M2 - STL

Algorithmes sur les séquences en bioinformatique

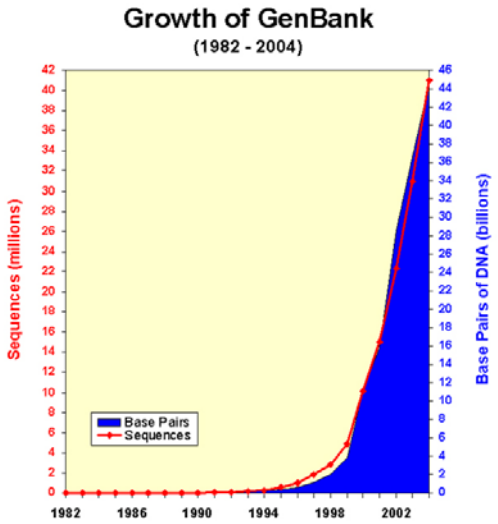
Cours 7 : Algorithmes de recherche des gènes chez les procaryotes et les eucaryotes

Alessandra Carbone
Université Pierre et Marie Curie

Prédiction de gènes: défi algorithmique

- Gène: une séquence de nucléotides codant pour une protéine

- Problème de la prédiction des gènes: Déterminer le début et la fin des gènes dans un génome



GenBank Data		
Year	Base Pairs	Sequences
1982	680,338	606
1983	2,274,029	2,427
1984	3,368,765	4,175
1985	5,204,420	5,700
1986	9,615,371	9,978
1987	15,514,776	14,584
1988	23,800,000	20,579
1989	34,762,585	28,791
1990	49,179,285	39,533
1991	71,947,426	55,627
1992	101,008,486	78,608
1993	157,152,442	143,492

GenBank Data		
Year	Base Pairs	Sequences
1994	217,102,462	215,273
1995	384,939,485	555,694
1996	651,972,984	1,021,211
1997	1,160,300,687	1,765,847
1998	2,008,761,784	2,837,897
1999	3,841,163,011	4,864,570
2000	11,101,066,288	10,106,023
2001	15,849,921,438	14,976,310
2002	28,507,990,166	22,318,883
2003	36,553,368,485	30,968,418
2004	44,575,745,176	40,604,319

La plupart des séquences n'est pas codantes. Le génome de l'homme, par exemple, ayant 3 billions de bases, a seulement le 2-3% de la séquence qui est codant.

Méthodes de recherche automatique des gènes

Prédiction de gènes: défi algorithmique

aatgcatgcggctatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgct
aatgcatgcggctatgcaagctgggatccgatgactatgctaagctgggatccgatgacaatgcatgcggc
tatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgc
taatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggatccgatgacaa
tgcatgcggctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgc
taatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgc
aagctgggatcctgcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatg
acaatgcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgcta
agctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcgg
gctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcat
gcggctatgctaagctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatg
ctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggct
atgctaagctcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaat
gcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctg
ggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctat
gcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcgg

Prédiction de gènes: défi algorithmique

aatgcatgcggctatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgct
aatgcatgcggctatgcaagctgggatccgatgactatgctaagctgggatccgatgacaatgcatgcggc
tatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgc
taatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggatccgatgacaa
tgcatgcggctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgc
taatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgc
aagctgggatcctgcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatg
acaatgcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgcta
agctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcgg
gctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcat
gcggctatgctaagctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatg
ctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggct
atgctaagctcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaat
gcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctg
ggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctat
gcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcgg

Prédiction de gènes: défi algorithmique

aatgcatgcggctatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgct
aatgcatgcggctatgcaagctgggatccgatgactatgctaagctgggatccgatgacaatgcatgcggc
tatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgc
taatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggatccgatgacaa
tgcatgcggctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgc
taatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgc
aagctgggatcctgcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatg
acaatgcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgcta
agctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcgg
gctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcat
gcggctatgctaagctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatg
ctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggct
atgctaagctcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaat
gcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctg
ggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctat
gcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcgg

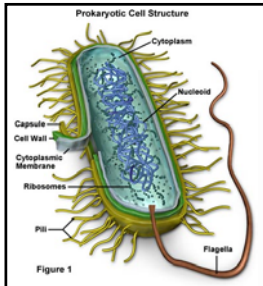
Gene!

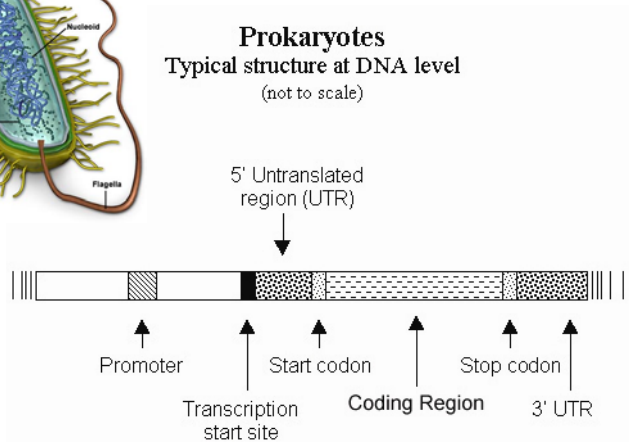
Quelques différences entre procaryotes et eucaryotes qui jouent un rôle dans la transcription

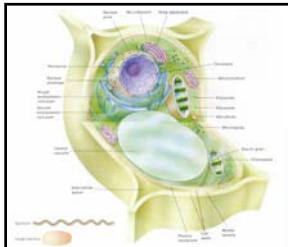
Les procaryotes se répliquent assez rapidement, donc moins de temps est dépensé pour la mise en route de mécanismes d'organisation moléculaire et de processus biologiques sophistiqués.

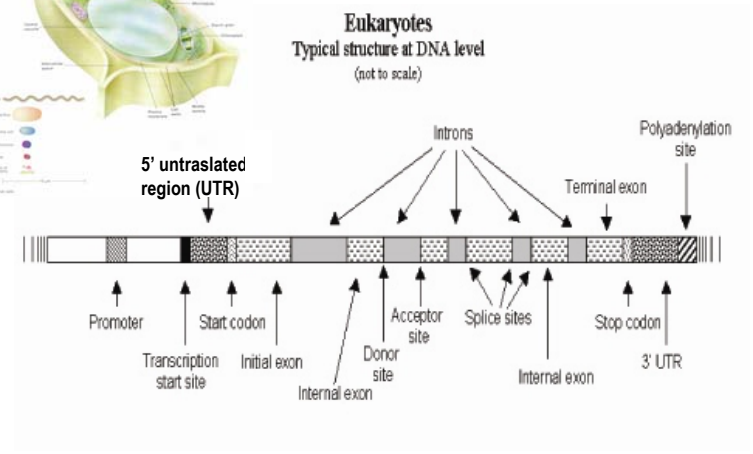
Dans les génomes procaryotes la plupart de la séquence est codante pour des protéines. Par exemple le 70% du génome de *H.influenzae* est codant.

Dans les procaryotes chaque gène est une séquence de bases où il n'y a pas d'introns.

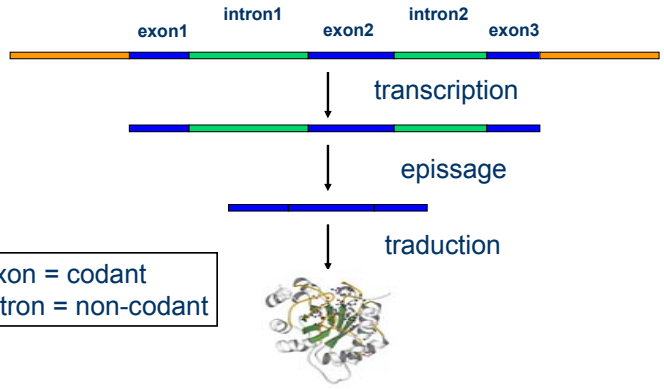








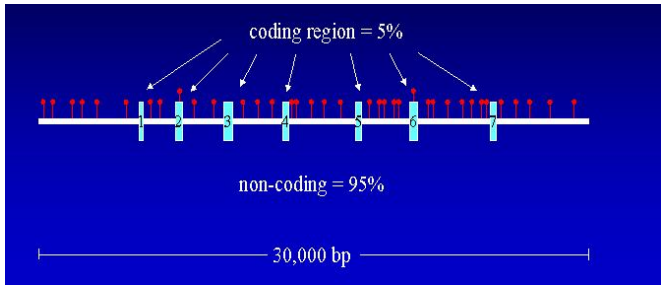
Epissage chez les eucaryotes



exon = codant
intron = non-codant

Batzoglou

Structure d'un gène chez les eucaryotes

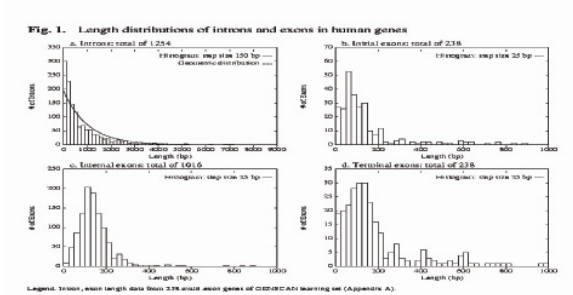


Exemple de gène moyen dans les vertébrés :

gene de 30kb
partie codante de 1-2kb
6 exons de 150bp chacun

Déviation du comportement moyen dans l'homme:

Dystrophyn	gène de 2.4MB en longueur
Facteur VIII de coagulation du sang	26 exons de taille entre 69bp et 3106bp



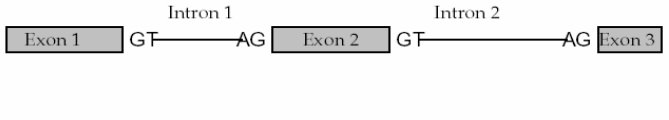
Epissage et jonction d'épissage dans les eucaryotes (et certains, rares, procaryotes!)

