# Support de cours
# Annotation des génomes
# (Partie II)

## Recherche des régions codant pour des protéines chez les procaryotes

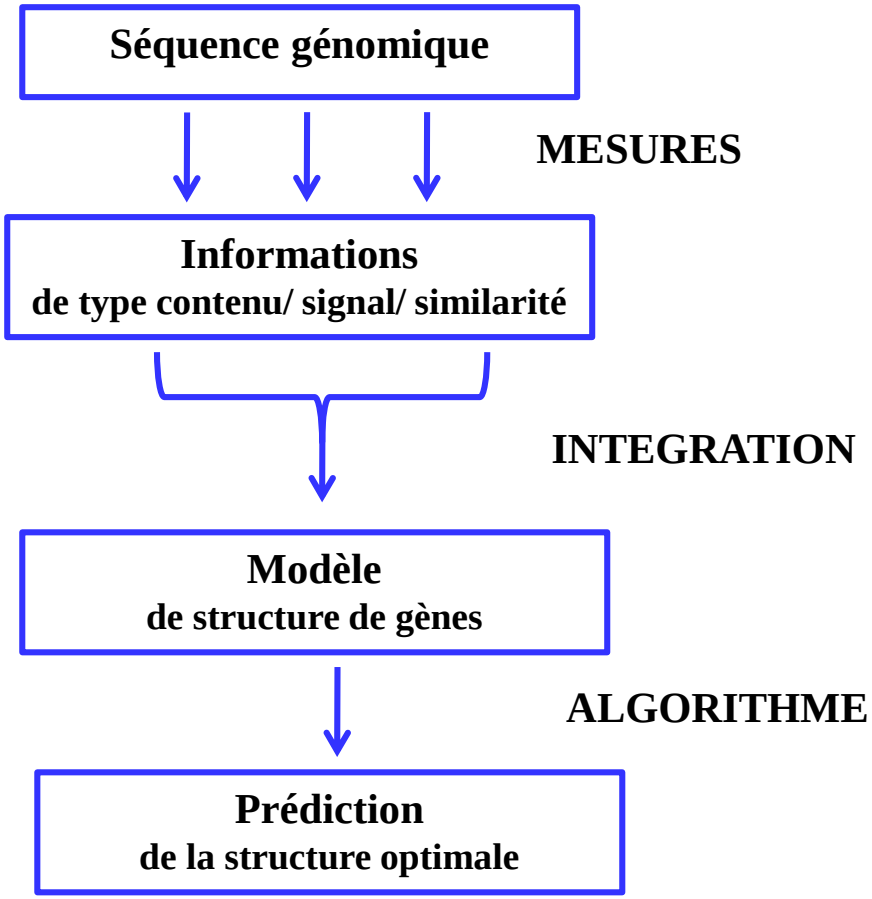
- recherche des ORFs (Open reading frame)
- recherche des unités de traduction. Même si les gènes sont co-transcrits, ils sont en général traduits de façon indépendante (recherche des Shine Dalgarno en 5' du codon initiateur). Permet d'identifier le « bon » codon initiateur.
- recherche des unités de transcription. Chez les procaryotes, certains gènes sont co-transcrits donc recherche de la structure en opérons (promoteurs et terminateurs de transcription)

## Recherche des régions codant pour des protéines chez les eucaryotes

- recherche de la structure en exon/intron du gène
- recherche des 5'UTR et 3' UTR
- recherche des promoteurs et des sites de polyadénylation

# Recherche des régions codant pour des protéines

## Fonctionnement schématique d'un logiciel de prédiction de gènes



# Une méthode simple: ORFfinder (NCBI)

**Recherche les phases ouvertes de lecture, les ORFs, dans les 6 cadres de lecture (les 3 cadres du brin direct et les 3 cadres du brin complémentaire).**

**Attention problème de sémantique :**

**Alors qu'une ORF est normalement définie entre deux codons stop**

**stop XXXXXXXXXXXXXXXXX stop**

**n codons**

**Dans ORF Finder, elle est définie entre un codon start et un codon stop**

**ATG XXXXXXXXXXXXXXXXX stop**

**n codons**

**On considère en général que les ORFs supérieures à 100 codons (300 pb) comme étant potentiellement codantes (analyse statistique a montré que bien que des gènes de taille inférieure à 100 codons existent, la majorité des petites ORFs étaient des faux positifs, donc lors de l'annotation d'un génome, dans un premier temps on ne retient que les ORFs de taille supérieure ou égale à 100 codons).**

>BS 1-8301
tttcgaggaaaatgtgcaataaccaactcatttcccgggcaattccgccg
gttccgaatgatacgaacaactgagactgagccgcaaatggttcagtctt
tttacatggcagccagagggctttgtgcacttgacatttgtgaaaaagaa
agtaaaatattttactaaaacaatgcgagctgaataatggaggcagatac
aatggcgacaattaaagatatcgcgcaggaagcgggattttcaatctcaa
ccgtttcccgcgttttaaataacgatgaaagcctttctgttcctgatgag
acacgggagaaaatctatgaagcggcggaaaagctcaattaccgcaaaaa
aacagtaaggccgctggtgaaacatattgcgtttttatattggctgacag
ataaagaagaattagaagatgtctattttaaaacgatgagattagaagta
gagaaactggcgaaagcattcaatgtcgatatgaccacttataaaatagc
ggatggaatcgagagcattcctgaacatacggaagggtttattgccgtcg
gcacattttcagatgaagagctggctttcctcagaaatctcactgaaaac
ggcgtgttcatcgattcaactcctgatcccgatcattttgactcggtaag
gcccgatttggcacaaatgacaaggaagacggtaaacatcctgactgaga
aggggcataagagcatcggttttatcggcggcacatacaaaaatccgaat
accaatcaggatgaaatggacatccgtgaacaaaccttcagatcctatat
gagggaaaaagccatgctggacgagcgctatattttctgtcatcgcggat
tctctgtagaaaacggctaccgcctgatgtcagcagcgatcgacacatta
ggcgatcagcttccgactgcttttatgattgcagcggacccgattgcagt
gggctgtctgcaagccctgaacgaaaaaggaattgccataccaaacaggg
taagcattgtgagtatcaacaacatcagcttcgcgaagtatgtctcgcct
cctctgacgacgtttcatattgatatacatgaattatgtaaaaacgctgt
tcaattactgcttgaacaagtgcaggacaagagaagaacggtaaaaacat
tatatgtgggcgcagaattaatcgtcaggaagagtatgaattaaggatga
cttaggacactaagtcatttttatttaggtaaaaaaatttactctatga
agtaaatagtttgtttacacattttctcaggcatgctatattatctttaa
agcgctttcattcctaccgaaagggtgacaatcaatgaaaatggcaaaaa
agtgttccgtattcatgctctgcgcagctgtcagtttatccttggcggct
tgcggcccaaaggaaagcagcagcgccaaatcgagttcaaaagggtcaga
gcttgttgtatgggaggataaagaaaagagcaacggcattaaagacgctg
tggctgcatttgaaaaagagcatgatgtgaaggtcaaagtcgttgaaaaa
ccgtatgccaagcagattgaagatttgcgaatggatggaccggccggcac
aggccctgacgtgttaacaatgccaggggaccaaatcggaaccgctgtca
cggaaggattactcaaggaattacatgtcaaaaaagacgttcaatcactt
tatactgacgcttccattcagtctcaaatggtagatcaaaagctttatgg
actgccaaaagcggtcgaaacgactgtgctttttacaacaaagatctca
tcacagaaaaggaattgcccaaaacgctggaagagtggtacgactattcc

# Exemple traité : fragment de 8300 pb du génome de *Bacillus subtilis*

## Open Reading Frame Finder


ORF finder searches for open reading frames (ORFs) in the DNA sequence you enter. The program returns the range of each ORF, along with its protein translation. Use ORF finder to search newly sequenced DNA for potential protein encoding segments, verify predicted protein using newly developed SMART BLAST or regular BLASTP.

This web version of the ORF finder is limited to the subrange of the query sequence up to 50 kb long. Stand-alone version, which doesn't have query sequence length limitation, is available for Linux x64.

**Examples** (click to set values, then click Submit button) :

- NC_011604 Salmonella enterica plasmid pWES-1; genetic code: 11; 'ATG' and alternative initiation codons; minimal ORF length: 300 nt
- NM_000059; genetic code: 1; start codon: 'ATG only'; minimal ORF length: 150 nt

### Enter Query Sequence

**Enter accession number, gi, or nucleotide sequence in FASTA format:**

```
>BS 1-8301
tttcgaggaaaatgtgcaataaccaactcatttcccgggcaattccgccggttccgaatg
atacgaacaactgagactgagccgcaaatggttcagtctttttacatggcagccagaggg
ctttgtgcacttgacatttgtgaaaaagaaagtaaaatattttactaaaacaatgcgagc
tgaataatggaggcagatacaatggcgacaattaaagatatcgcgcaggaagcgggattt
tcaatctcaaccgtttcccgcgttttaaataacgatgaaagcctttctgttcctgatgag
acacgggagaaaatctatgaagcggcggaaaagctcaattaccgcaaaaaaacagtaagg
ccgctggtgaaacatattgcgtttttatattggctgacagataaagaagaattagaagat
gtctattttaaaacgatgagattagaagtagagaaactggcgaaagcattcaatgtcgat
atgaccacttataaaatagcggatggaatcgagagcattcctgaacatacggaagggttt
attgccgtcggcacattttcagatgaagagctggctttcctcagaaatctcactgaaaac
```

From: To:

