 2e-06 4.2e-03 2.37 2 0 55
acgtgg acgtgg|ccacgt 0.0002257465554 13 2.54 2.8e-06 5.9e-03 2.23 3 1 65
cacgtg cacgtg|cacgtg 0.0001299168211 10 1.46 3.3e-06 6.8e-03 2.17 4 0 100
cgcacg cgcacg|cgtgcg 0.0001322750472 10 1.49 3.8e-06 8.0e-03 2.10 5 0 50
cgtata cgtata|tatacg 0.0005113063008 17 5.76 0.00011 2.2e-01 0.65 6 1 85
agagat agagat|atctct 0.0006913890231 19 7.78 0.00047 9.8e-01 0.01 7 0 95
```

PSI-Blast (recherche itérative de profil)

- **Principe**
  - Une première séquence est recherchée dans une base de données
  - Les séquences similaires significatives sont alignées sur la séquence requête.
  - Un profil est construit
  - Ce profil est recherché dans la banque de donnée pour collecter des séquences supplémentaires, etc.
- **Avantages et inconvénients**
  - Excellent pour la recherche d'homologues éloignés.
  - Si une séquence sans parenté avec la première est collectée accidentellement, celle-ci entraine tous ses homologues avec elle au tour suivant. Le profil perd son sens.
  - Les protéines multidomaines posent le même problème
  - N’existe que pour les protéines

Output de Psi-Blast