L'épissage est la suppression d'introns réalisée par un complexe appelé spliceosome.
Les spliceosomes sont des enzymes contenant soit des protéines que des snRNAs.
Le snRNA reconnaît le site d'épissage à travers un appariement de bases ARN-ARN.
La reconnaissance des sites d'épissages a besoin d'être précise parce que n'importe quel erreur peut translater le reading frame en provocant la génération de séquences qui n'ont aucun sens.

Plusieurs gènes ont des épissages alternatifs, cad que dans des variations différentes d'un gène, certains exons ne sont pas utilisés. Il a été estimé que cela se passe dans plus du 50% des gènes et que en moyenne chaque gène a plus que 2 variations.

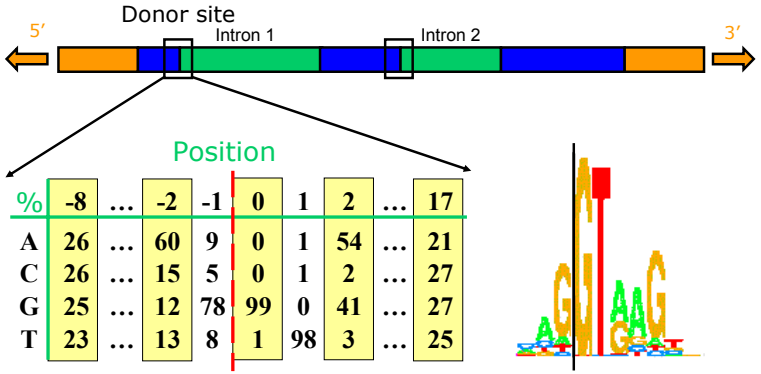
En générale, les jonctions intron-exon ont une composition nucléotidique similaire.
Des signaux pour les détecter sont difficiles à définir.

Signaux d'épissage



Les exons sont intercalés avec les introns et d'habitude suivis et précédés par GT et AG

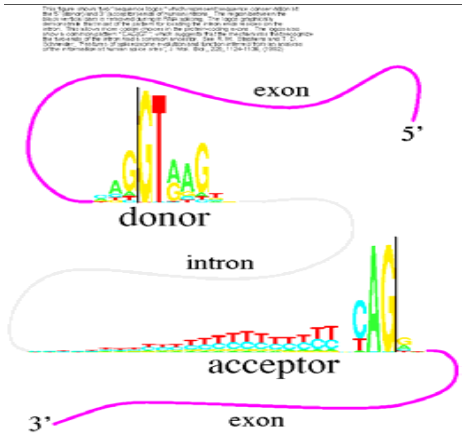
Détection d'un site d'épissage



%	-8	…	-2	-1	0	1	2	…	17
A	26	…	60	9	0	1	54	…	21
C	26	…	15	5	0	1	2	…	27
G	25	…	12	78	99	0	41	…	27
T	23	…	13	8	1	98	3	…	25

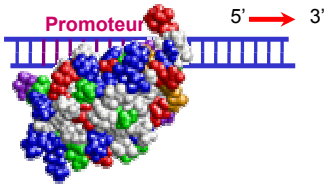
Sites d'épissage consensus

Donateur: 7.9 bits
Accepteur: 9.4 bits



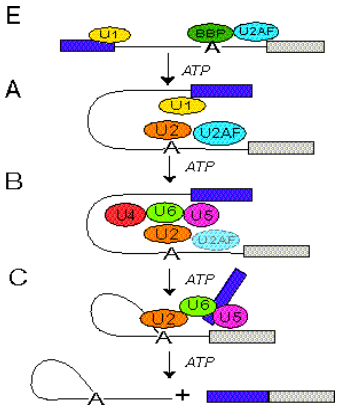
Promoteurs

- Les promoteurs sont des segments d'ADN qui précèdent les zones qui régulent le début de la transcription

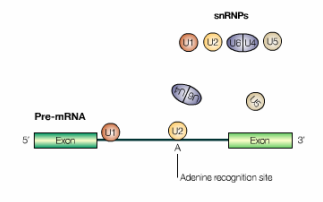


- Promoteur *attire* l'ARN-polymerase vers le site de départ de la transcription

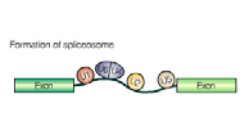
Mécanisme d'épissage



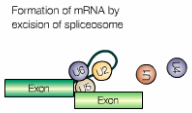
1. Activation du snRNPs



2. Formation du spliceosome



3. Excision d'un intron

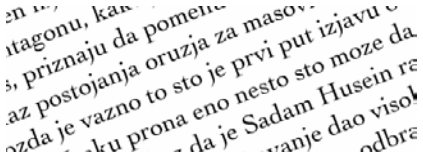


4. L'ARN messager (mRNA) est prêt

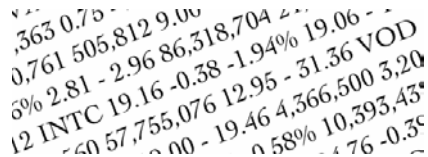
Analogie dans la prédiction des gènes

- Supposons de considérer un journal écrit dans une langue inconnue
  - Certaines pages contiennent un message encodé, disons 99 lettres à la page 7, 30 à la page 12 et 63 à la page 15.
- Comment reconnaître le message? On pourrait probablement distinguer entre de la pub et une rubrique (la pub contenant le symbole "$" assez souvent)
- L'approche à la prédiction des gènes basée sur les statistiques essaie de détecter des différences entre exons et introns.

Approche statistique: métaphore dans un langage inconnu



Basé sur les différences en fréquences de certains symboles (e.g. '%', '.', '-') et de symboles numériques peut-on distinguer entre rubrique faits divers et rubrique économique dans un journal étranger?

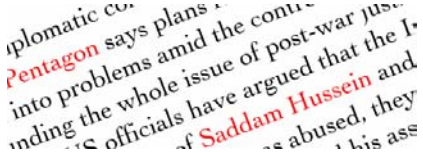


Deux approches a la prédiction des gènes

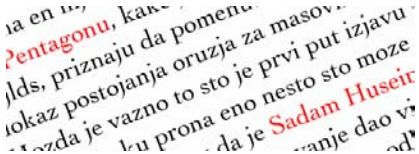
• **Statistique**: segments codants (exons) ont des séquences typiques aux extrémités et ils utilisent des sous-mots (de 2 ou 3 lettres) différents par rapport à des segments non-codants (introns).

• **Basé sur la similarité**: pleins de gènes humains sont similaires aux gènes de la souris, du poulet, ou même des bactéries. Donc, les gènes déjà connus de sourie, poulet et bactérie peuvent en principe aider à trouver les gènes de l'homme.

Approche basé sur la similarité: métaphore dans des langages différents



la comparaison des ''news'' avec les mêmes nouvelles dans un journal étranger peuvent révéler certaines similarités.



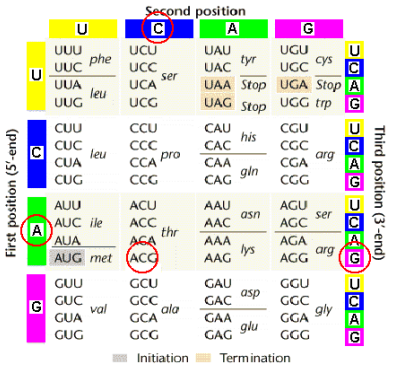
Code génétique et codons stop

Le code génétique est redondant. Par exemple, les acides-aminés Leucine, Alanine et Tryptophan sont codé par 6, 4 et 1 codons. Si on suppose que la distribution des codons sur la séquence est uniforme, alors ces acides-aminés doivent apparaître avec un ratio de 6:4:1, mais, en réalité, dans une protéine ils apparaissent typiquement avec un ratio différent: 6.9:6.5:1.

On en déduit que l'ADN n'est pas une séquence aléatoire.

Noter aussi que A et T apparaissent à la troisième position d'un codon avec une fréquence que peut rejoindre plus de 90% dans certaines espèces.