### Choose Search Parameters

**Minimal ORF length (nt):** 300

**Genetic code:** 1. Standard

**ORF start codon to use:**

- "ATG" only
- "ATG" and alternative initiation codons
- Any sense codon

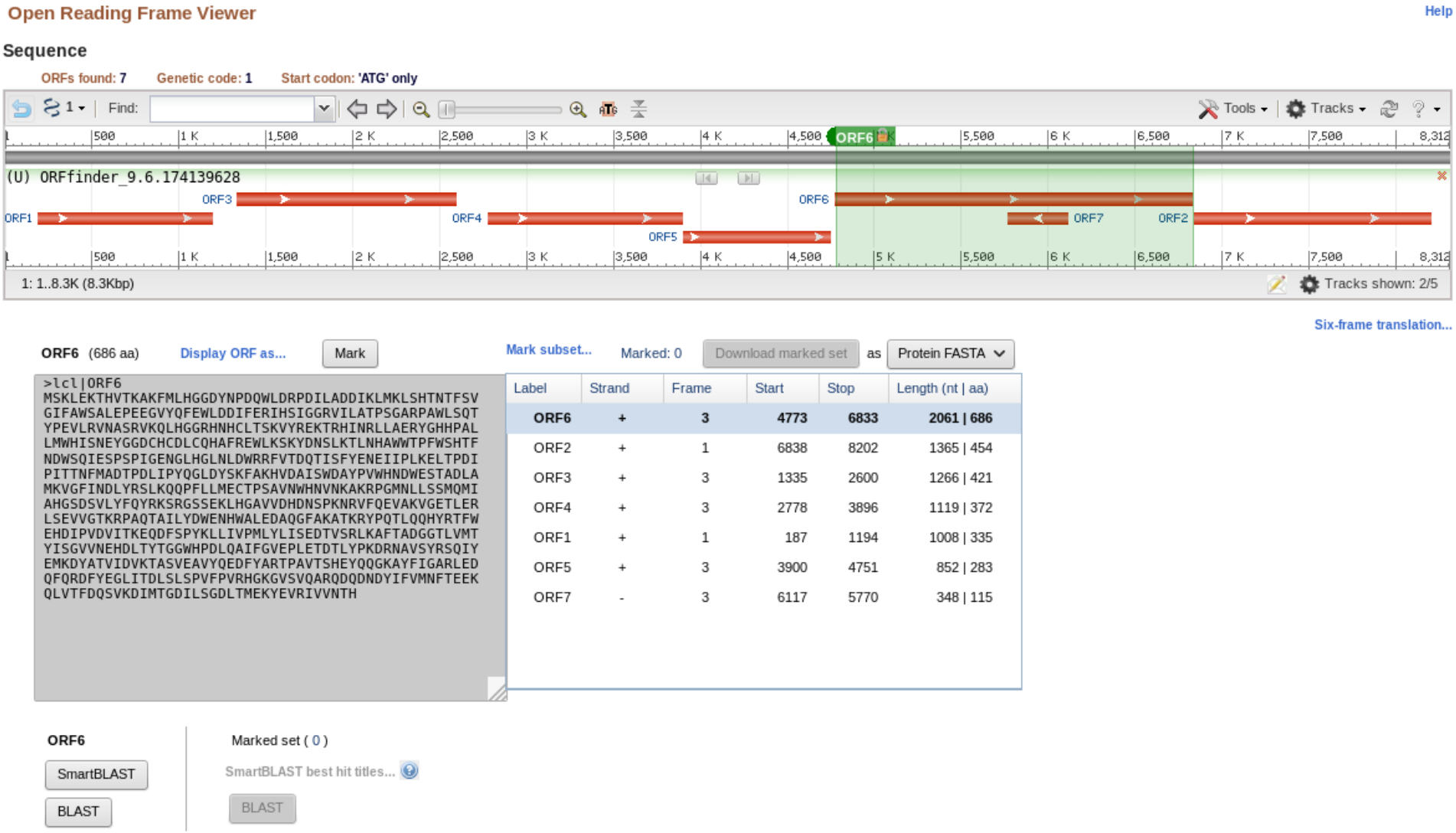
**Ignore nested ORFs:**

### Start Search / Clear

Submit Clear

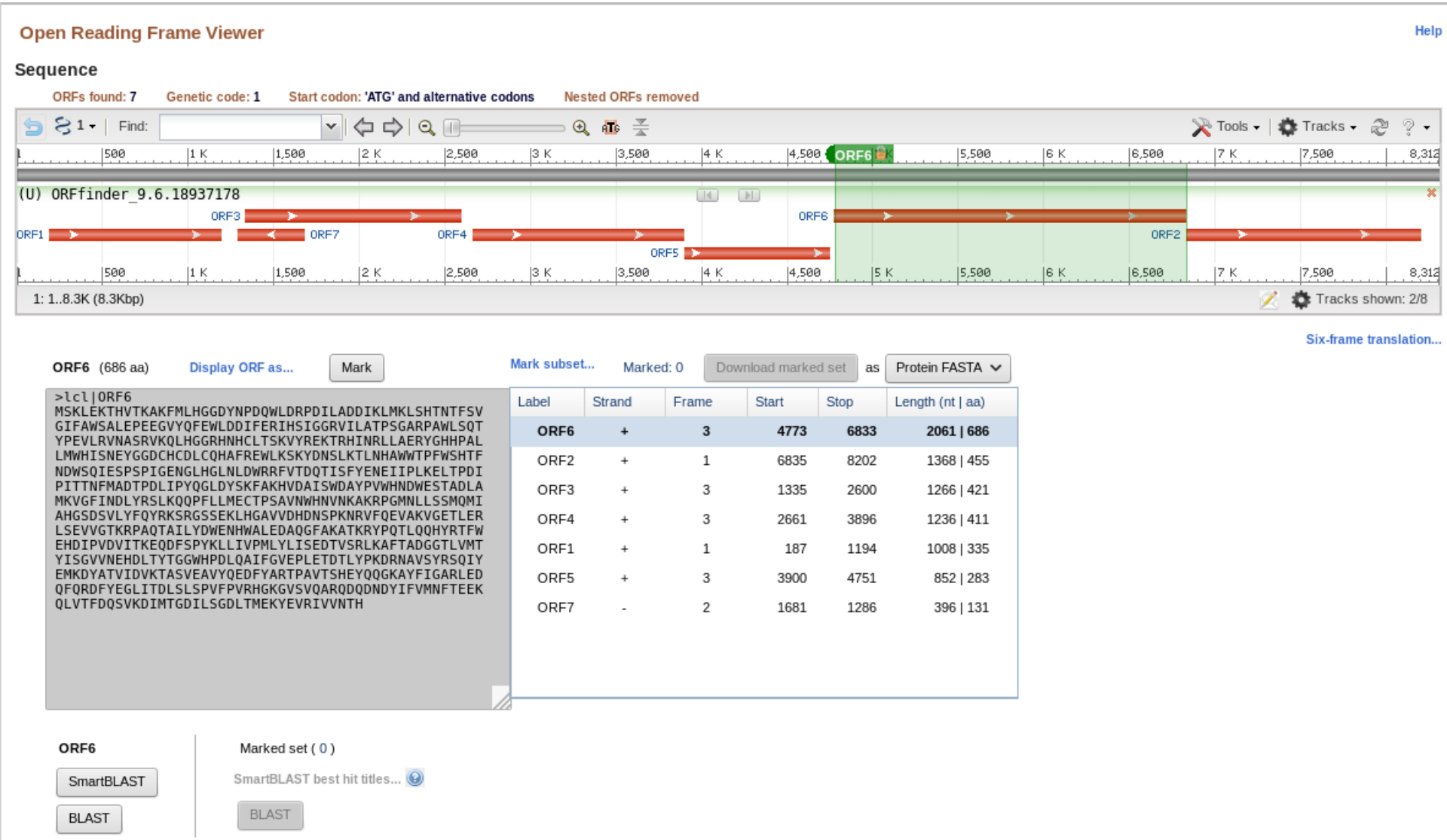
# Résultat de ORFfinder : ORFs de plus de 300 pb

Options : - ATG only
- Ignore nested ORF pas coché



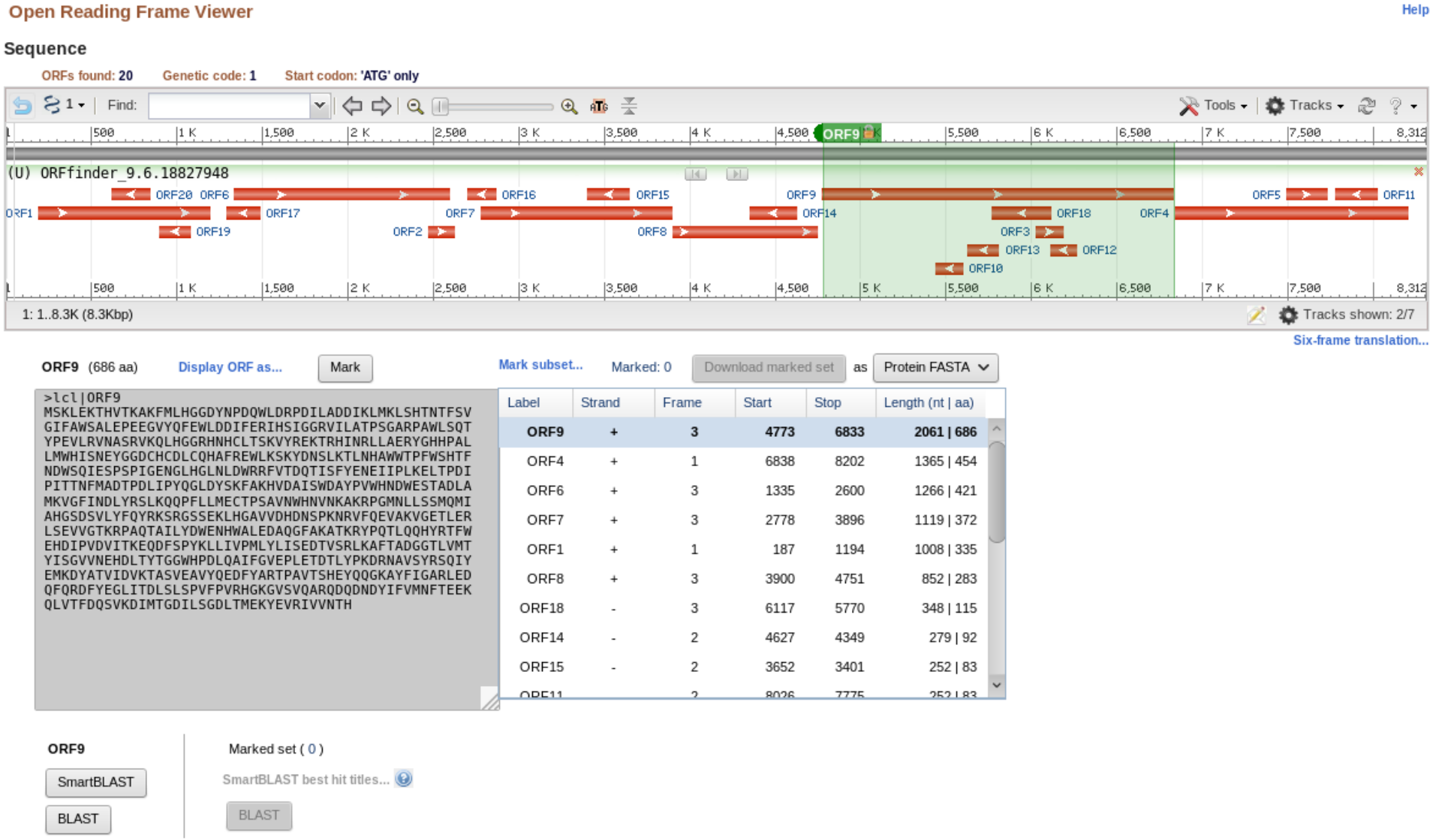
# Résultat de ORFfinder : ORFs de plus de 300 pb

Options : - ATG and alternative initiation codons
- Ignore nested ORF coché



# Résultat de ORFfinder : ORFs de plus de 150 pb

Options : - ATG and alternative initiation codons
- Ignore nested ORF coché



Les codons initiateurs alternatifs chez les procaryotes sont GTG et TTG (chez *B. subtilis* GTG 13%, TTG 9%)

## Limites d'ORFfinder :

- ne prend pas en compte le biais de l'utilisation des triplets existant dans les phases codantes car structurées en codons.

# Traitement de l'information de type contenu

Prise en compte du biais de l'utilisation des triplets existant dans les phases codantes par rapport aux régions non codantes car structurées en codons.

Biais dans l'utilisation des codons dus à :

- la différence de fréquence des acides aminés (Leu plus fréquent que Trp par exemple)

- la dégénérescence du code génétique (61 codons -> 20 aa)

- pour un acide aminé donné, certains codons peuvent être plus fréquemment utilisés que d'autres. Ces préférences varient en fonction :

    - la composition en bases de l'organisme ou de la région génomique (isochores chez les vertébrés) (riche ou pauvre en C+G)

    - du taux d'expression du gène : il a été montré chez *E. coli* que les gènes fortement exprimés utilisaient préférentiellement certains codons correspondant aux ARNt les plus abondants dans la cellule (efficacité de la traduction, coadaptation codons/ARNt.

# Exemples d'usage des codons chez les procaryotes

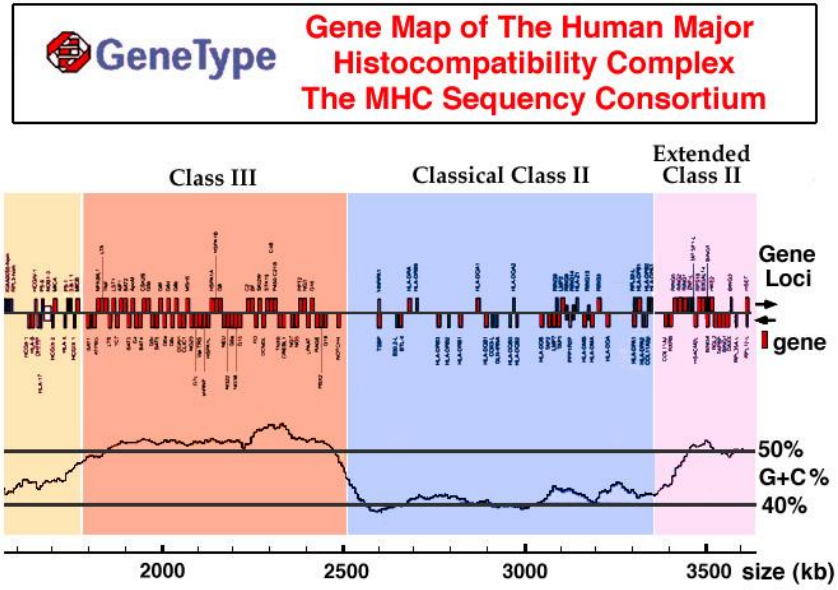
| Espèce | GC% codant | GC% 1ère pos. codon | GC% 2ème pos. codon | GC% 3ème pos. codon |
|---|---|---|---|---|
| *Synechocystis sp.* | 48.25 | 55.82 | 39.74 | 49.19 |
| *Streptomyces coelicolor* | 72.30 | 72.67 | 51.39 | 92.83 |
| *Escherichia coli 0157:H7* | 51.54 | 58.44 | 41 | 55.17 |
| *Bacillus subtilis* | 44.36 | 52.10 | 36.08 | 44.91 |

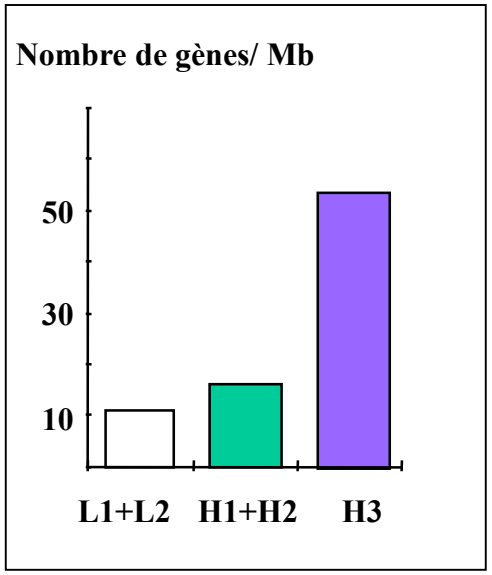
| A.A. | codon | % S. sp. (cyano) | % S. coelicolor | % E. coli | % B. subtilis |
|---|---|---|---|---|---|
| Gly | GGG | 0.24 | 0.19 | 0.164 | 0.16 |
| Gly | GGA | 0.18 | 0.075 | 0.123 | 0.315 |
| Gly | GGT | 0.27 | 0.096 | 0.331 | 0.187 |
| Gly | GGC | 0.31 | 0.64 | 0.382 | 0.337 |
| Glu | GAG | 0.264 | 0.846 | 0.325 | 0.32 |
| Glu | GAA | 0.736 | 0.154 | 0.675 | 0.68 |
| Asp | GAT | 0.646 | 0.05 | 0.631 | 0.636 |
| Asp | GAC | 0.354 | 0.95 | 0.369 | 0.364 |

# Modèle de la structure en isochores chez les vertébrés

Isochores : régions > 300 kb homogène dans sa composition en bases
5 types d'isochores en fonction de leur pourcentage en G+C (2 légers et 3 lourds)

| Isochore | %C+G | % total genomic DNA |
|---|---|---|
| L1+L2 | 33%-44% | 62 % |
| H1+H2 | 44%-51% | 31% |
| H3 | 51%-60% | 3-5% |

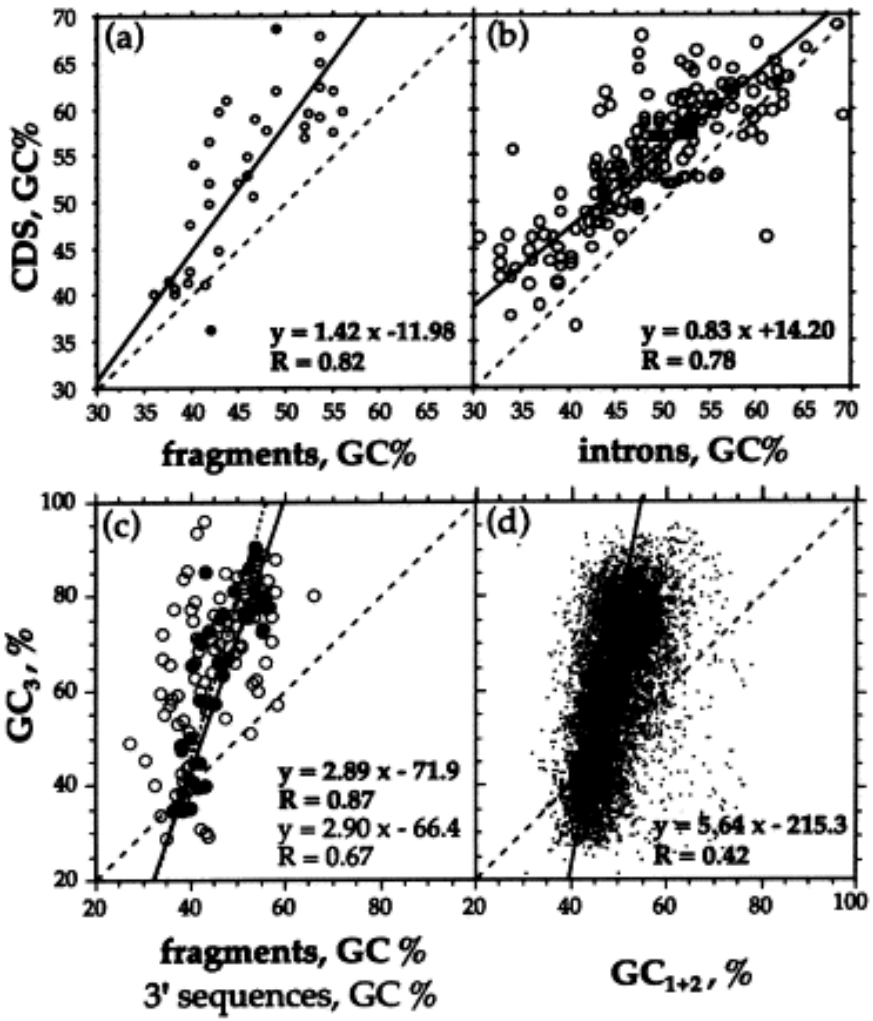



MHC locus (3.6 Mb) (The MHC sequencing consortium 99)

- Class I, class II (H1-H2 isochores): 20 gènes/Mb, beaucoup de pseudogènes
- Class III (H3 isochore): 84 gènes/Mb, pas de pseudogène

# Modèle de la structure en isochores chez les vertébrés




Correlation between GC levels of human coding sequences and :

(a) the GC levels of the large DNA fragments in which sequences were localized, or

(b) the GC levels of the corresponding introns (top frames).