UAA, UAG et UGA correspondent a 3 codons stop que (avec le codon de départ ATG) déligne les Open Reading Frames (ORF)

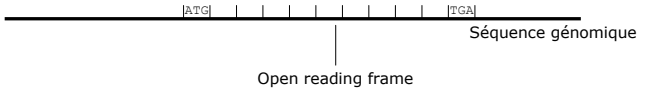


Six phases de lecture dans la sequence d'ADN

- codons stop– TAA, TAG, TGA
- codons de depart - ATG

Phases ouvertes de lecture - Open Reading Frames (ORFs)

- Détecte régions codantes potentielles en cherchant des **ORFs**
  - Un génome de longueur *n* est constitué de *n*/3 codons
  - Les codons stop coupent le génome en segments, défini par des codons stop consécutifs
  - les sous-segments de ces-ci qui commencent par des codons de départ (ATG) sont des ORFs
    - ORFs dans des phases différentes peuvent chevaucher



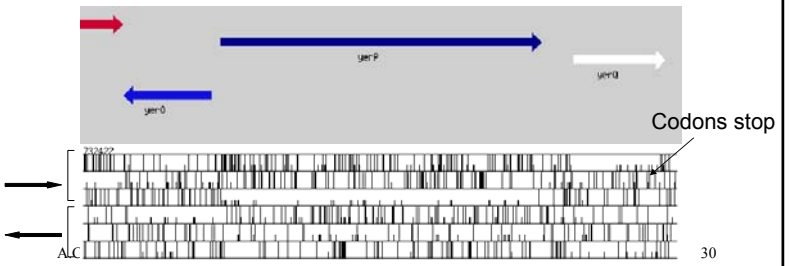
ORFs longs vs.ORFs courts

- Idée: un ORF long peut correspondre à un gène
  - aléatoirement, on doit attendre un codon stop chaque (64/3) ~= 21 codons
  - **D'autre part**, les gènes sont d'habitude plus longs que ça
- une approche de base consiste à lire le génome pour trouver les ORFs ayant une longueur qui excède un certain seuil
  - Cet approche est naïve parce qu'il y a des gènes (e.g. comme les gènes neuronaux ou les gènes du système immunitaire) qui sont relativement courts.

…dans les génomes procaryotes :

http://www.tigr.org

***Bacillus subtilis 168* Region View Display: Region Surrounding yerP**



Test des ORFs: usage des codons

- Créer une table d'hashage de 64 éléments et compter les fréquences des codons dans les ORFs
- Les acides aminés ont d'habitude plus qu'un codon associé, et certains codons sont plus utilisés que d'autres
- Un usage biaisé de codons peut permettre de caractériser un "vrai" gène.

⇧

Cette propriété est utilisée pour compenser au problème de la longueur des ORFs.

Usage des codons chez l'homme : mais attention aux calculs !

	U		C		A		G	
U	UUU Phe	57	UCU Ser	16	UAU Tyr	58	UGU Cys	45
	UUC Phe	43	UCC Ser	15	UAC Tyr	42	UGC Cys	55
	UUA Leu	13	UCA Ser	13	UAA Stp	62	UGA Stp	30
	UUG Leu	13	UCG Ser	15	UAG Stp	8	UGG Trp	100
C	CUU Leu	11	CCU Pro	17	CAU His	57	CGU Arg	37
	CUC Leu	10	CCC Pro	17	CAC His	43	CGC Arg	38
	CUA Leu	4	CCA Pro	20	CAA Gln	45	CGA Arg	7
	CUG Leu	49	CCG Pro	51	CAG Gln	66	CGG Arg	10
A	AUU Ile	50	ACU Thr	18	AAU Asn	46	AGU Ser	15
	AUC Ile	41	ACC Thr	42	AAC Asn	54	AGC Ser	26
	AUA Ile	9	ACA Thr	15	AAA Lys	75	AGA Arg	5
	AUG Met	100	ACG Thr	26	AAG Lys	25	AGG Arg	3
G	GUU Val	27	GCU Ala	17	GAU Asp	63	GGU Gly	34
	GUC Val	21	GCC Ala	27	GAC Asp	37	GGC Gly	39
	GUA Val	16	GCA Ala	22	GAA Glu	68	GGA Gly	12
	GUG Val	36	GCG Ala	34	GAG Glu	32	GGG Gly	15

Usage de codons chez la sourie : mais attention aux calculs !

AA	codon	/1000	frac
Ser	TCG	4.31	0.05
Ser	TCA	11.44	0.14
Ser	TCT	15.70	0.19
Ser	TCC	17.92	0.22
Ser	AGT	12.25	0.15
Ser	AGC	19.54	0.24
Pro	CCG	6.33	0.11
Pro	CCA	17.10	0.28
Pro	CCT	18.31	0.30
Pro	CCC	18.42	0.31

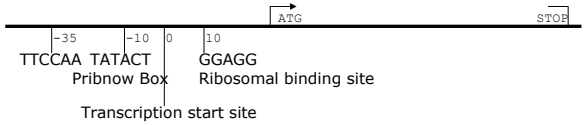
AA	codon	/1000	frac
Leu	CTG	39.95	0.40
Leu	CTA	7.89	0.08
Leu	CTT	12.97	0.13
Leu	CTC	20.04	0.20
Ala	GCG	6.72	0.10
Ala	GCA	15.80	0.23
Ala	GCT	20.12	0.29
Ala	GCC	26.51	0.38
Gln	CAG	34.18	0.75
Gln	CAA	11.51	0.25

Usage des codons et ratio de vraisemblance

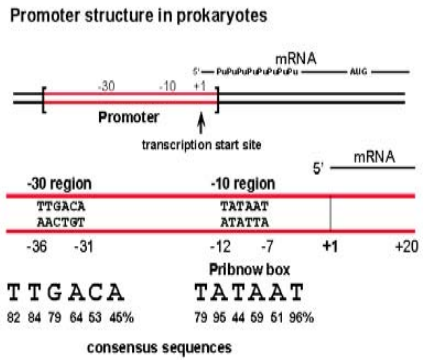
- Un ORF est plus "vraisemblable" qu'un autre s'il est composé par des codons plus "attendus".
- Faire des calcules avec une fenêtre glissante pour trouver les ORFs qui ont un usage de codons "probable".
- Ce test permet une précision de la prédiction des ORF bien plus fiable et précise que le test des longueurs.
- D'autre part, la longueur moyenne des exons des vertébrés est de 130 nucléotides, et cette longueur est souvent trop petite pour produire des piques fiables dans un ratio de vraisemblance.
- Améliorations : **comptage des hexamères** (fréquences de paires de codons consécutifs).

Prédiction de gènes et motifs

- Les régions en amont des gènes contiennent souvent des motifs qui peuvent être utilisés pour la prédiction des gènes.



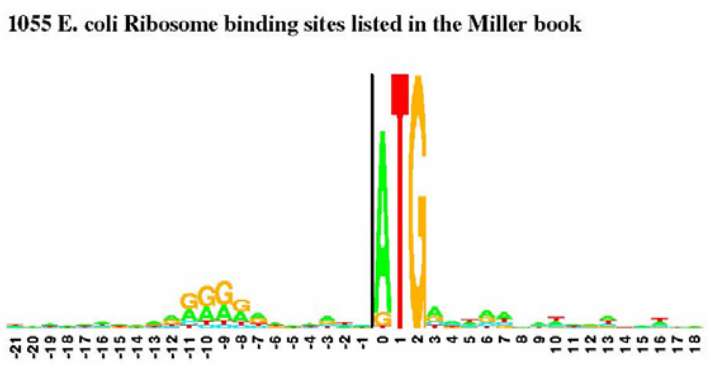
Structure des promoteurs dans les Procaryotes (*E.Coli*)



Transcription commence a 0.

- Pribnow Box (-10)
- Gilbert Box (-30)
- site de fixation du ribosome (+10)

Site de fixation du ribosome



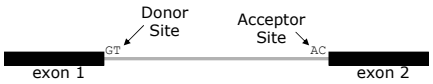
Attention: la numérotation a changée

Pour les eucaryotes : signaux d'épissage

- Problème: reconnaître les positions des signaux d'épissage et les jonctions exon-intron
  - Les sites d'épissage sont faiblement conserves
- Profiles faibles de reconnaissance des sites : les approches basées sur les Hidden Markov Model (HMM) ne fonctions pas de façon optimale parce qu'elles sont sensées à capturer les dépendances statistiques entre sites.

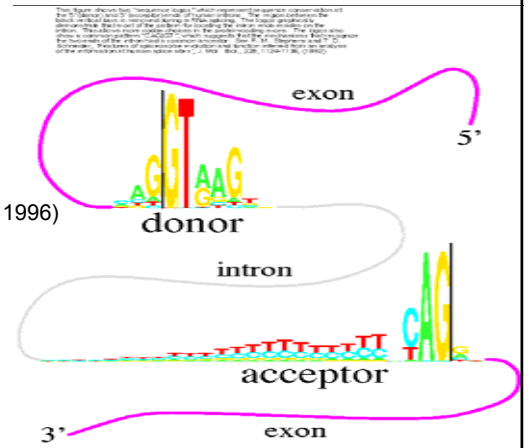
Site donateurs et site accepteur: les dinucléotides GT et AG

- Le début et la fin des exons sont signalés par deux sites distinctes, d'habitude ils'agit des dinucléotides GT et AC
- Détecter ces sites est difficiles, parce que GT et AC apparaissent très souvent.



Sites donateur et accepteur: logos des motifs

Donateur: 7.9 bits
Accepteur: 9.4 bits
(Stephens & Schneider, 1996)



(http://www-lmmb.ncifcrf.gov/~toms/sequencelogo.html)

TestCode

- Idee du test statistique décrit par James Fickett in 1982: tendance des nucléotides dans les régions codantes a être répétés avec une périodicité de 3
  - Détection de régions aléatoires par rapport à la fréquence des codons
  - Détection des régions codantes “putatives”, sans introns, exons, ou sites d’ épissage
- TestCode trouve des ORFs basés sur le biais en composition avec périodicité de 3.