The bottom frames show the correlations between $GC_3$ of human coding sequences and

(c) the GC levels of the DNA fractions in which the genes were localized (filled circles) and of 3′ flanking sequences further than 500 bp from the stop codon (open circles; the solid and the broken lines are the regression lines through the two sets of points); or

(d) $GC_1$+$GC_2$ values of human sequences. Diagonals (unity slope lines) are also shown

Extrait de Bernardi, 2000, *Gene*, 241 3-17

# Traitement de l'information de type contenu

Utilisation de méthodes statistiques prenant en compte ces biais d'utilisation des codons. Plus récemment avec l'augmentation des données pour établir les systèmes de référence, prise en compte de la composition en hexanucléotides (mots de longueur 6).

Les méthodes statistiques suivantes seront abordées :

- Modèles de Markov
- Modèles de Markov interpolés (IMM)
- Modèles de Markov caché (HMM)

# Modèle de Markov : Présentation de GeneMark

(Borodovsky et al., Nucleic Acids Res.,22,4756-67)

La méthode repose sur le modèle probabiliste suivant appelé modèle de Markov:

Hypothèse 1: La probabilité d'observer une base à une position donnée dépend:

- des bases précédant cette position
- de sa localisation dans le codon

Modélisé par

modèle de Markov homogène pour les régions non-codantes.
modèle de Markov non-homogène pour les séquences codantes.

Hypothèse 2: Une région particulière ne peut être que dans un des 7 états suivants:

- 1. codant en phase 1 sur le brin direct
- 2. codant en phase 2 sur le brin direct
- 3. codant en phase 3 sur le brin direct
- 4. codant en phase 4 sur le brin indirect
- 5. codant en phase 5 sur le brin indirect
- 6. codant en phase 6 sur le brin indirect
- 7. non-codant

Prédiction : calculer les probabilités d'observer la région dans un état *i* sachant que l'un des 7 états est réalisé (formule de Bayes).

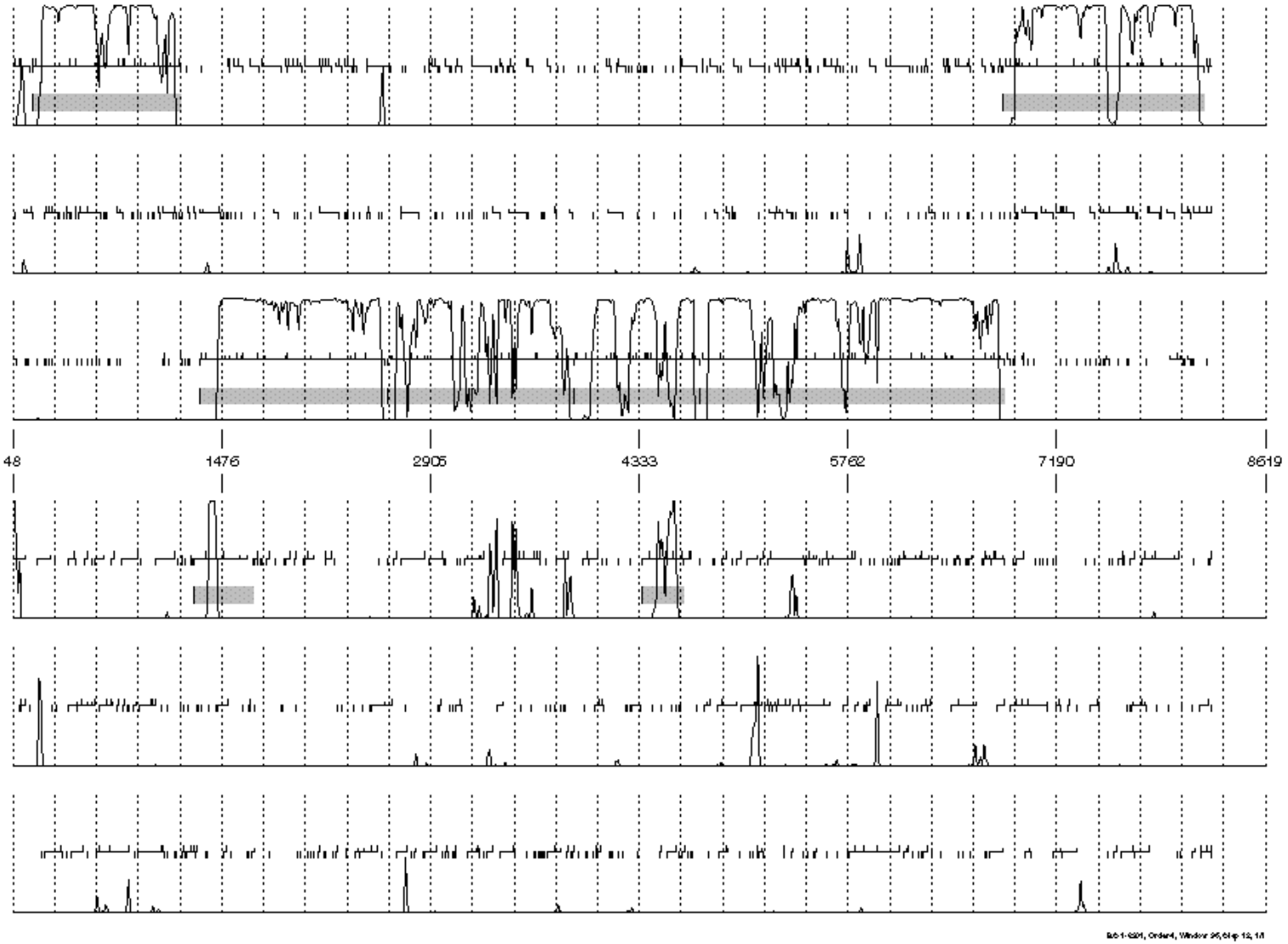
# Modèle de Markov

Un modèle de Markov d'ordre *k* appliqué aux séquences ADN est entièrement défini par les deux probabilités suivantes :

- $P_0(w_1^k)$ ⟶ Probabilité initiale du mot $w^k$
- $P(x / w^k)$ ⟶ Probabilité d'observer x sachant que le mot $w^k$ le précède

Modèle probabiliste qui représente une séquence comme un processus qui peut être décrit comme une séquence de variable aléatoire $X_1, X_2, \ldots$ où $X_i$ correspond à la position $i$ de la séquence. Chaque variable aléatoire $X_i$ prend une valeur dans l'ensemble des bases (A,C,G,T). La probabilité que va prendre la variable $X_i$ dépend du contexte c'est à dire des bases immédiatement adjacentes à la base à la position $i$.

# Résultat graphique de GeneMark sur le fragment de *B. subtilis*



# Résultat de GeneMark sur le fragment de *B. subtilis*

**Entête du fichier :**

```
Sequence: EMBOSS_001 Reversed:
Sequence file: seq.fna
Sequence length: 8312
GC Content:  45.19%
Window length: 96
Window step: 12
Threshold value: 0.500
---
Matrix: Bacillus_subtilis_168
Matrix author: -
Matrix order: 4
```

**Fin du fichier :**

**List of Regions of interest**
(regions from stop to stop codon w/ a signal in between)

| LEnd | REnd | Strand | Frame |
|---|---|---|---|
| 181 | 1194 | direct | fr 1 |
| 1286 | 1693 | complement | fr 1 |
| 1326 | 2600 | direct | fr 3 |
| 2610 | 3896 | direct | fr 3 |
| 3894 | 4751 | direct | fr 3 |
| 4749 | 6833 | direct | fr 3 |
| 6820 | 8202 | direct | fr 1 |

GENEMARK PREDICTIONS

**List of Open reading frames predicted as CDSs,** shown with alternate starts
(regions from start to stop codon w/ coding function >0.50)

| Left end | Right end | DNA Strand | Coding Frame | Avg Prob | Start Prob | |
|---|---|---|---|---|---|---|
| 187 | 1194 | direct | fr 1 | 0.80 | 0.99 | **-> ORF Finder** |
| 202 | 1194 | direct | fr 1 | 0.81 | 0.89 | |
| 367 | 1194 | direct | fr 1 | 0.82 | 0.29 | |
| 436 | 1194 | direct | fr 1 | 0.81 | 0.03 | |
| 481 | 1194 | direct | fr 1 | 0.80 | 0.02 | |
| 1335 | 2600 | direct | fr 3 | 0.85 | 0.01 | **-> ORF Finder** |
| 1341 | 2600 | direct | fr 3 | 0.85 | 0.00 | |
| 1365 | 2600 | direct | fr 3 | 0.87 | 0.08 | |
| 1500 | 2600 | direct | fr 3 | 0.93 | 0.07 | |
| 1527 | 2600 | direct | fr 3 | 0.93 | 0.00 | |
| 1581 | 2600 | direct | fr 3 | 0.92 | 0.03 | |
| 2631 | 3896 | direct | fr 3 | 0.73 | 0.67 | |
| 2640 | 3896 | direct | fr 3 | 0.73 | 0.77 | |
| 2778 | 3896 | direct | fr 3 | 0.76 | 0.53 | **-> ORF Finder** |
| 2814 | 3896 | direct | fr 3 | 0.75 | 0.02 | |
| 2868 | 3896 | direct | fr 3 | 0.74 | 0.40 | |
| 3900 | 4751 | direct | fr 3 | 0.65 | 0.17 | **-> ORF Finder** |
| 3912 | 4751 | direct | fr 3 | 0.66 | 0.02 | |
| 3966 | 4751 | direct | fr 3 | 0.71 | 0.34 | |
| 4116 | 4751 | direct | fr 3 | 0.71 | 0.11 | |
| 4137 | 4751 | direct | fr 3 | 0.70 | 0.02 | |
| 4158 | 4751 | direct | fr 3 | 0.69 | 0.06 | |
| 4770 | 6833 | direct | fr 3 | 0.85 | 0.76 | |
| 4773 | 6833 | direct | fr 3 | 0.85 | 0.82 | **-> ORF Finder** |
| 4815 | 6833 | direct | fr 3 | 0.86 | 0.12 | |
| 4890 | 6833 | direct | fr 3 | 0.85 | 0.05 | |
| 5226 | 6833 | direct | fr 3 | 0.85 | 0.01 | |
| 6838 | 8202 | direct | fr 1 | 0.79 | 0.03 | **-> ORF Finder** |
| 6877 | 8202 | direct | fr 1 | 0.82 | 0.86 | |
| 6913 | 8202 | direct | fr 1 | 0.83 | 0.67 | |
| 6925 | 8202 | direct | fr 1 | 0.83 | 0.01 | |
| 6931 | 8202 | direct | fr 1 | 0.83 | 0.01 | |
| 6952 | 8202 | direct | fr 1 | 0.84 | 0.00 | |
| 7009 | 8202 | direct | fr 1 | 0.85 | 0.63 | |
| 7057 | 8202 | direct | fr 1 | 0.86 | 0.28 | |

# Interpolated Markov Model (IMM)

## Glimmer (Salzberg et *al.*, Nucleic Acids Res.,26,544-48)

Modèle de Markov d'ordre $k$ : apprendre $4^{k+1}$ probabilités
Dans le cadre de la prédiction des CDS prise en compte des 6 cadres de lecture, donc nécessité d'apprendre 6 * $4^{k+1}$ probabilités
Si modèle de Markov d'ordre 5 : 4096 probabilités à définir (hexamères)
Si on considère les 6 cadres de lecture : 24 576 probabilités

Plus l'ordre du modèle est élevé, moins l'estimation des paramètres du modèles va être fiable

Pour certains kmers rares même avec un grand jeu d'apprentissage comme un génome entier, il peut être difficile d'obtenir des estimations précises et inversement certains kmers fréquents même avec un modèle de markov d'ordre élevé des estimations précises peuvent être obtenues.

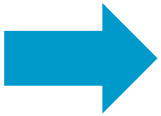
Souhait : un modèle de Markov qui utilise les ordres les plus élevés quand il y a assez de données disponibles et des ordres moins élevés dans les cas où les données sont insuffisantes.

Interpolation des modèles de Markov

# Interpolated Markov Model (IMM)

A partir d'un ensemble d'apprentissage :

- ✓ Identification de toutes les "Open Reading Frames" (ORFs) dans les 6 cadres avec une longueur > seuil (~ 90 bp par défaut)
- ✓ Sélection des ORFs qui vont constituer la banque de référence
  - à partir de gènes connus (expérimentalement)
  - à partir du génome { longueur > 500 bp ; absence de chevauchement avec une ORF > 500 bp }



**Obtenir un ensemble importants de gènes "fiables"**

- ✓ Calcul des fréquences observées des oligomères (longueur 1 à 9)
- ✓ Estimation de la probabilité d'occurrence d'une base connaissant son contexte (i bases précédentes avec i<=8)

**exemple : probabilité du 5mer ATTCA**

$$P(ATTCA) = P(A|ATTC) = \frac{f(ATTCA)}{f(ATTCA) + f(ATTCC) + f(ATTCG) + f(ATTCT)}$$

*avec f fréquence observée*

# Interpolated Markov Model (IMM)

Interpolation linéaire simple : combinaison linéaire des probabilités associées aux mots de tailles inférieures à $k$ contenus dans $w_k$

$$P_{IMM}(x_i|x_{i-n},\dots,x_{i-1}) = \lambda_0 P(x_i) + \lambda_1 P(x_i|x_{i-1}) + \cdots + \lambda_n P(x_i|x_{i-n}\dots x_{i-1})$$

avec $\sum_i \lambda_i = 1$

Les poids peuvent aussi dépendre des données observées. Pour un ordre donné, on peut avoir plus de données pour estimer certains mots que d'autres.

$$P_{IMM}(x_i|x_{i-n},\dots,x_{i-1}) = \lambda_0 P(x_i) + \lambda_1(x_{i-1})P(x_i|x_{i-1}) + \cdots + \lambda_n(x_{i-n},\dots,x_{i-1})\, P(x_i|x_{i-n}\dots x_{i-1})$$

$\lambda$ est une fonction des données observées. Pour un ordre donné, on peut avoir plus de données pour estimer certains mots que d'autres.

# Interpolated Markov Model (IMM)

Autrement dit , un IMM utilise une combinaison de toutes les probabilités basées sur $0, 1, 2, \ldots, k$ bases précédentes, où $k$ est un paramètre donné à l'algorithme. Dans le cas de GLIMMER $k$ = 8
Donc pour les oligomères fréquents, le IMM peut utiliser un modèle d'ordre 8 alors qu'il pourra utiliser par exemple un modèle d'ordre 5 voir d'un ordre inférieur pour des oligomères rares.
Afin de " lisser " ses prédictions, un IMM utilise les prédictions des modèles d'ordre inférieur, pour lesquels on dispose de beaucoup plus de données, afin d'ajuster les prédictions faites à partir des modèles d'ordre supérieur.

Donc les poids attribués aux différents modèles vont définir le poids des modèles inférieurs dans le calcul de la probabilité du modèle d'ordre supérieur.

Ces poids sont appelés les paramètres d'interpolation, avec $0 \leq \lambda \leq 1$.

# Interpolated Markov Model (IMM)

Dans Glimmer, utilisation d'un IMM non homogène d'ordre 8

Le calcul de la probabilité se fait de façon récursive :

$$P_{IMM,n}(x_i|x_{i-n},\dots,x_{i-1}) = \lambda_n(x_{i-n},\dots,x_{i-1})P(x_i|x_{i-n},\dots,x_{i-1}) + [1-\lambda_n(x_{i-n},\dots,x_{i-1})]P_{IMM,n-1}(x_i|x_{i-n+1},\dots,x_{i-1})$$

Avec $\lambda_n(x_{i-n},\dots,x_{i-1})$ le poids associé au mot $(x_{i-n},\dots,x_{i-1})$

Soit $c(x_{i-n},\dots,x_{i-1})$ le nombre de mots correspondant à $x_{i-n},\dots,x_{i-1}$ dans notre ensemble d'apprentissage :

Si $c(x_{i-n},\dots,x_{i-1}) > 400$ alors $\lambda_n(x_{i-n},\dots,x_{i-1}\,x_{i-n},\dots,x_{i-1}) = 1$

Si ce n'est pas le cas, utilisation d'un test statistique

# Interpolated Markov Model (IMM)

On va comparer les occurrences des mots entre deux ordres pour savoir si elles dépendent de l'ordre du modèle de Markov

| modèle ordre n + base | modèle ordre n-1 + base |
|---|---|
| $x_{i-n}, \ldots, x_{i-1}$, a | $x_{i-n+1}, \ldots, x_{i-1}$, a |
| $x_{i-n}, \ldots, x_{i-1}$, c | $x_{i-n+1}, \ldots, x_{i-1}$, c |
| $x_{i-n}, \ldots, x_{i-1}$, g | $x_{i-n+1}, \ldots, x_{i-1}$, g |
| $x_{i-n}, \ldots, x_{i-1}$, t | $x_{i-n+1}, \ldots, x_{i-1}$, t |

Utilisation du test du $\chi^2$ :
Hypothèse nulle : la distribution des valeurs de $x_i$ est indépendante de l'ordre du modèle

Soit $d = 1 - pvaleur$ (si pvaleur grande on ne peut pas rejeter $H_0$)

Si $d$ est petit, pas besoin de prendre en compte l'ordre le plus grand

# Interpolated Markov Model (IMM)

Résumé de la détermination des poids $\lambda$ dans GLIMMER

$$\lambda_n(x_{i-n}, \dots, x_{i-1}) = \begin{cases} 1 & si\ c(xi-n, \dots, xi-1) > 400 \\ 0 & si\ d < 0.5 \\ d \times \dfrac{c(xi-n, \dots, xi-1)}{400} & si\ d \geq 0.5 \end{cases}$$

# Interpolated Markov Model (IMM)

Exemple : Supposons que nous avons les fréquences observées des mots suivants dans notre ensemble d'apprentissage

| ordre 3 | | ordre 2 | | ordre 1 | |
|---|---|---|---|---|---|
| ATCA | 25 | TCA | 100 | CA | 175 |
| ATCC | 40 | TCC | 90 | CC | 140 |
| ATCG | 15 | TCG | 35 | CG | 65 |
| ATCT | 20 | TCT | 75 | CT | 120 |
| total | 100 | | 300 | | 500 |

$\chi^2$ test : $d = 0.857$ $d = 0.140$

$$\lambda_3(\mathrm{ATC}) = 0.857 \times 100/400 = 0.214$$

$$\lambda_2(\mathrm{TC}) = 0 \ (\mathrm{d} < 0.5 \text{ et } \mathrm{c} < 400)$$

$$\lambda_1(\mathrm{C}) = 1 \ (\mathrm{c} > 400)$$

# Interpolated Markov Model (IMM)

<u>Rappel :</u> $$P_{IMM,n}(x_i|x_{i-n},\dots,x_{i-1}) = \lambda_n(x_{i-n},\dots,x_{i-1})P(x_i|x_{i-n},\dots,x_{i-1}) + [1-\lambda_n(x_{i-n},\dots,x_{i-1})]P_{IMM,n-1}(x_i|x_{i-n+1},\dots,x_{i-1})$$

Si on veut calculer $P_{\mathrm{IMM},3}(\mathrm{G}|\mathrm{ATC})$ :

$$P_{IMM,1}(G|C) = \lambda_1(C)P(G|C) + [1-\lambda_1(C)]P_{IMM,0}(G) = P(G|C)$$ car $\lambda_1(C) = 1$

$$P_{IMM,2}(G|TC) = \lambda_2(TC)P(G|TC) + [1-\lambda_2(TC)]P_{IMM,1}(G|C) = P_{IMM,1}(G|C) = P(G|C)$$

car $\lambda_2(TC) = 0$

$$P_{IMM,3}(G|ATC) = \lambda_3(ATC)P(G|ATC) + [1-\lambda_3(ATC)]P_{IMM,2}(G|TC)$$
$$= 0.214\,P(G|ATC) + (1-0.214)\,P_{IMM,2}(G|TC)$$

car $\lambda_3(ATC) = 0.214$

$$P_{IMM,3}(G|ATC) = 0.214\,P(G|ATC) + 0.786\,P(G|C)$$

# Interpolated Markov Model (IMM)

$$P_{IMM,3}(G|ATC) = 0.214\,P(G|ATC) + 0.786\,P(G|C)$$

$$P(G|C) = P(CG) = \frac{f(CG)}{f(CA) + f(CC) + f(CG) + f(CT)} = \frac{65}{500} = 0.13$$

$$P(G|ATC) = P(ATCG) = \frac{f(ATCG)}{f(ATCA) + f(ATCC) + f(ATCG) + f(ATCT)} = \frac{15}{100} = 0.15$$

$$P_{IMM,3}(G|ATC) = 0.214 \text{ x } 0.15 + 0.786 \text{ x } 0.13 = 0.13428$$

# Identification des CDS avec GLIMMER

- ✓ n'utilise pas de fenêtre glissante
- ✓ Identifie en premier les ORF plus long qu'un certain seuil
- ✓ Calcul le score de chacun dans les six cadres de lecture comme suit :

La probabilité que le modèle M génère la séquence S est :

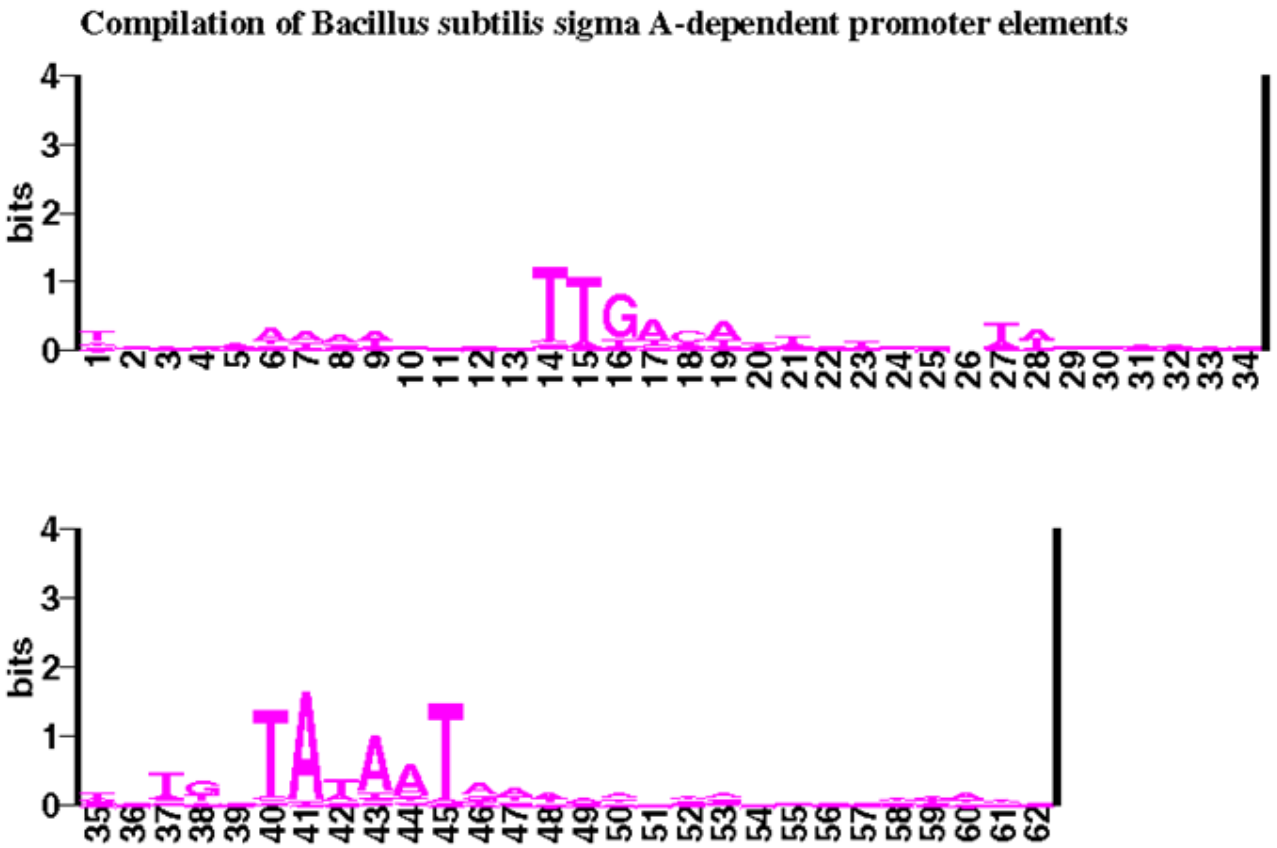
$$P(S|M) = \sum_{x=1}^{n} IMM_8(S_x)$$

Avec $S_x$ est l'oligomer se terminant à la position $x$ et $n$ la taille de la séquence

- ✓ Les ORF qui ont un score supérieur à un seuil spécifié dans le cadre de lecture correct sont sélectionnées pour un traitement ultérieur des chevauchements

# Traitement de l'information de type signal

Différentes façon de représenter la conservation des séquences impliquées dans un processus donné (promoteur lors de la transcription, ribosome binding site lors de la traduction, jonction d'épissage etc…) et ensuite de rechercher ces « signaux » dans une nouvelle séquence.



# Théorie de l 'information

Shannon et Weaver (1949).

La valeur de l'information *I* à la position *j* d'un signal est donnée par :

$$I(j) = \sum_i f_{ij} \log_2 f_{ij} - \sum_i P_i \log_2 P_i$$

où :

$P_i$ ($i$ = 1 à 4) est la fréquence de la base *i* dans l'ensemble du génome (probabilité théorique)

$f_{ij}$ est la fréquence observée de la base *i* à la position *j* d'un signal sur un ensemble d'exemples.

Les $P_i$ étant estimées à 0.25 pour chacune des 4 bases on a :

$$\sum_i P_i \log_2 P_i = -2$$

donc

$$I(j) = \sum_i f_{ij} \log_2 f_{ij} + 2$$

Les positions du signal qui contiendront de l'information seront celles qui auront une composition très biaisées par rapport à ce qui est attendu.

Si à une position *j* du signal, présence d'une seule base invariante *i* alors $f_{ij} = 1$ et $\log_2 f_{ij} = 0$ donc

$f_{ij} \log_2 f_{ij} = 0$ et les fréquences observées des autres bases sont nulles. On aura

$I(j) = 2$ information maximale

# Recherche des signaux d'initiation de la traduction

**Programme utilisé: Scan_For_Matches**

Motif du Shine-Dalgarno recherché : **GGAGG 6...11 DTG** correspond à la présence de la séquence GGAGG à 6 ou 11 pb en amont d'un codon AUG, GUG ou UUG.

Résultats:

```
BS:[189,204]  :  ggagg cagataca   atg    ->    A
BS:[3175,3192]:  ggagg tcgacttttt  ttg    ->    dans le gène C
BS:[3887,3902]:  ggagg cataaggt   atg    ->    D
BS:[4760,4775]:  ggagg agaatgtg   atg    ->    E
BS:[7501,7516]:  ggagg atttgccg    gtg    ->    dans le gène F
```

Donc:

**Gène A : début en 202**
**Gène D : début en 3900**
**Gène E : début en 4773**

Les autres SD des gènes B, C et F trouvés avec une autre représentation (matrice de poids) car ils sont modifiés.

| | | |
|---|---|---|
| AAGGAGGTG | | consensus |
| G**A**AAG**GGTG** | 7 | ATG pour B |
| **A**GA **GAGGTG** | 6 | GTG pour C |
| GG**GG**G**G**A**TG** | 5 | ATG pour F |

**Unités de traduction prédites**

```
202       1194  direct      fr 1    -> A
1335      2600  direct      fr 3    -> B    SD modifié
2640      3896  direct      fr 3    -> C    SD modifié
3900      4751  direct      fr 3    -> D
4773      6833  direct      fr 3    -> E
6877      8202  direct      fr 1    -> F    SD modifié
```

# Recherche des unités de transcription

**Chez *B. subtilis,* l'initiation de la transcription fait intervenir le facteur sigma A qui reconnaît une séquence spécifique localisée environ en -10 et -35 pb du +1 de transcription.**

**Séquence consensus: TTGACA 16...35 TATAAT**

**Grand nombre de promoteurs de type sigma A identifiés expérimentalement chez *B. subtilis*:**

**matrices de poids**

# Représentation : Matrice de poids

**Exemples de 242 séquences de promoteurs (-10) chez *E. coli* :**

**Matrices des fréquences de chaque base *b* à chaque position *i* ($f_{b,i}$) du motif -10 (6 positions) :**

| Pos. | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| A | 0.04 | 0.88 | 0.26 | 0.59 | 0.49 | 0.03 |
| C | 0.09 | 0.03 | 0.11 | 0.13 | 0.22 | 0.05 |
| G | 0.07 | 0.01 | 0.12 | 0.16 | 0.12 | 0.02 |
| T | 0.80 | 0.08 | 0.51 | 0.13 | 0.18 | 0.89 |