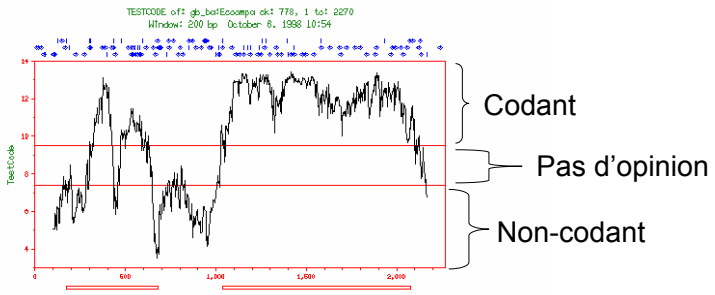
Statistiques de TestCode

- Définir une taille de fenêtre de pas moins de 200bp, glisser la fenêtre le long de la séquence avec des pas de 3 bases. Dans chaque fenêtre:
  - Calculer pour chaque base {A, T, G, C}
    - max (n_{3k+1}, n_{3k+2}, n_{3k}) / min (n_{3k+1}, n_{3k+2}, n_{3k})
  - Utiliser ces valeurs pour obtenir une probabilité à partir d’une table de référence (qui a été précédemment construite à partir de données connues)

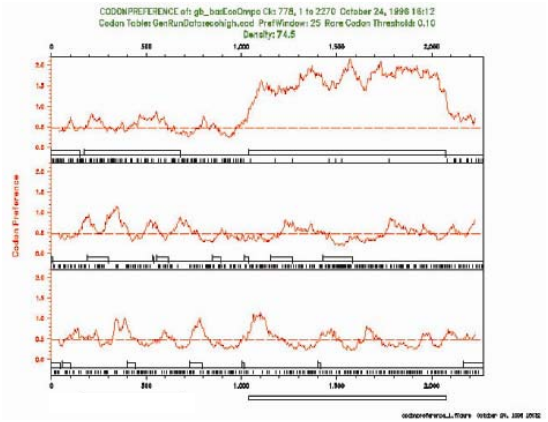
Statistiques de TestCode
(continuation)

- Les probabilités seront alors utilisées comme des indicateurs de régions " codantes” ou “non-codantes”, ou bien pour détecter des régions sur lesquelles on exprime “pas d’opinion” quand il n’est pas claire ce que c’est le contenu aléatoire d’une séquence donnée.
- La suite résultante des probabilités peut être ainsi représentée dans un graphe.

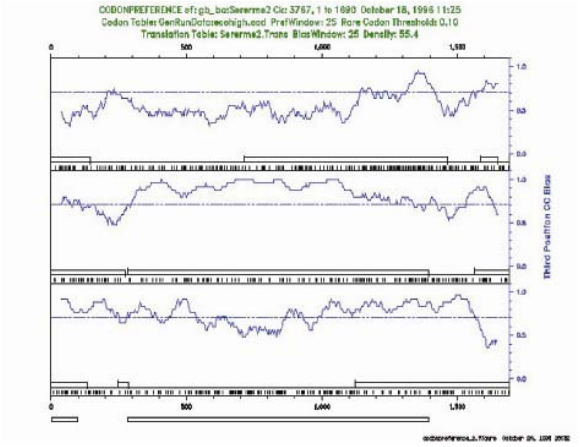
Exemple de résultat de TestCode



D'autres algorithmes utilisent des statistiques sur les codons: CODONPREFERENCE



Plot de vraisemblance logarithmique log(P/1-P) pour les trois reading frames. Chaque point représente le score obtenu avec une fenêtre de 25 codons définie autour du point.



CODONPREFERENCE utilise ici seulement l'information provenant de la troisième position du codon.

En générale, les algorithmes qui utilisent des statistiques sur les codons:

1. dépendent de la fiabilité des statistiques sur la fréquence des codons faites au préalable à partir des séquences connues
2. ont des difficultés à détecter les gènes transfères horizontalement et d'autres cause d'hétérogénéité.

Algorithmes de prédiction des gènes

- **GENSCAN**: utilise les Chaînes de Markov Cachées (Hidden Markov Models - HMMs)
- **TWINSCAN:** utilise les HMM et la similarité (e.g., entre les génomes de l'homme et de la sourie)

Prédiction des gènes: approches basées sur la similarité

Plan

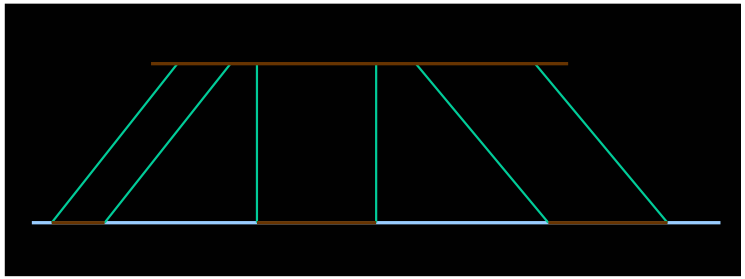
- L'idée d'une approche basée sur la similarité dans la prédiction des gènes
- Exon Chaining Problem
- Spliced Alignment Problem
- Outils de prédiction des gènes

Utilisation de gènes connus pour la prédiction de nouveaux gènes

- Certaines génomes sont très bien étudiés et leurs gènes ont été vérifiés expérimentalement.
- **Idée**: des organismes proches peuvent avoir des gènes similaires
- Gènes inconnus dans une espèce peuvent être comparés à des gènes dans les espèces proches.

Problème: étant donné un gène connu et un génome pas annoté, trouver un ensemble de sous-chaînes de la séquence génomique dont la concaténation correspond au mieux au gène.

Comparaison de gènes dans deux génomes



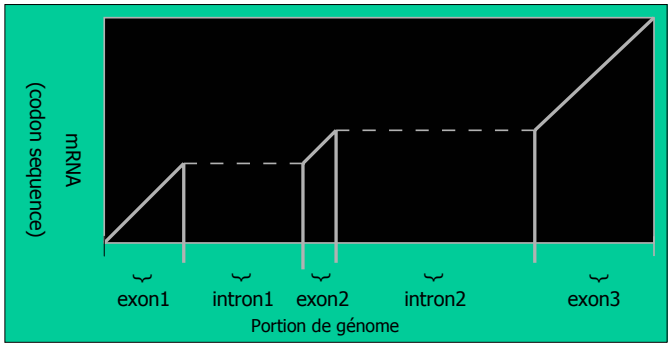
Petits îlots de similarité correspondants aux similarités entre exons.

Et s'il on utilisait les séquences protéiques ?: traduction renversée

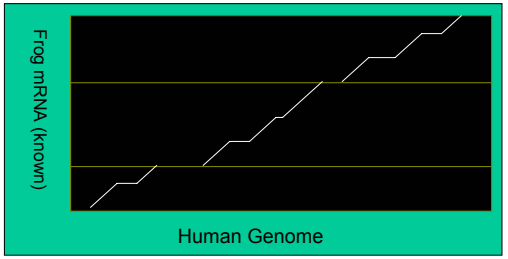
- Etant donnée une protéine connue, trouver un gène dans le génome qui codifie cette protéine
- Une possibilité est d'inférer la séquence d'ADN en renversant le processus de traduction:
  - correspondance non exacte: le même acide aminé correspond à >1 codon
  - Ce problème est essentiellement réductible à un problème d'alignement

- Ce problème de traduction renversée peut être modélisé avec un algorithme de complexité $n^{3,}$ où on réalise l'alignement en n^2 et on arrive a n^3 par la modélisation de l'insertion des introns.
- Le problème de cette approche: les nucléotides sont appariés des que possible et les introns sont introduits à chaque opportunité.

Comparaison des génomes contre les mRNA



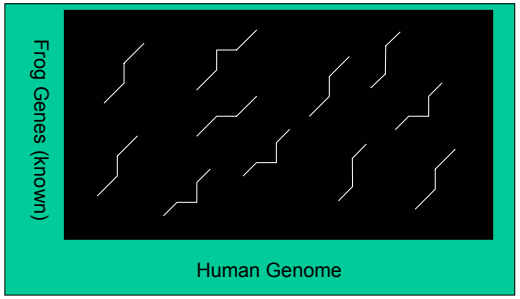
- un mRNA de la grenouille est aligné à des différentes positions du génome humain
- Trouver le "meilleur" chemin pour identifier la structure des exons du gène de l'homme.



Un mRNA est un seul epissage possible !

Trouver les alignement locaux

Recherche de tous les îlots de similarité



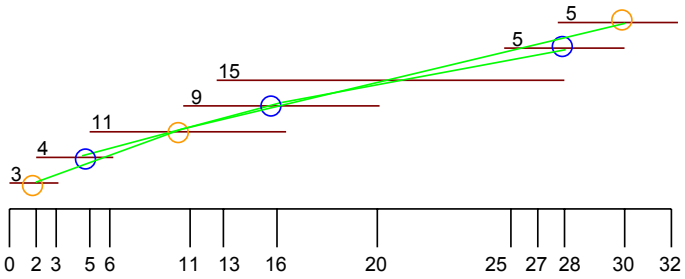
Enchaînement des alignement locaux

- Trouver les sous-chaînes qui s'apparient bien avec un gène donné (exons candidats)
- Définir un exon candidat comme un triplet

  (*l, r, w*)

  (*left, right, weight* défini comme un score d'alignement locale)
- Chercher la chaîne maximale de sous-chaînes locales.
  - Chaîne: un ensemble d'intervalles non-chevauchants et non-adjacents.