**Avec** $f_{b,i} = n_{b,i} / n_{tot}$ $n_{tot}$ **: nombre total de séquences analysées**

# Représentation : Matrice de poids ou PWM (Position Weight Matrix)

Un exemple simple : 242 séquences de promoteurs (-10) chez *E. coli* :

**Normalisation de la matrice : log matrice $\log_2(f_{b,i}/P_b)$**

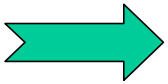
**$f_{b,i}$ = fréquence observée de la base *b* à la position *i* dans toutes les séquences**
**$P_b$ = fréquence de cette base dans l'ensemble du génome**

| Pos. | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| A | -2.76 | 1.88 | 0.06 | 1.23 | 0.96 | -2.92 |
| C | -1.46 | -3.11 | -1.22 | -1.00 | -0.22 | -2.21 |
| G | -1.76 | -5.00 | -1.06 | -0.67 | -1.06 | -3.58 |
| T | 1.67 | -1.66 | 1.04 | -1.00 | -0.49 | 1.84 |

**Le rapport $f_{b,i}/P_b$ est une mesure de l'écart entre fréquence observée et attendue.**

# Résultats de la recherche des promoteurs

**Utilisation du programme Scan_For_Matches et de la matrice de poids**



| | -35 | -10 |
|---|---|---|
| **BS:[1264,1292]:** | **tttaca** | **cattttctcaggcatgc** **tatatt** |
| **BS:[131,158] :** | **ttgaca** | **tttgtgaaaaagaaag** **taaaat** |

# Recherche des terminateurs de transcription

**2 types de terminateurs :**

- **Rho dépendant. Une protéine, le facteur ρ, aide au décrochage de l'ARN polymérase.**

- **Rho indépendant. Pas d'intervention de protéines.**

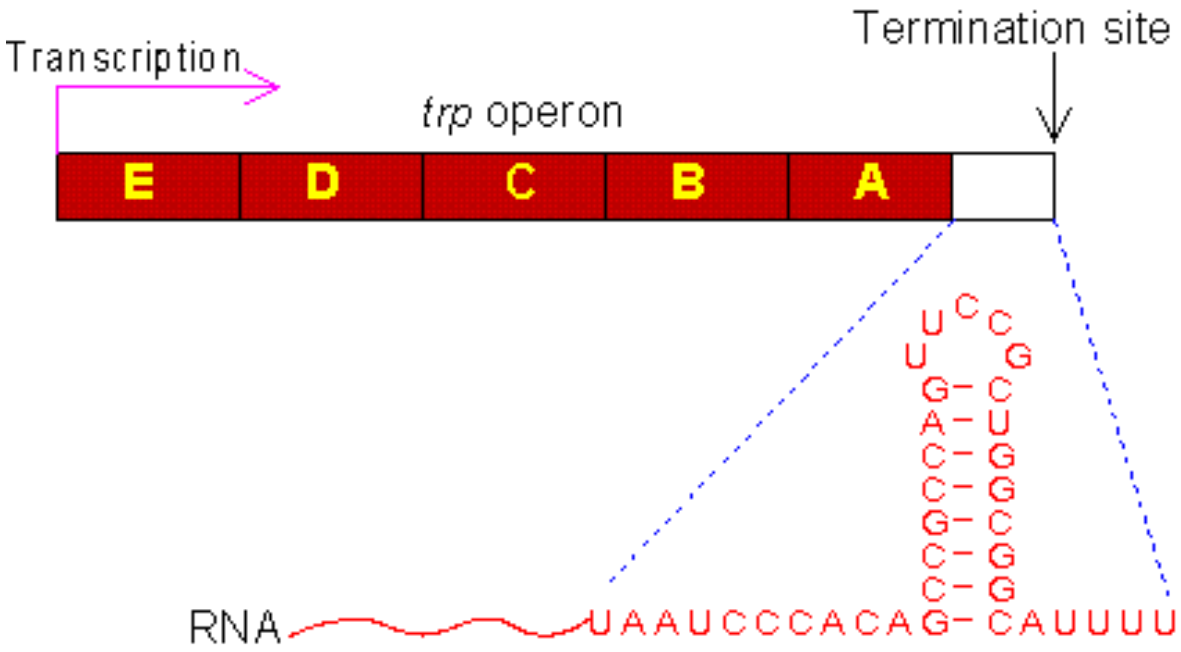
**Au niveau séquence, on ne sait modéliser que les seconds.**

**Mécanisme proposé pour les terminateurs Rho indépendant.**

**Quand l'ARN est en cours de transcription, on a une hybridation ARN/ADN sur environ 12 pb. Le site de terminaison de la transcription est précédé par une séquence capable de former une structure secondaire stable. Il y a compétition entre la formation de cette structure et l'appariement avec l'ADN. La présence d'un poly(U) en cours de synthèse déplace l'équilibre en faveur de la tige-boucle et il y a alors décrochage de l'ARN et arrêt de la transcription.**

**Dans les séquences, on va donc rechercher des séquences répétées inversées suivies d'un poly(U).**

# Terminateurs rho indépendant



# Terminateurs rho indépendant

Le modèle : Formation d'une tige boucle en amont d'une région riche en U qui déstabilise l'appariement ADN/ARN et conduit au décrochage de l'ARN.

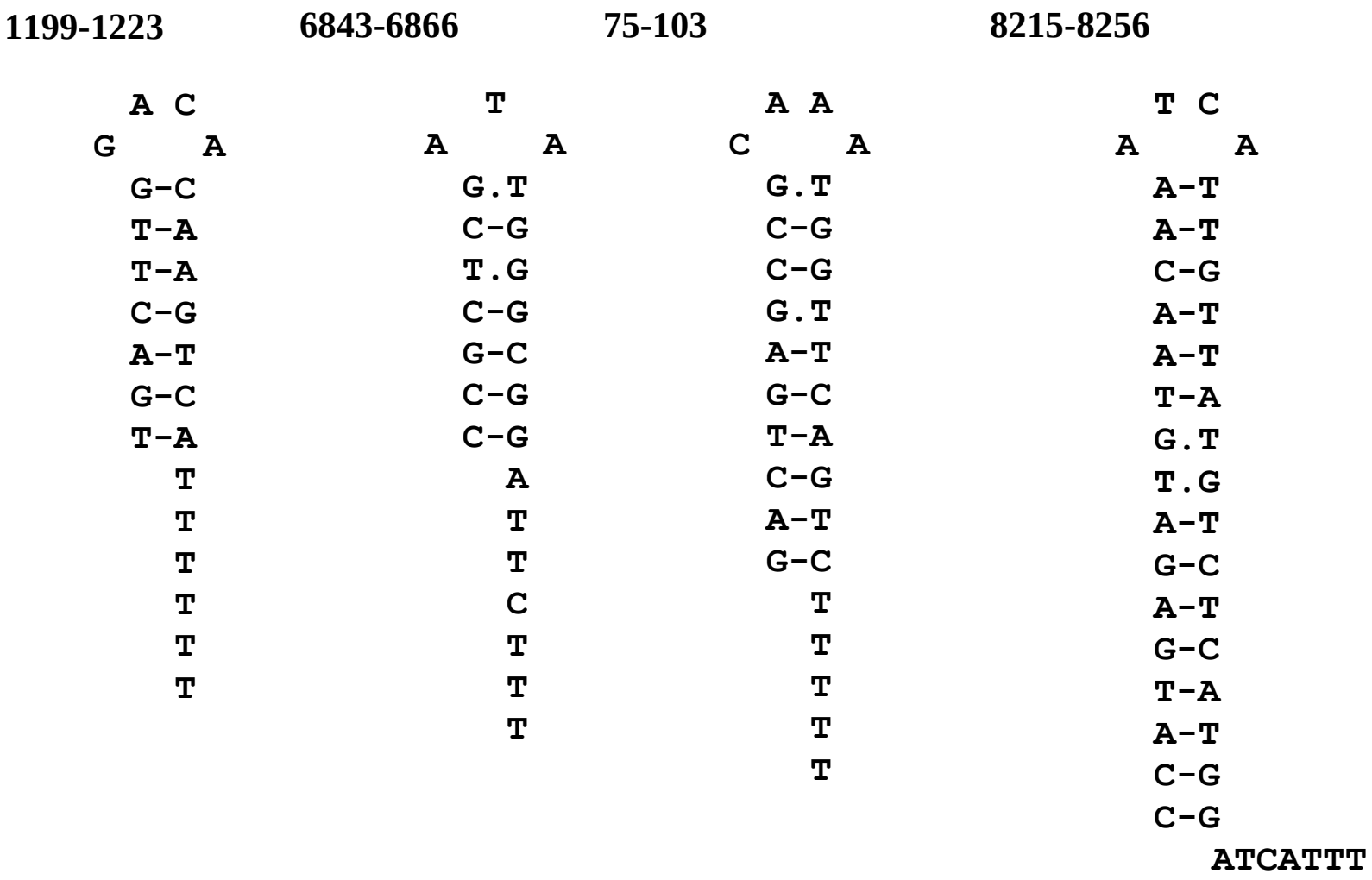
Deux classes de terminateurs:

•petite tige de 5 à 7 pb très stable et d'une boucle de 4 pb suivie d'une région riche en U.

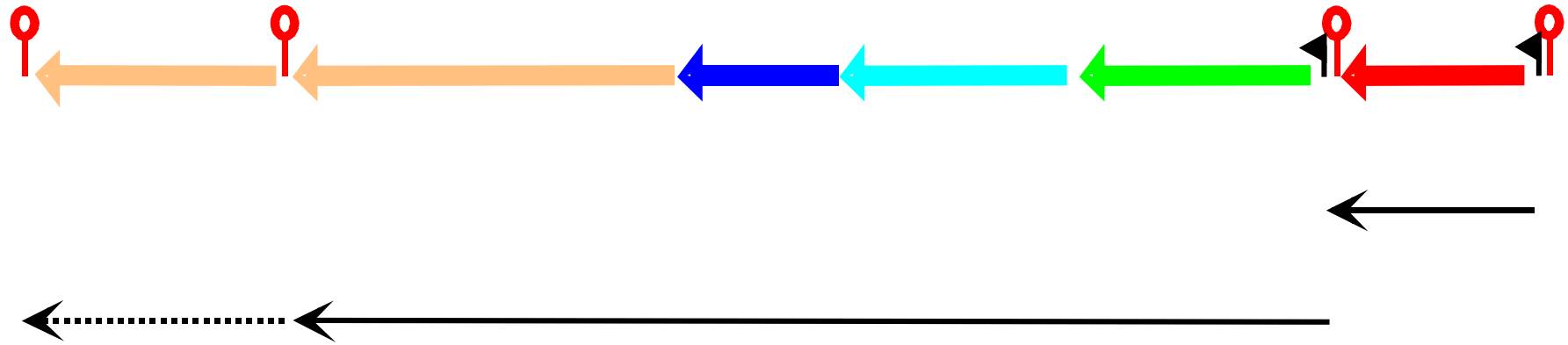
•une longue tige qui peut se décomposer en deux tiges imbriquées l'une dans l'autre.

- •La première plus stable doit faire au moins 3 pb de long avec un appariement GC à son pied.
- •La seconde est incluse dans la première et comporte au moins 3 appariements. Elle est généralement moins stable que la première. La boucle est de 3 à 7 pb de long.

# Résultat de la recherche des terminateurs sur le fragment de *B. subtilis*



# Prédiction des unités de traduction et de transcription



terminateur rho-indépendant

promoteurs de transcription de type sigma

transcrit putatif

# Prédictions fonctionnelles

Identification
- homologues
- motifs
- domaines

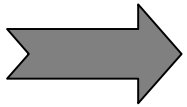
Localisation cellulaire
- fragments trans-membranaires
- peptide signal

Structure
- secondaire
- tertiaire

Recherche de liens fonctionnelles
- réseaux de régulation
- voies métaboliques
- interactions moléculaires

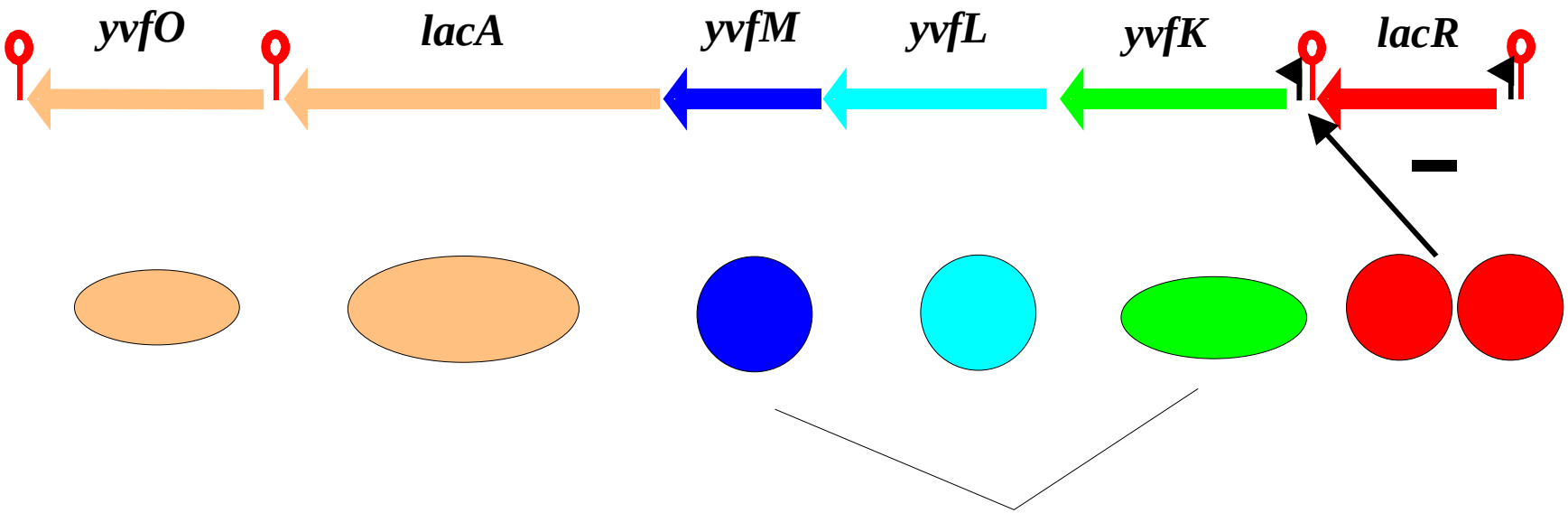
# Prédiction fonctionnelle

**Recherche par similitude dans les bases de données: programme BLAST**

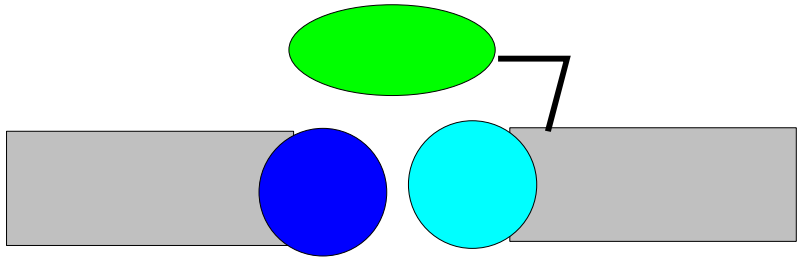


- **A LACR protéine régulatrice de type LacI/GalR**
- **B YVFK protéine affine d'un ABC transporteur**
- **C YVFL perméase d'un ABC transporteur**
- **D YVFM perméase d'un ABC transporteur**
- **E LACA galactosidase**
- **F YVFO arabino-galactosidase**

# Synthèse des résultats




**Système à la membrane**



? ?