Exon Chaining Problem



- Localiser le début et la fin de chaque intervalle (*2n* points)
- Trouver le "meilleur" chemin

Exon Chaining Problem: Formulation

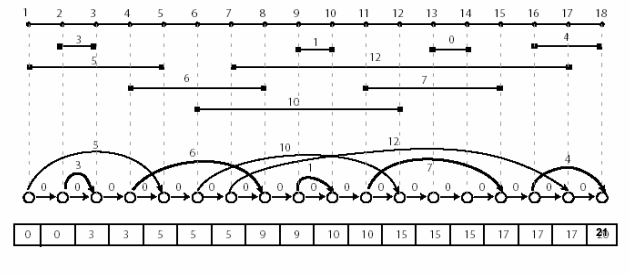
- **Exon Chaining Problem:** étant donné un ensemble d'exons putatifs, trouver un ensemble maximale d'exons putatifs non-chevauchants.
- **Input**: un ensemble d'intervalles pesés (exons putatifs)
- **Output**: une chaîne maximale d'intervalles de cet ensemble

Exon Chaining Problem: Formulation

- **Exon Chaining Problem:** étant donné un ensemble d'exons putatifs, trouver un ensemble maximale d'exons putatifs non-chevauchants.
- **Input**: un ensemble d'intervalles pesés (exons putatifs)
- **Output**: une chaîne maximale d'intervalles de cet ensemble

Est-ce que un algorithme gourmand résoudrait ce problème ?

Exon Chaining Problem: le graphe de représentation



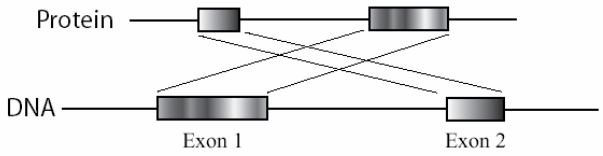
Ce problème peut être résolu avec un algorithme de programmation dynamique en temps O(*n*).

Algorithme d'enchaînement des Exons

ExonChaining (*G, n*) //Graph, number of intervals
1 **for** $i \leftarrow$ to $2n$
2 $s_i \leftarrow 0$
3 **for** $i \leftarrow 1$ to $2n$
4 **if** vertex v_i in G corresponds to right end of the interval I
5 $j \leftarrow$ index of vertex for left end of the interval I
6 $w \leftarrow$ weight of the interval I
7 $s_j \leftarrow max\{s_j + w, s_{i-1}\}$
8 **else**
9 $s_i \leftarrow s_{i-1}$
10 **return** s_{2n}

Exon Chaining: faiblesses

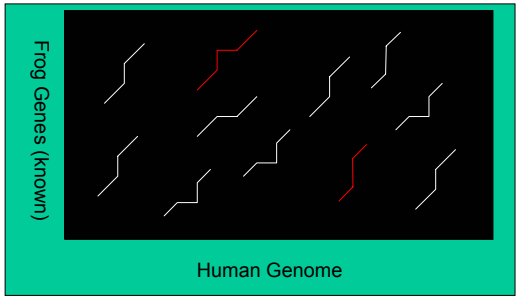
- La chaîne optimale d'intervalles peut correspondre a aucun des alignement valides
  - Le premier intervalle peut correspondre a un suffixe, et le deuxième intervalle peut correspondre a un préfixe
  - La combinaison de tels intervalles n'est pas un alignement valide



- Définition faible/problématique des extrémités des exons putatifs

Chaînes impossibles

Les similarités locales en rouges forment 2 intervalles non - chevauchants mais ils ne forment pas un alignement globale valide.



Spliced Alignment

Mikhail Gelfand et collègues (Gelfand, Mironov, Pevzner, 1996) ont proposé l'approche **spliced alignment** qui utilise une protéine d'un génome pour reconstruire la structure exon-intron d'un gène (codant pour une protéine suffisamment proche) dans un autre génome:

– On commence par sélectionner soit tous les exons putatifs entre des sites accepteurs et des sites donneurs potentiels soit par trouver toutes sous-chaînes similaires a une protéines cible (comme dans le Exon Chaining Problem).

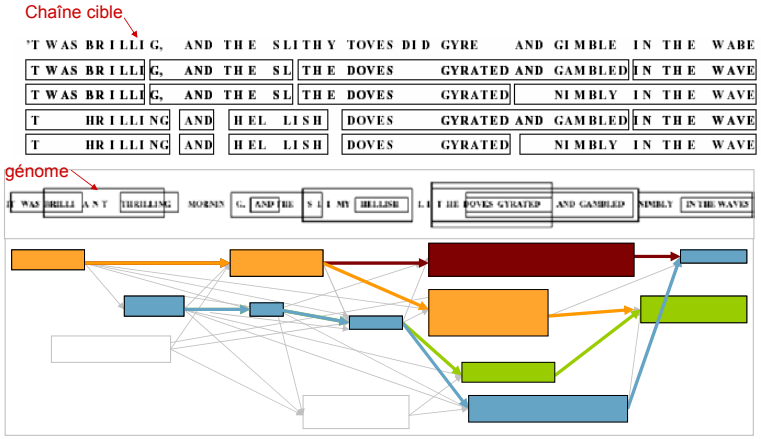
– Cet ensemble est en suite filtré de telle façon que tout exon vrai soit retenu et éventuellement des exons faux soient retenus aussi. Ca sera l'alignement avec la protéine que nous permettra de discriminer les vrai positifs.

Spliced Alignment Problem: formulation

- **Problème**: trouver une chaîne de bloques dans une séquence génomique que s'apparie le mieux à une séquence cible
- **Entrée**: une séquence génomique *G*, une séquence cible *T*, et un ensemble d'exons candidats *B*.
- **Sortie**: une chaîne d'exons Γ tel que le score d'alignement globale entre Γ* et T est maximum entre toutes les chaînes de bloques générées à partir de *B*.

Γ* - concaténation de tous les exons à partir de la chaîne Γ

Exemple de Lewis Carroll



Spliced Alignment: Idée

- Calculer le meilleur alignement entre le *i*-prefixe de la séquence génomique *G* et le *j*-prefixe du cible *T:*
- *S(i,j)*
- Mais qu'est-ce que le *"i*-prefixe*" de G ?* On peut avoir plusieurs *i*-prefixes *de G* dépendamment du bloque *B* que l'on considère.

- Calculer le meilleur alignement entre le *i*-prefixe de la séquence génomique *G* et le *j*-prefixe du cible *T* ***sous l'hypothèse*** que l'alignement utilise le bloque *B à* la position *i*

  S(i,j,B)

Récurrence de Spliced Alignment

si *i* n'est pas le noeud initiale du bloque *B*:

- $S(i, j, B) =$

  $\max \{ \; S(i-1, j, B) - \text{indel penalty}$

  $S(i, j-1, B) - \text{indel penalty}$

  $S(i-1, j-1, B) + \delta(g_i, t_j) \}$

si *i* est le noeud initiale du bloque *B*:

- $S(i, j, B) =$

  $\max \{ S(i, j-1, B) - \text{indel penalty}$

  $\max_{\text{tout bloque } B' \text{ qui précède } B} S(end(B'), j, B') - \text{indel penalty}$

  $\max_{\text{tout bloque } B' \text{ qui précède } B} S(end(B'), j-1, B') + \delta(g_i, t_j)$

  $\}$

Solution Spliced Alignment

- Apres avoir calculé le tableau tridimensionnel *S(i, j, B)*, le score du spliced alignment optimale est:

$$\max_{\text{tous bloques } B} S(end(B), length(T), B)$$

Spliced Alignment: Complications

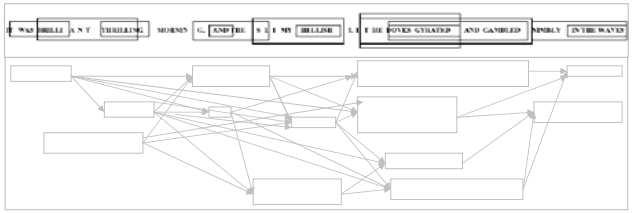
- Considérer plusieurs *i*-prefixes signifie ralentir le temps de calcul :

  $O(mn^2 |B|)$

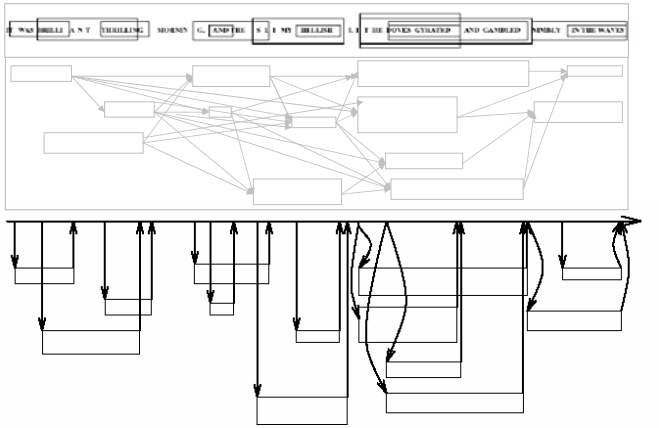
  ou *m* est la longueur du cible, *n* est la longueur de la séquence génomique et $|B|$ est le nombre de bloques.

- Un choix naif dans la construction des bloques peut induire a un ***effet mosaïque***: exons courts sont facilement combinables sur une protéines cible quelconque. Cela amène à des prédictions incorrectes.