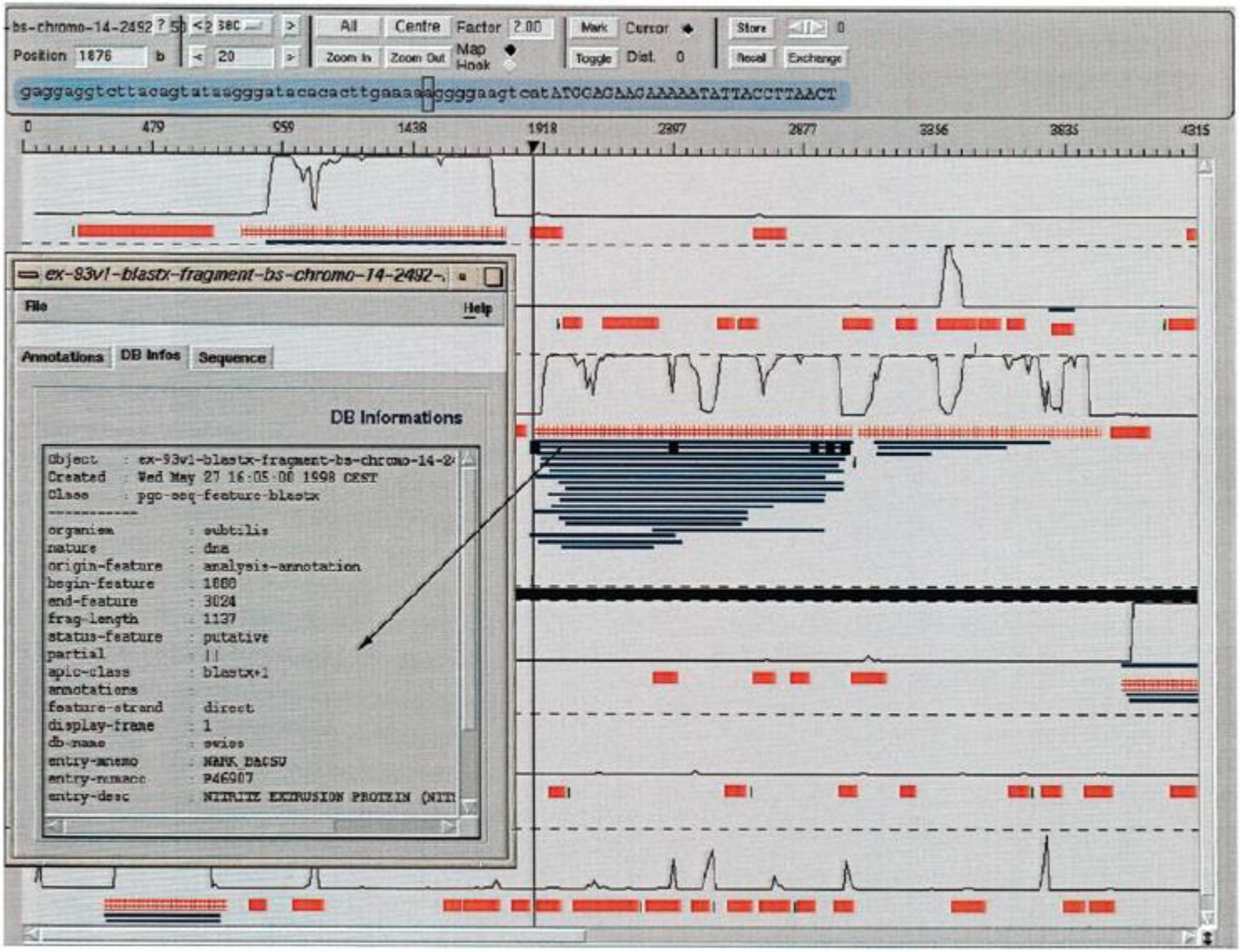
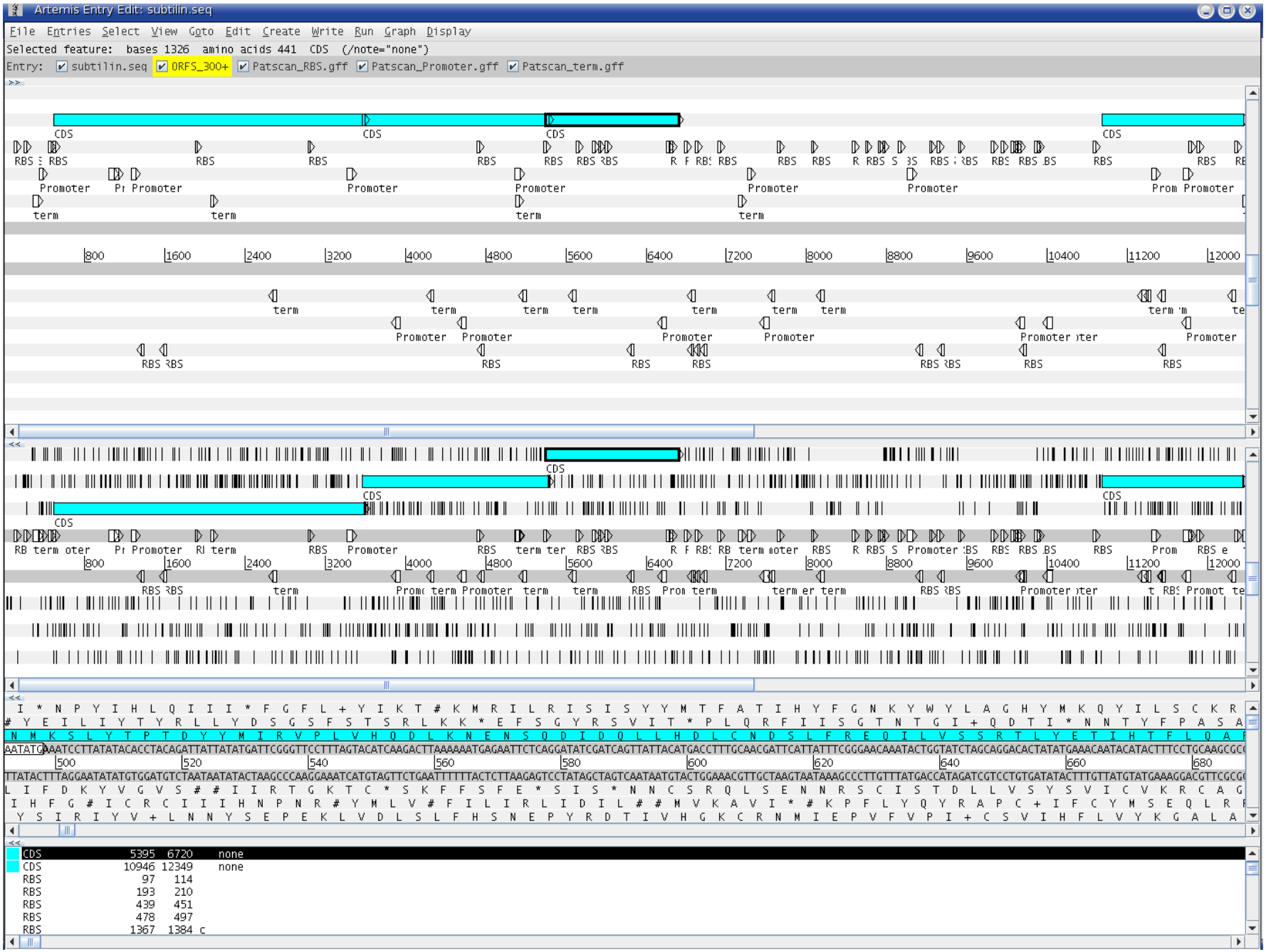



**Fig. 6.** Superimposition of results from three different strategies in the Imagene Result Manager (Cartographic Interface). Results obta ined with the CDS searching strategy are shown in red boxes (CDSs) and green triangles (RBSs), those obtained with the Blastx strategy ar e shown in blue rectangles and, finally, the GeneMark© coding predictions are displayed as black curves. The results are given in each of the six frames.


Artemis Entry Edit: subtilin.seq
File Entries Select View Goto Edit Create Write Run Graph Display
Selected feature: bases 1326 amino acids 441 CDS (/note="none")
Entry: subtilin.seq ORFS_300+ Patscan_RBS.gff Patscan_Promoter.gff Patscan_term.gff
CDS
RBS
Promoter
term
800 1600 2400 3200 4000 4800 5600 6400 7200 8000 8800 9600 10400 11200 12000
500 520 540 560 580 600 620 640 660 680
CDS 5395 6720 none
CDS 10946 12349 none
RBS 97 114
RBS 193 210
RBS 439 451
RBS 478 497
RBS 1367 1384 c

# Modèles de gènes : génomes procaryotes

## Modèle de Markov caché (HMM : Hidden Markov Model)

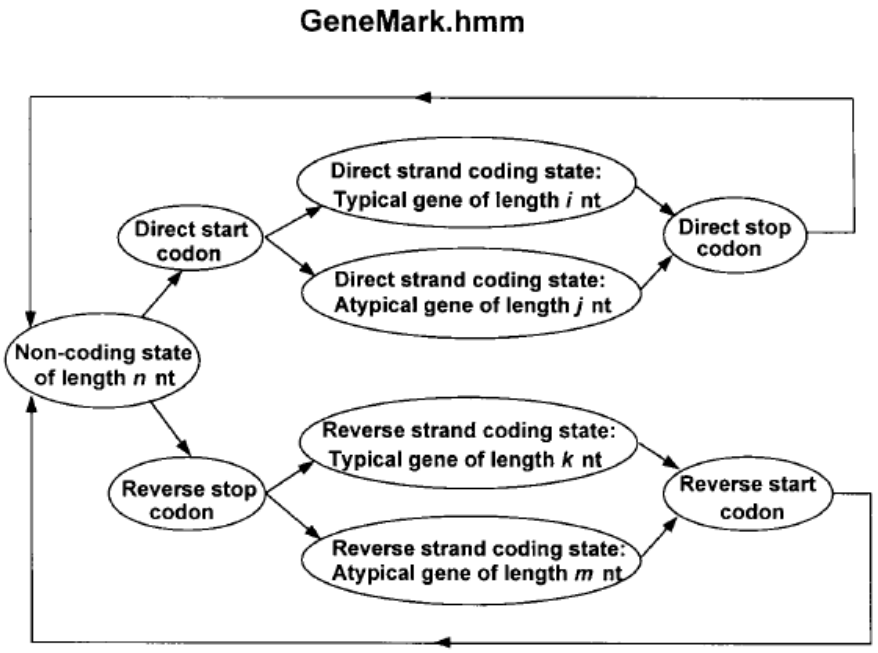



**Figure 1.** Hidden Markov model of a prokaryotic nucleotide sequence used in the GeneMark.hmm algorithm. The hidden states of the model are represented as ovals in the figure, and arrows correspond to allowed transitions between the states.

(extrait de Nucleic Acids Res. (1998), 26, 1107-1115)

# Modèles de gènes : génomes procaryotes

## Graphe orienté sans circuit (DAG : Directed Acyclic Graph)

**FrameD** : Nucleic Acids Res., (2003), 31, 3738-41

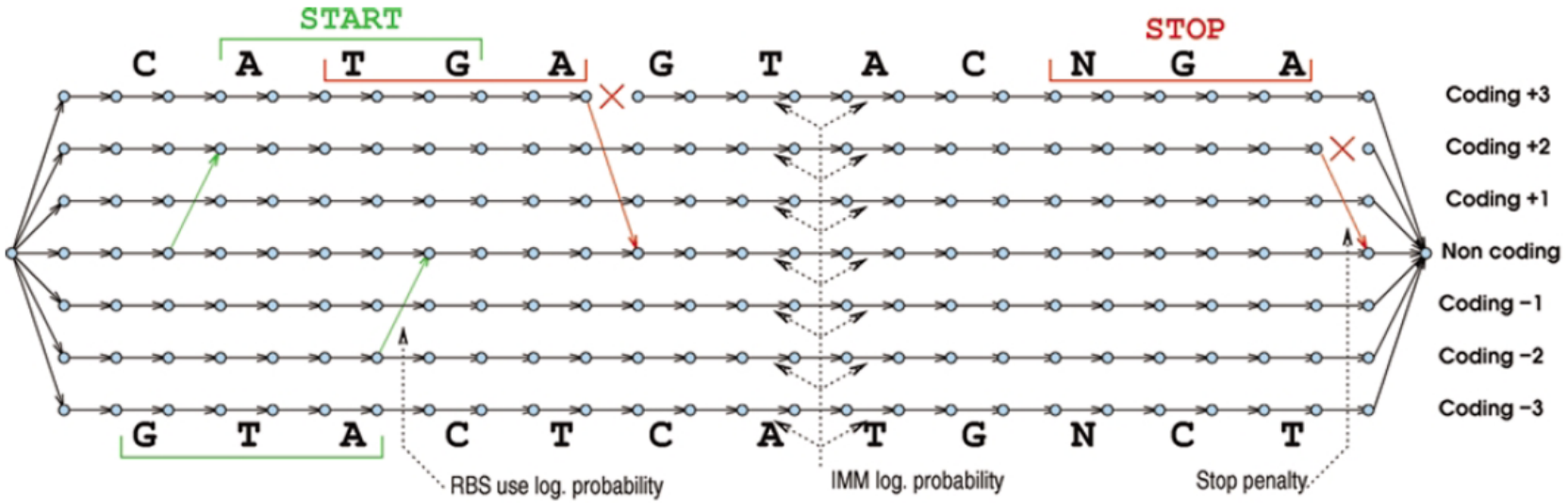
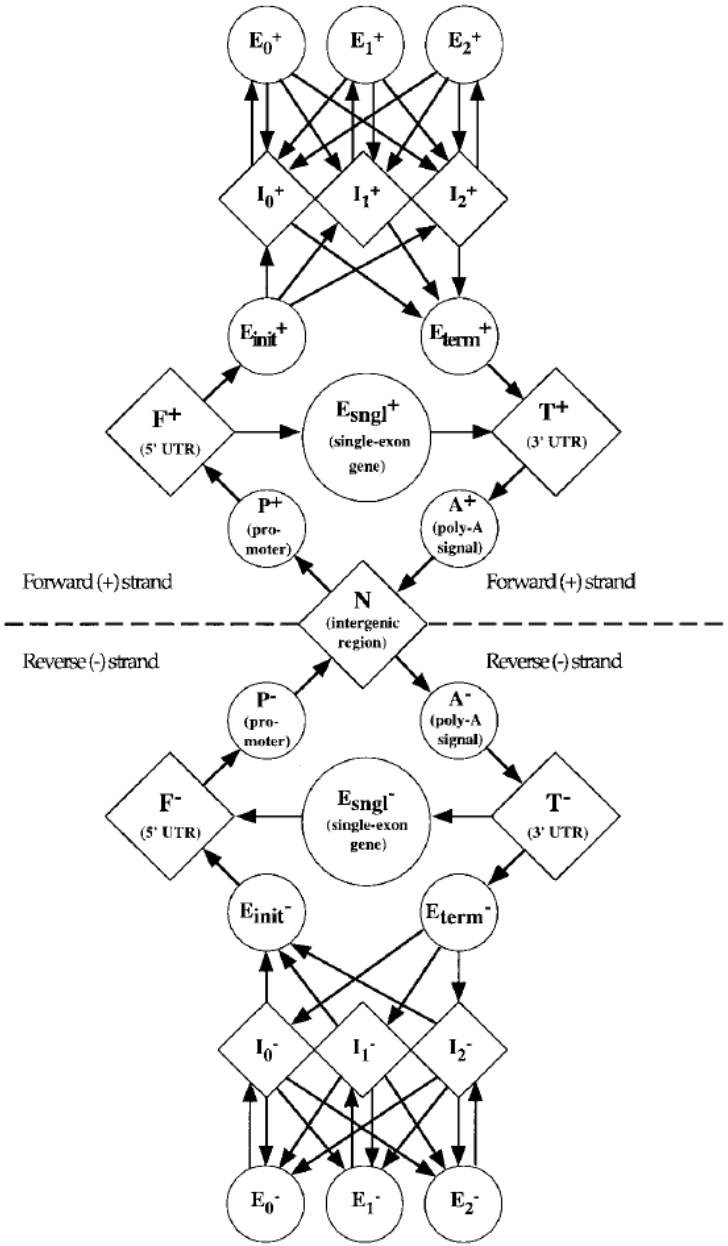