Spliced Alignment: Speedup

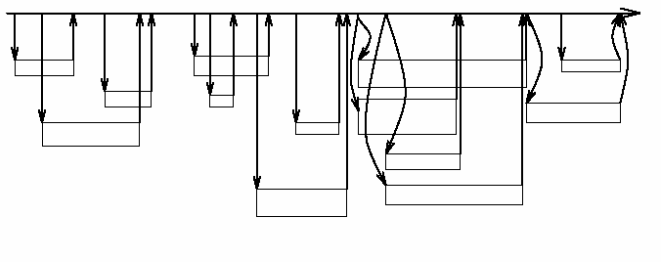


Spliced Alignment: Speedup



Spliced Alignment: Speedup

$P(i,j)=\max_{\text{all blocks } B \text{ preceding position } i} S(end(B), j, B)$



Exon Chaining vs Spliced Alignment

- Dans l'algorithme Spliced Alignment, chaque chemin examine les chaînes obtenues par concaténation d'étiquettes de ses arcs. Le poids du chemin est défini comme le score d'alignement optimale entre étiquettes concaténées (bloques) et séquence cible
- Il défini le poids du chemin entier dans le graphe, mais pas les poids pour les arcs individuels.
- Exon Chaining fait l'hypothèse que les positions et les poids des exons sont pre-définis.

Outils statistiques de prédiction des gènes

- GENSCAN/Genome Scan
- TwinScan
- Glimmer
- GenMark

L'algorithme GENSCAN

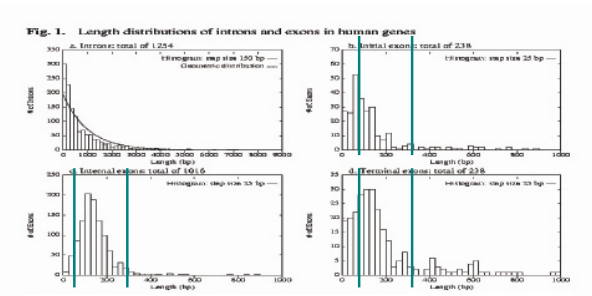
- L'algorithme est basé sur un modèle probabiliste de la structure des gènes similaire a un modèle de chaîne de Markov (*Hidden Markov Models - HMMs).*
-
- GENSCAN utilise un ensemble d'apprentissage pour estimer les paramètres *HMM*, et en suite il prédit la structure des exons en utilisant l'approche de vraisemblance maximale qui est devenu standard a plusieurs algorithmes bases sur les HMM (algorithme de *Viterbi*).
  - Entrée biologique : biais des codons dans les régions codantes, la structure des gènes (start et stop codons, longueur typique d'un exon et d'un intron, la présence d'un promoteur, la présence de gènes sur les deux brin, etc)
  - Il couvre les cas ou la séquence d'entrée ne contient pas de gènes, contient des gènes partiels, des gènes complets, des gènes multiples.

Exemple de « gène moyen » dans les vertébrés :

gène de 30kb
partie codante de 1-2kb
6 exons de 150bp chacun

Déviation du comportement moyen dans l'homme:

Dystrophyn — gène de 2.4MB en longueur
Facteur VIII de coagulation du sang — 26 exons de taille entre 69bp et 3106bp



Intron, exon length data from 238 multi-exon genes of GENSCAN learning set

Modèles de Markov pour la détection des régions codantes

On considère une fenêtre de 6 nucléotides dans la séquence d'ADN et on construit un modèle de Markov d'ordre 5.

Deux tables de probabilités (de taille 4^6) sont préparées en avance pour les régions codantes et les régions non-codantes: pour chaque 6-tuplet de bases la table enregistrera la probabilité d'observer la 6ème base étant données les 5 bases précédentes.

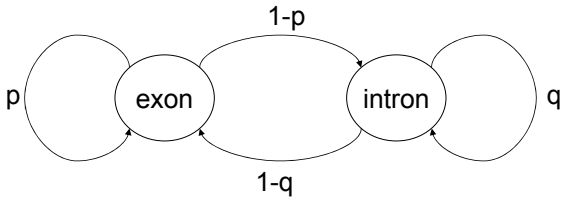
Etant donnée une séquence, on pourra alors estimer la probabilité qu'elle soit codante à partir des deux tables.

Ce modèle ne considère aucune information provenant des ORFs et pour cette raison est appelé modèle homogène.

Un modèle non-homogene est un modèle ayant des tables différentes pour les trois phases de lecture différentes. Le problème avec ces modèles quand on travail avec les génomes eucaryotes est que parfois les exons sont trop petits et que c'est difficile de détecter les jonctions d'épissage (splice junctions: donor and acceptor sites)

Distribution des longueurs

Un HMM simple pour identifier un gène eucaryote :



P(exon de longueur k) = $p^k(1-p)$ (distribution géométrique)

Mais la distribution des longueurs des exons ne peut pas être géométrique: la longueur semble jouer un rôle fonctionnel dans l'epissage. Exons qui sont trop courts (<50pb) ne sont pas détectées par le spliceosome et exons qui sont trop longues (>300pb) sont difficiles à détecter. D'autres modèles de la longueurs des exons ont été proposés.

HMM généralisés (GHMM)

Dans un GHMM la sortie d'un état peut être une chaîne de symboles de longueur finie. Pour un état donné, la longueur et la chaîne de symboles peut être choisi aléatoirement selon une distribution de probabilité. La distribution de probabilité peut être différente pour des états différents.

- Q ensemble fini d'états
- π distribution de probabilité associée à l'état initiale
- T_{ij} probabilité de transition pour chaque paires d'états i,j $\in$Q
- f distribution des longueurs par état (f_q est la distribution des longueurs de l'état q)
- Modèles de probabilités pour chaque état selon les chaînes de sorties issues après la visite d'un état

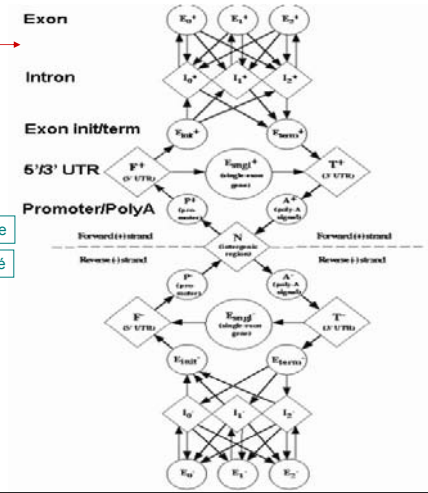
Modèle GENSCAN

Le modèle probabiliste de la structure des gènes (Berge et Karlin, 1997) est basé sur un GHMM

Division des états selon les 3 reading frames

Modélisation du gène dans le brin de lecture

Modélisation du gène dans le brin opposé



États = unités fonctionnelles d'un gène
Transitions entre états = le modèle assure que l'ordre de visite des états est biologiquement cohérent

Prédiction de la structure d'un gène

Une lecture Φ d'une séquence S de longueur L est une séquence ordonnée d'états (q_1,…, q_t) de durée d_i associée à chaque etat ($L = \sum_{i=1}^{t} d_i$).

Φ est une annotation possible d'une séquence.

Supposons de connaître une lecture Φ et une séquence S. Soit S_i le segment de S produit par q_i et soit $P(S_i|d_i)$ la probabilité de générer S_i à l' état q_i et avec durée d_i. La probabilité que le modèle est passé par l'état q_i pour générer la séquence selon Φ est:

$$P(\Phi, S) = \pi_{q1} f_{q1}(d_1) P(S_1|d_1) \prod_{k=2}^{t} T_{qk-1qk} f_{qk}(d_k) P(S_k|d_k)$$

Supposons de connaître S ainsi que une lecture Φ, les deux de longueur L. La probabilité conditionnée de Φ étant donné que la séquence génératrice est S, peut être calculée comme suit:

$$P(\Phi|S) = \frac{P(\Phi,S)}{P(S)} = \frac{P(\Phi,S)}{\sum_{\Phi i \text{ est une lecture de longueur L}} P(\Phi_i,S)}$$

Considérations sur :

- la détermination des paramètres dans GENSCAN
- la performance de GENSCAN (sur l'ensemble de 570 gènes de Burset/Guigo)

se trouvent dans les articles cités en bibliographie.

GENSCAN : Limites

- Il n'utilise pas la similarité pour prédire les gènes.
- Il ne s'occupe pas d'epissage alternatif.
- Il peut combiner deux exons appartenant a deux gènes consécutifs.

GenomeScan

- Inclus l'information sur la similarité dans GENSCAN: prédit la structure des gènes qui correspond a une probabilité maximale conditionnée a l'information sur la similarité.
- L'algorithme est une combinaison de deux sources d'information
  - Modèle probabiliste de exons-introns
  - Information sur la similarité des séquences

TwinScan

- Il aligne deux sequences et marque chaque base comme gap (-), mismatch (:), match (|), en produisant un nouveau alphabet de 12 lettres: Σ= {A-, A:, A |, C-, C:, C |, G-, G:, G |, T-, T:, T|}.
- Il execute un algorithme de Viterbi en utilisant les emissions $e_k(b)$ ou $b \in$ {A-, A:, A|, …, T|}.
- Les probabilités d'émission sont estimée a partir de paires de gènes homme/sourie.
  - Ex. $e_I(x|) < e_E(x|)$ car matches sont favorises dans les exons, et $e_I(x-) > e_E(x-)$ car gaps (ainsi que mismatches) sont favorisés dans les introns.
  - Compensation pour l'occurrence dominante d'une région poly-A dans les introns

Glimmer

- **G**ene **L**ocator and **I**nterpolated **M**arkov **M**odel**ER**
- Trouve les gènes dans les génomes bactériens
- Utilise des modèles de Markov interpolés
- Il est constitué de 2 programmes
  - **BuildIMM**
    - Il prends les séquences en entrée et sort des Modèles de Markov Interpolés (IMMs)
  - **Glimmer**
    - Il prends IMMs et sort tous les gènes candidats
    - Automatiquement il résolu les gènes chevauchants en choisissant un gène (limite)
    - Il marque les gènes “suspected to truly overlap” pour suggérer a l’utilisateur de regarder explicitement

GenMark

- Basé sur des modèles de chaînes de Markov *non-stationnaires*
- Les résultats sont donnes graphiquement avec les probabilités d’une position codante vs une position non-codante dépendamment de la position du nucléotide dans la séquence.