**Figure 1.** A simplified view of the directed acyclic graph built for analyzing the sequence CATGAGTACNGA. This view ignores the additional complexity induced by gene overlapping regions and frameshift modeling. The occurrence of a START codon at position 2 to 4 induces a 'signal' edge that goes from the non-coding track to the +2 coding track. Similarly, the occurrence of the NGA codon at the end induces a STOP signal edge. Edge weights sources are indicated using dotted arrows.

# Modèles de gènes : génomes eucaryotes




## Modèle de Markov caché (HMM : Hidden Markov Model)

Exemple du modèle de GENSCAN
(extrait de J. Mol. Biol. (1997) 268, 78-94)

# Distribution des longueurs (extrait de J. Mol. Biol. (1997) 268, 78-94)

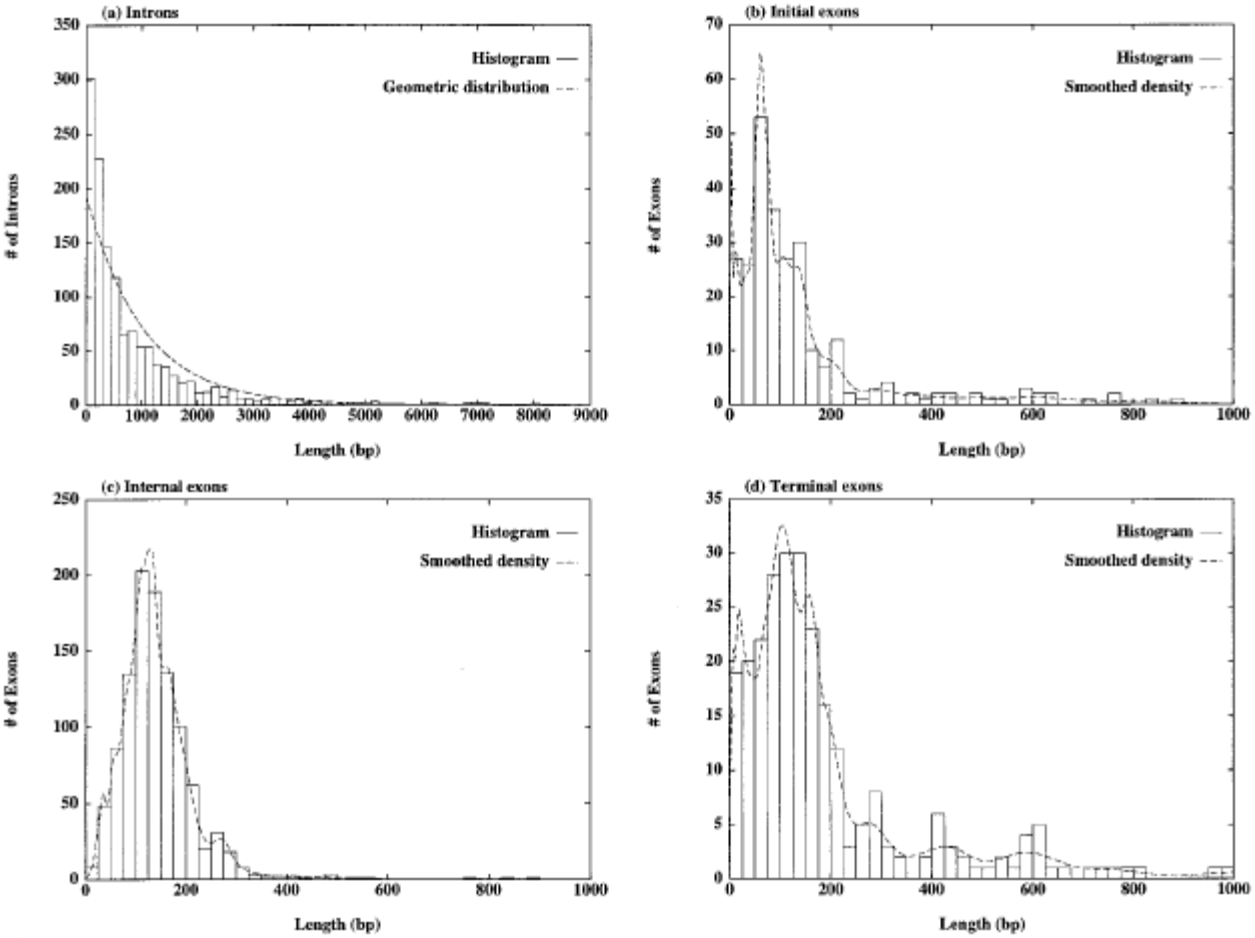
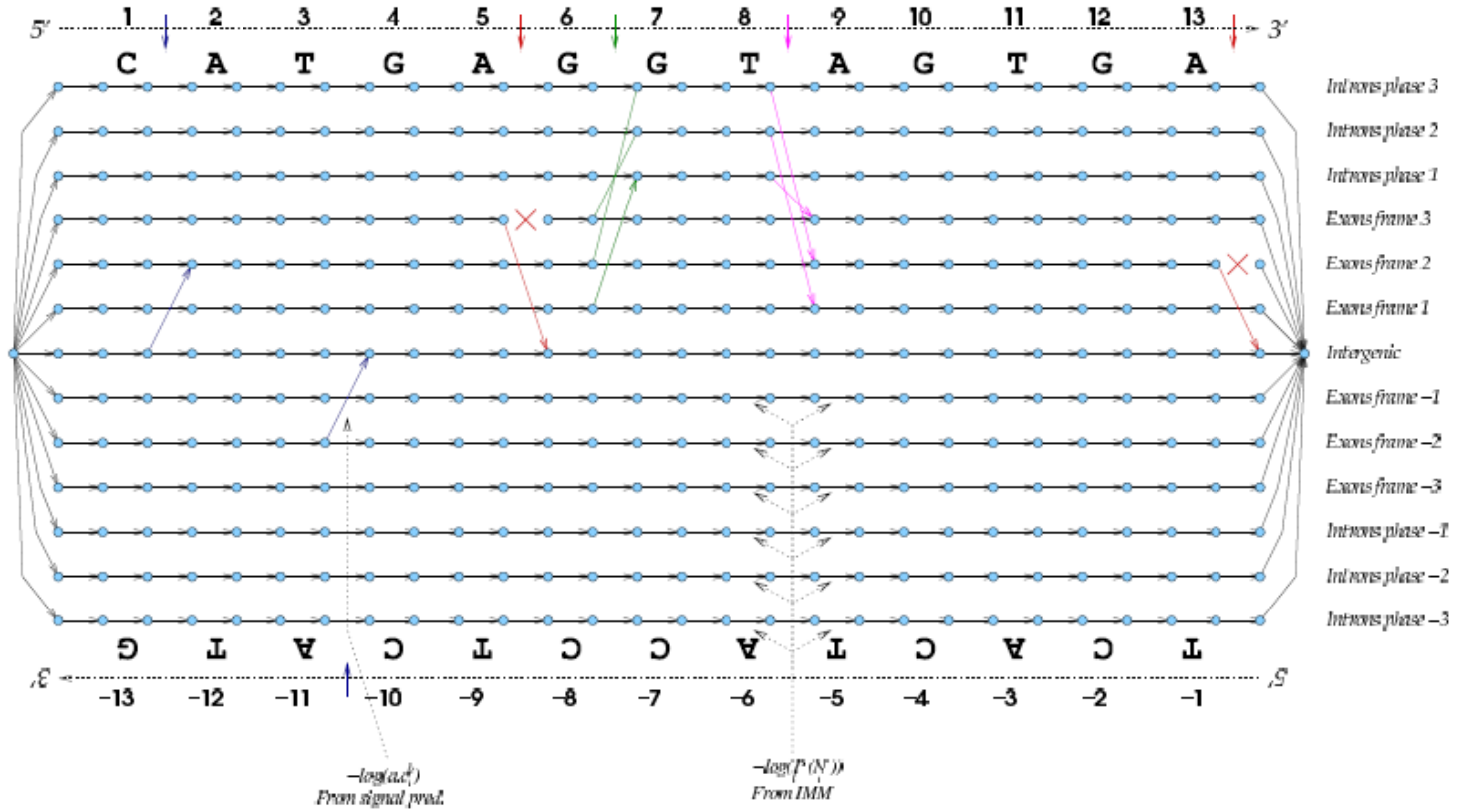



**Figure 4.** Length distributions are shown for (a) 1254 introns; (b) 238 initial exons; (c) 1151 internal exons; and (d) 238 terminal exons from the 238 multi-exon genes of the learning set $\mathscr{L}$. Histograms (continuous lines) were derived with a bin size of 300 bp in (a), and 25 bp in (b), (c), (d). The broken line in (a) shows a geometric (exponential) distribution with parameters derived from the mean of the intron lengths; broken lines in (b), (c) and (d) are the smoothed empirical distributions of exon lengths used by GENSCAN (details given by Burge, 1997). Note different horizontal and vertical scales are used in (a), (b), (c), (d) and that multimodality in (b) and (d) may, in part, reflect relatively small sample sizes.

# Modèles de gènes : génomes eucaryotes

## Graphe orienté sans circuit (DAG : Directed Acyclic Graph)

**EuGène** : LNCS, 2066, 118-133. Springer Verlag



# Information de type similarité

Information externe à la séquence elle-même (de type extrinsèque) contrairement au contenu statistique ou aux signaux qui sont internes à la séquence (de type intrinsèque)

Comparer la séquence à analyser avec des séquences connues peut permettre de refléter la présence de gènes et donner des informations sur leur structure. Notamment, la structure en exons/introns pour les gènes eucaryotes.

Types de séquences utilisées pour la comparaison :

- les ADNc
- les EST
- les protéines
- des séquences génomiques

# Information de type similarité

Méthodes prédisant la structure en exons-introns par alignement de la séquence génomique soit avec un ARNm (ou un ADNc), soit avec une protéine. Parmi les plus utilisés, on trouve :

| Méthode | Séquence de référence | Référence |
|---|---|---|
| BLAT | ARNm ou protéine | Genome Research 12(4):656-64 (2002). |
| sim4 | ARNm | Genome Research 8:967-74 (1998). |
| Scipio | ARNm ou protéine | BMC Bioinformatics 9:278 (2008) |
| GeneWise | protéine | Genome Research 14(5):988-95 (2004) |
| GenomeWise | ADNc, EST | Genome Research 14(5):988-95 (2004) |
| WebScipio | protéine | Bioinformatics 12:270 (2011) |

L'extension du logiciel WebScipio permet de rechercher une forme spécifique d'épissage alternatif (exons mutuellement exclusifs)

# Prise en compte de la similarité

Avec des séquences protéiques : GENOMESCAN (intégration de cette information dans le modèle GENSCAN.

Combine le résultats d'une méthode *ab initio* de recherche de la structure en exons-introns des gènes (GenScan) avec une recherche par similarité avec les protéines connues (BlastX)

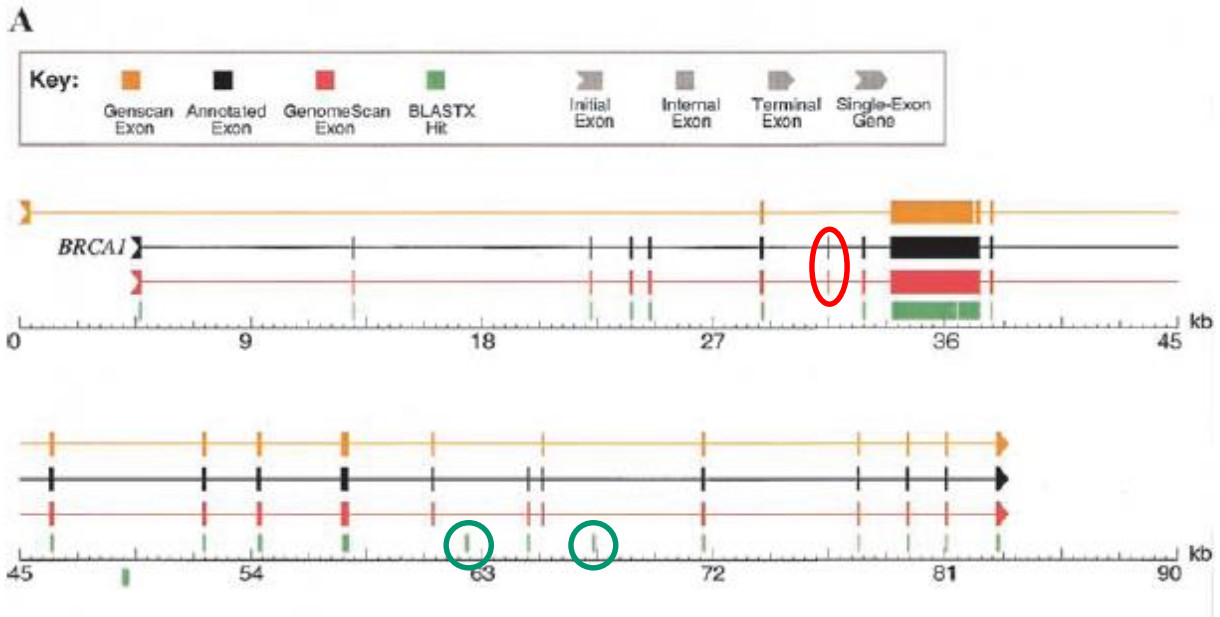
Différence par rapport à GenScan (méthode utilisant un HMM) : la probabilité du chemin est renforcée si la région prédite comme exon présente un hit avec BlastX et est diminuée dans le cas contraire

- Les hits de BlastX qui tombent en N-ter ou C-ter de la protéine sont utilisés pour mieux prédire les codons initiateur et terminateur
- Deux hits BlastX ajacents sur la protéine mais séparés par une distance $\geq$ 60 pb dans la séquence génomique sont utilisés pour prédire les régions introniques putatives
- si plusieurs hits BlastX chevauchants (information redondante), les résultats sont pré-traités pour ne garder que le hit le plus significatif.