EXOGEAN – EXpert On GEne ANnotation

Les annotations de référence sont des annotations manuelles, i.e. générées semi-automatiquement par des experts humains

Les experts humains combinent des objets biologiques en utilisant des règles euristiques

Les objets et les règles qu’ils utilisent peuvent changer

Les transparents concernants EXOGEAN sont basés sur les deux présentations cités dans les références bibliographiques.

Un cadre générique pour l’annotation experte

But ultime: pouvoir combiner de façon très flexible tout type d’objet biologique en utilisant tout type de règles heuristique

Exogean utilise des **multi-graphes orientés acycliques colorés** (MOAC) dans lesquels:

- Les nœuds représentent des objets biologiques
- Les arcs multiples entre noeuds représentent des relations déduites de l’expertise humaine

En terme de MOAC, le protocole d'annotation de l'expert humain peut être vue comme une succession de **constructions et quotientements de MOACs**.

A partir d'objets simples tels que des alignements de séquences, les quotientements successifs génèrent des objets de plus en plus complexes jusqu'aux gènes.

Avantage: avoir un formalisme unique pour toutes les heuristiques

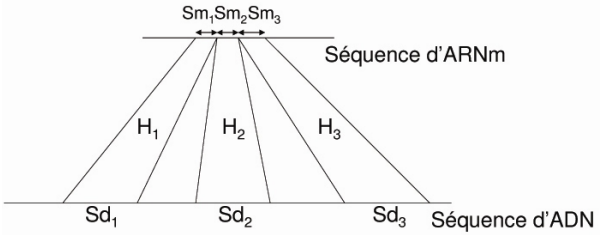
Objets de base: les HSPs

HSP (High-scoring Segment Pair)

HSP = résultat d'un alignement heuristique local (cf Blast) entre la séquence d'ADN à annoter et d'autres séquences biologiques (protéines, ARNm)

Modèle de transcrit

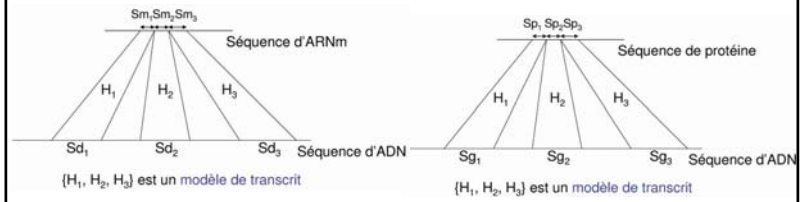
Modèle de transcrit = ensemble d'HSPs désignant la même structure de transcrit



{H_1, H_2, H_3} est un modèle de transcrit

Modèle de transcrit

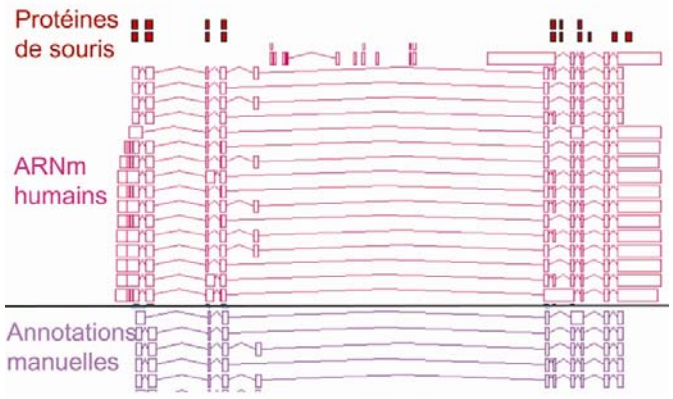
Modèle de transcrit = ensemble d'HSPs désignant la même structure de transcrit



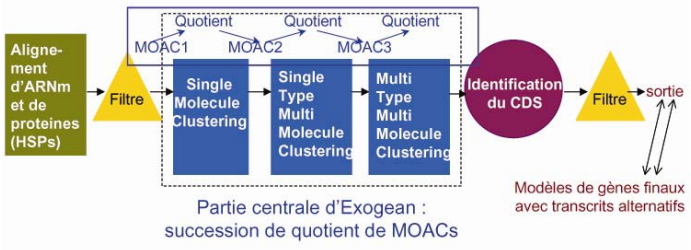
A partir de ces HSPs le but est de reconstruire le gène.
Le but est compliqué car:

- Similarité pas à 100%
- Redondance
- Pas de bornes précises
- Conflits, imperfections, artefacts
- Epissage alternatif

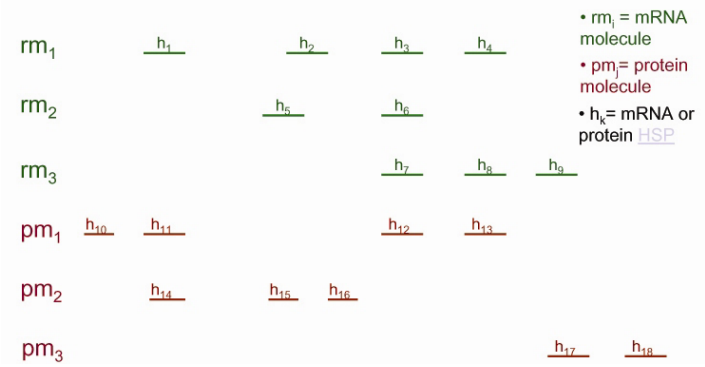
Un annotateur face aux HSPs



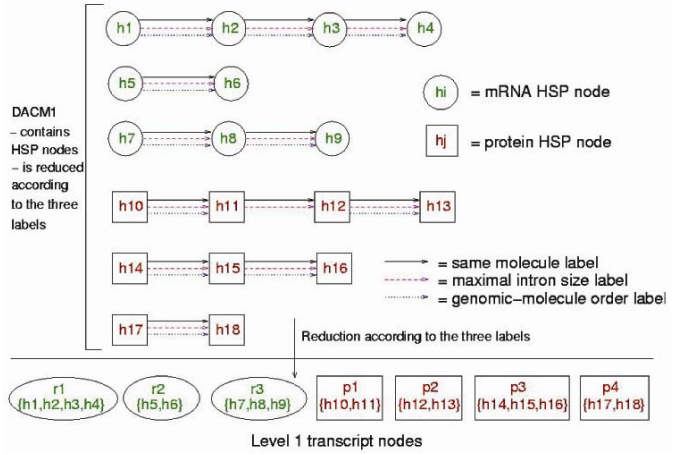
Le programme Exogean (EXpert On Gene Annotation)



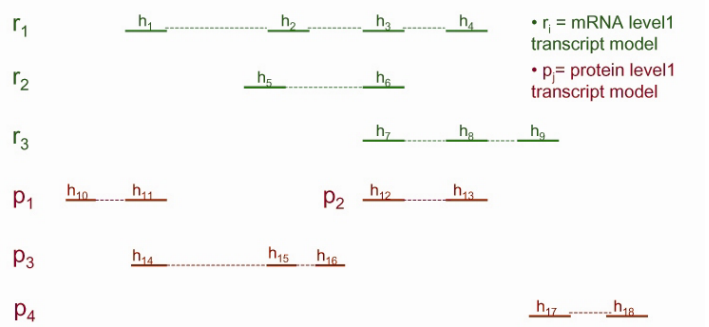
Exemple : zoom sur une région où des molécules d'ARNms et protéines se sont alignés



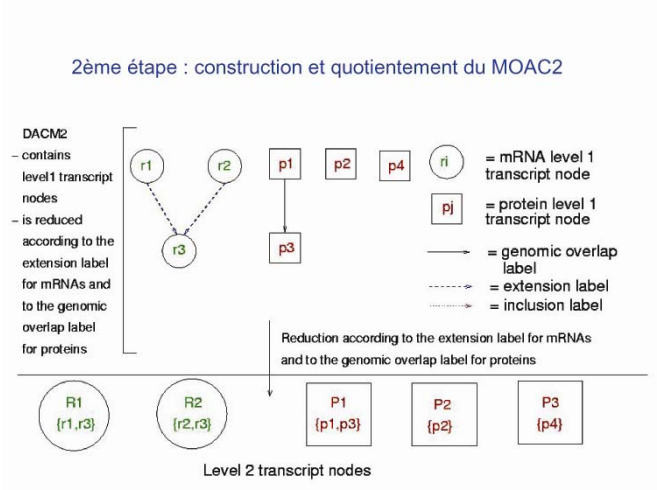
1ère étape : construction et quotientement du MOAC1



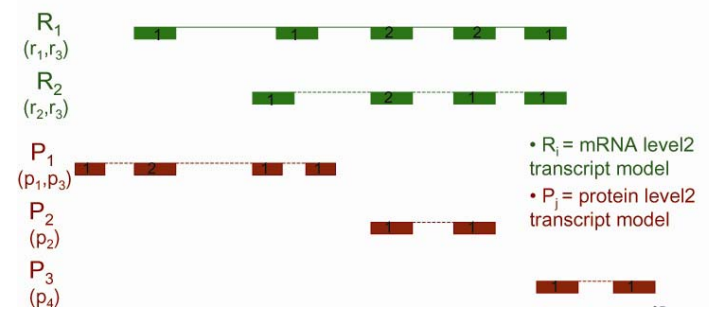
Quotientement du MOAC 1 : génère des modèles de transcrits de niveau 1



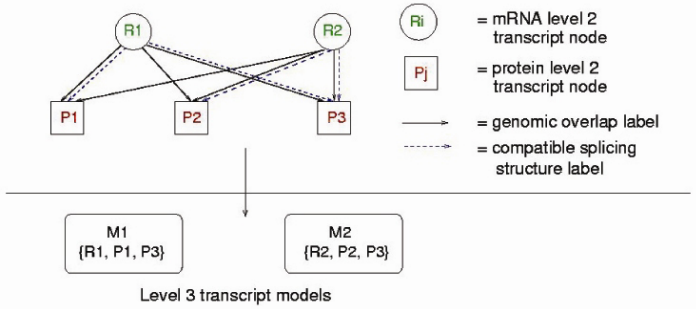
2ème étape : construction et quotientement du MOAC2



Quotientement du MOAC 2 : génère des modèles de transcrit de niveau 2



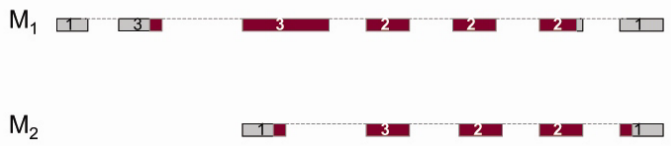
3ème étape : construction et quotientement du MOAC3



Level 3 transcript models

Quotientement du MOAC 3 : génère les modèles de transcrits finaux

M_i = modèle de transcrit final dans lequel un CDS est recherché (en rouge)



Validation: workshop EGASP'05

Trouver tous les éléments fonctionnels présents dans 44 séquences d'ADN génomique représentant 1% du génome humain

Programmes participants

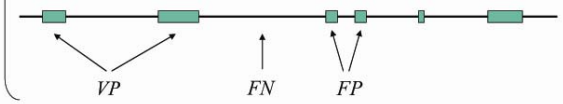
Catégorie	Type de méthode	Participants
1	Utilise tout type d'information disponible	Paragon+n-scan, Fgenesh++
2	*Ab initio* utilisant un seul génome	Augustus, Genemark, Genzilla
3	Utilise des ESTs, ARNm et protéines	Exonhunter, **Exogean**, Augustus Aceview, Paragon+n-scan, Ensembl
4	Utilise des génomes multiples	Augustus, Saga, N-scan, Cstminer, Dogfish-C, Twinscan-mars
5	Prédit des gènes non usuels	Spida
6	Pseudogènes, gènes non canoniques …	Uncover, Sbpseudo, Geneid+Sgp
7	Prédit des exons uniquement	Augustus, Dogfish-C-E
8	Prédit des promoteurs	MCpromoter, Fprom

→ 20 programmes d'annotation de gènes en compétition

Quelles mesures pour l'évaluation?

Annotation de référence (Havana) :

Prédiction (programme X) :



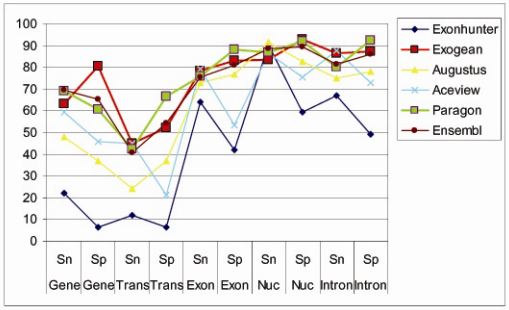
$$Sensibilité = Sn = \frac{VP}{VP + FN} \qquad Spécificité = Sp = \frac{VP}{VP + FP}$$

Sn, Sp calculés à différents niveaux (nucléotide, exon, transcrit, gène) ⇒ 8 mesures au total

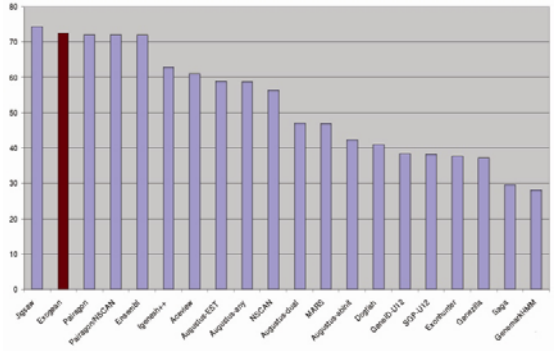
Résultat de l'évaluation

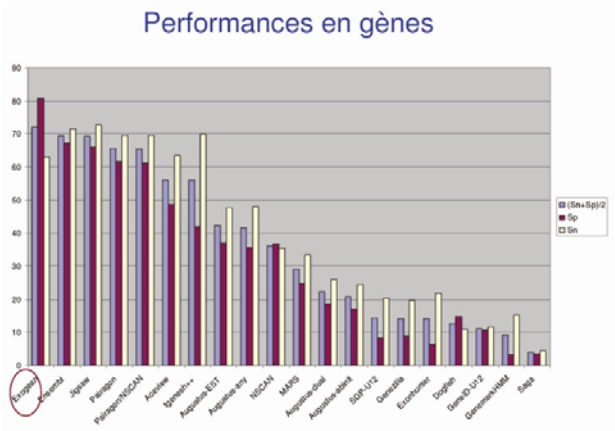
	Sn Nucleotide	Sp Nucleotide	Sn Exon	Sp Exon	Sn Transcrit	Sp Transcrit	Sn Gène	Sp Gène
Augustus-any	94.42	82.43	74.67	76.76	22.65	35.59	47.97	35.59
Pairagon/NSCAN	87.77	92.78	76.85	88.91	39.29	60.34	69.59	61.32
Jigsaw	**94.56**	92.19	80.61	**89.33**	34.05	**65.95**	**72.64**	65.95
Igenesh++	91.09	76.89	75.18	69.31	36.21	41.61	69.93	42.09
Augustus-abinit	78.65	75.29	52.39	62.93	11.09	17.22	24.32	17.22
GenemarkHMM	78.43	37.97	50.58	29.01	6.93	3.24	15.20	3.24
Genezilla	87.56	50.93	62.08	50.25	9.09	8.84	19.59	8.84
Exonhunter	90.46	59.67	64.44	41.77	10.48	6.33	21.96	6.33
Exogean	84.18	94.33	79.34	83.45	42.53	52.44	63.18	**80.82**
Augustus-EST	92.62	83.45	74.10	77.40	22.50	37.01	47.64	37.01
Aceview	90.94	79.14	**85.75**	56.98	**44.68**	19.31	63.51	48.65
Pairagon	87.56	92.77	76.63	88.95	39.29	60.64	69.59	61.71
Ensembl	90.18	92.02	77.53	82.65	39.75	54.64	71.62	67.32
Augustus-dual	88.86	80.15	63.06	69.14	12.33	18.64	26.01	18.64
NSCAN	85.38	89.02	67.66	82.05	16.95	36.71	35.47	36.71
Saga	52.54	81.39	38.82	50.73	2.16	3.44	4.39	3.44
CSTminer	66.54	27.64	0.00	0.00	0.00	0.00	0.00	0.00
Dogfish	64.81	88.24	53.11	77.34	5.08	14.61	10.81	14.61
MARS	84.25	74.13	65.56	61.65	15.87	15.11	33.45	24.94
GeneID-U12	75.03	78.83	51.41	63.92	5.39	10.69	11.49	10.69
SGP-U12	82.84	66.80	59.73	49.20	9.71	8.44	20.27	8.44
Spida	35.99	94.25	29.81	35.09	0.00	0.00	0.00	0.00
Dogfish-exon	8.065	**95.77**	1.66	27.22	0.00	0.00	0.00	0.00
Augustus-exon	94.42	82.43	39.80	40.89	0.00	0.00	0.00	0.00

Résultats des programmes de la meilleure catégorie



Performances globales





Références bibliographiques

C.B. Burge and S. Karlin. Finding the genes in genomic DNA. *J. Mol. Bio*, 268:78–94, 1997.

M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 1996.

M.S. Gelfand, A.A. Mironov, and P.A. Pevzner. Gene recognition via spliced sequence alignment. In *Proc. Natl Acad Sci USA*, volume 93, pages 9061–9066, 1996.

A. Krogh R. Durbin, S. Eddy and G. Mitchison. *Biological sequence analysis*. Cambridge university press, Cambridge, 1998.

S.Djebali, F.Delaplace, H.Roest Crollius, EXOGEAN: an Expert in Eukariotic Gene Annotation, Manuscript, 2005.

S.Djebali, F.Delaplace, H.Roest Crollius, Multi-graphes orientés acycliques colorés pour une annotation de gènes experte, présentation à l'atelier de réflexion Génopole, 2005.

S.Djebali, F.Delaplace, H.Roest Crollius, Un cadre formel pour l'annotation experte des genes eucaryotes, présentation à la journée des doctorants, IBISC, 2006.