# Prise en compte de la similarité

<u>Avec des séquences protéiques</u> : GENOMESCAN (intégration de cette information dans le modèle GENSCAN.

Exemple d'un résultat de prédiction (extrait de Genome Research (2001), 11, 803-816)




Hit BlastX faux-positifs -> pas de prédiction d'exon

Exon 7 : prédit par GenomeScan alors que pas prédit par GenScan et pas de hit BlastX -> pas seulement addition des deux résultats mais fait également sa propre inférence. Utilise le fait qu'il y a une incompatibilité de phase de l'intron en les exons 6 et 8 -> nécessite la présence d'un autre exon entre les deux.

# Prise en compte de la similarité

Avec des séquences génomiques : TWINSCAN (Bioinformatics (2001), 17 suppl. 1, S140-S148, intégration de cette information dans le modèle GENSCAN.

Exploite la similarité entre deux génomes apparentés : ceci permet de détecter les régions codantes et les éléments de régulation.

target sequence : séquence à annoter
informant sequence : séquence génomique apparentées

Les répétitions sont masquées. Alignement des deux séquences avec le programme Wu-BlastN

## Codage de la conservation

La conservation de la séquence est traduite en symboles sous la séquence ADN :
| = match
: = mismatch
. = non aligné
Les gaps dans la séquence « target » sont ignorés et ceux dans la séquence « informant » sont considérés comme des mismatches.

```
                          10         20         30
position dans la séquence target  123456789|123456789|123456789|123456789
séquence target                   ATTTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC
séquence conservation             ||:|||.........|:|:||||||||||:||:|||::||
```

**Fig. 1.** An example DNA sequence together with the corresponding conservation sequence.

# Prise en compte de la similarité

## Calcul de la probabilité

Quand on recherche le chemin le plus probable dans le HMM, calcul de la probabilité que la séquence soit dans l'état *i* (par exemple : exons, UTR etc.)

$$P_e(i) = \text{Proba\_GeneSan}(i) * \text{Proba\_Conservation}(i)$$

## Calcul de la probabilité de conservation

Modèle de Markov d'ordre 5
Estimée par état (codant, UTR, non codant…) à partir de séquences annotées formant l'ensemble d'apprentissage.

# Prise en compte de la similarité

Avec des séquences génomiques : TWINSCAN (intégration de cette information dans le modèle GENSCAN.

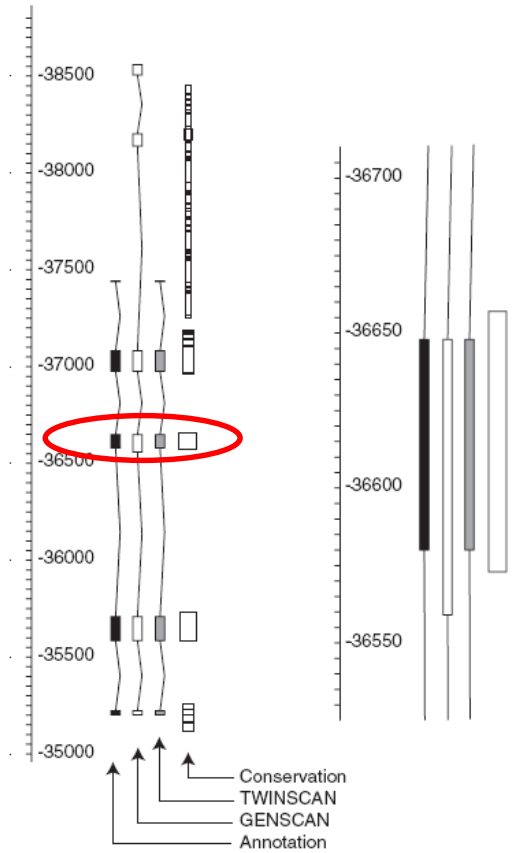


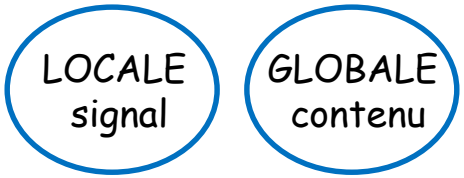

**Fig. 5.** Detailed view of the annotation, gene predictions and conservation at the L44L gene (AAB47245.1) from the *Mus musculus* Bruton's tyrosine kinase locus (U58105.1). The magnification at right shows the region around exon 3. The width of boxes representing BLAST alignments corresponds to the quality of the alignment. The image comes from ACEDB.

Exemple d'un résultat de prédiction
(extrait de Bioinformatics (2001), 17 suppl. 1, S140-S148)

GenScan ne prédit pas le premier exon et en prédit deux supplémentaires.
TwinScan prédit correctement le premier exon car le niveau de conservation de séquence à son niveau augmente la probabilité totale de l'exon.

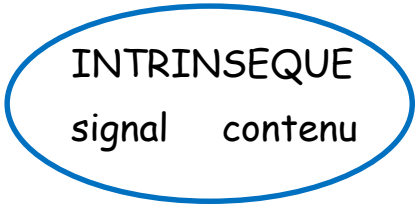
Exon 3 : mauvais site d'épissage en 5' choisi par GenScan.
Bonne prédiction de TwinScan car la conservation de séquence s'arrête avant la position de fin de l'exon prédit par GenScan

# Evolution de l'intégration des sources d'informations
(extrait de la thèse de Sylvain Foissac, 2004)



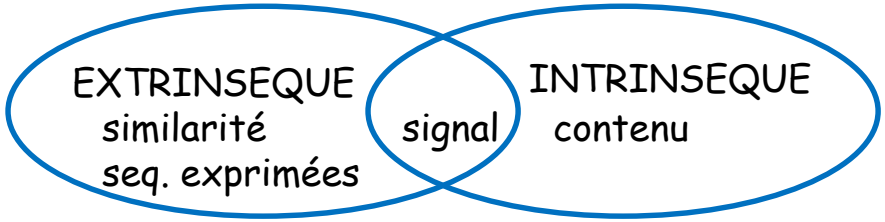

Deux sources traitées par des méthodes indépendantes (ex : Staden, 1984; Gelfand, 1990)

---



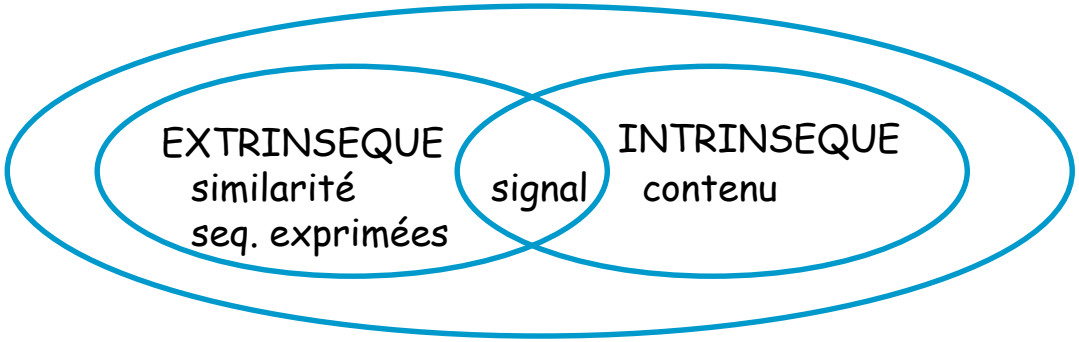

Intégration de ces deux sources dans un même logiciel (ex : Guigo et al., 1992, logiciel GENSCAN (Burge et Karlin, 1997)

---

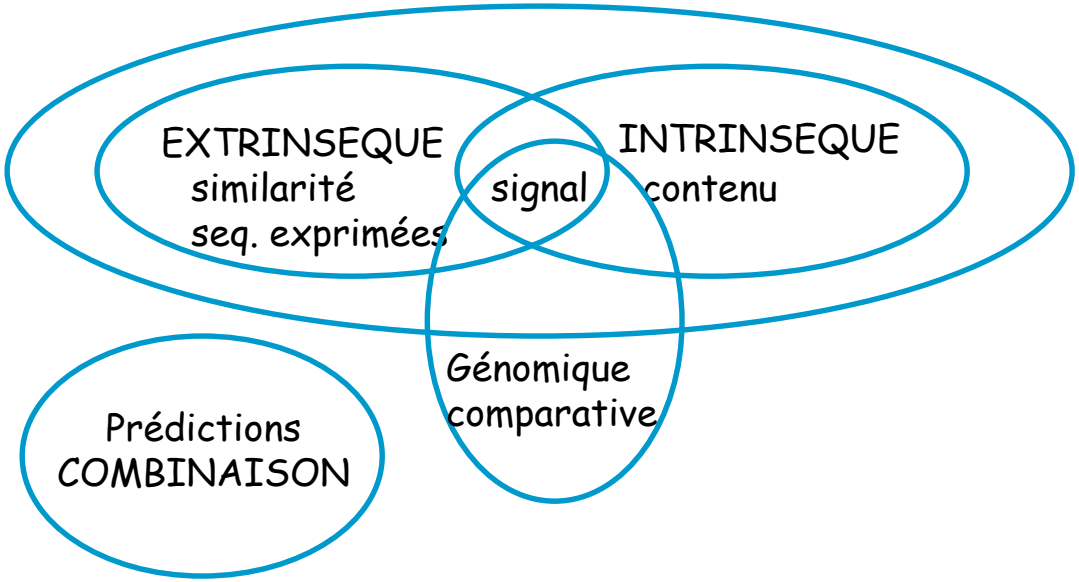



Augmentation des données d'expression, prise en compte de la similarité de séquences (Borodovsky et al., 1994, Fickett, 1995)

---

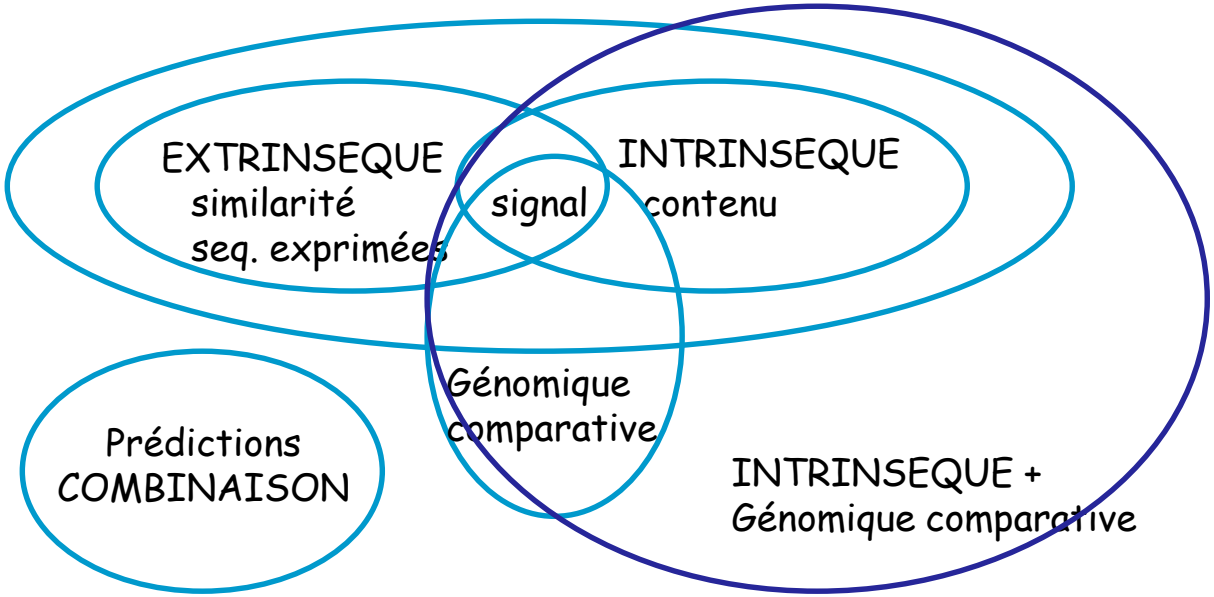



Intégration de ces deux types d'information dans de nombreux logiciels. GENOMESCAN (Yeh et al., 2001) résulte de l'intégration de similarité protéique dans GENSCAN




Conservation entre séquences génomiques exploitées en combinaison seulement avec des informations de type signal

---




Génomique comparative intégrée uniquement dans des méthodes intrinsèque. Par exemple, TWINSCAN (Korf et al., 2001) intègre la génomique comparative dans GENSCAN.

# Environnements intégrés pour l'annotation des génomes procaryotes

✓ **RAST server** (BMC Genomics. 2008 Feb 8;9:75. doi: 10.1186/1471-2164-9-75)

DESCRIPTION:

We describe a fully automated service for annotating bacterial and archaeal genomes. The service identifies protein-encoding, rRNA and tRNA genes, assigns functions to the genes, predicts which subsystems are represented in the genome, uses this information to reconstruct the metabolic network and makes the output easily downloadable for the user

✓ **Prokka** (Bioinformatics. 2014, 30:2068-69)

DESCRIPTION:

Prokka, a command line software tool to fully annotate a draft bacterial genome in about 10min on a typical desktop computer. It produces standards-compliant output files for further analysis or viewing in genome browsers.

| Tool (reference) | Features predicted |
|---|---|
| Prodigal (Hyatt 2010) | Coding sequence (CDS) |
| RNAmmer (Lagesen *et al.*, 2007) | Ribosomal RNA genes (rRNA) |
| Aragorn (Laslett and Canback, 2004) | Transfer RNA genes |
| SignalP (Petersen *et al.*, 2011) | Signal leader peptides |
| Infernal (Kolbe and Eddy, 2011) | Non-coding RNA |

Méthodes utilisées dans Prokka (issu de Bioinformatics. 2014, 30:2068-69)

# Environnements intégrés pour l'annotation des génomes procaryotes

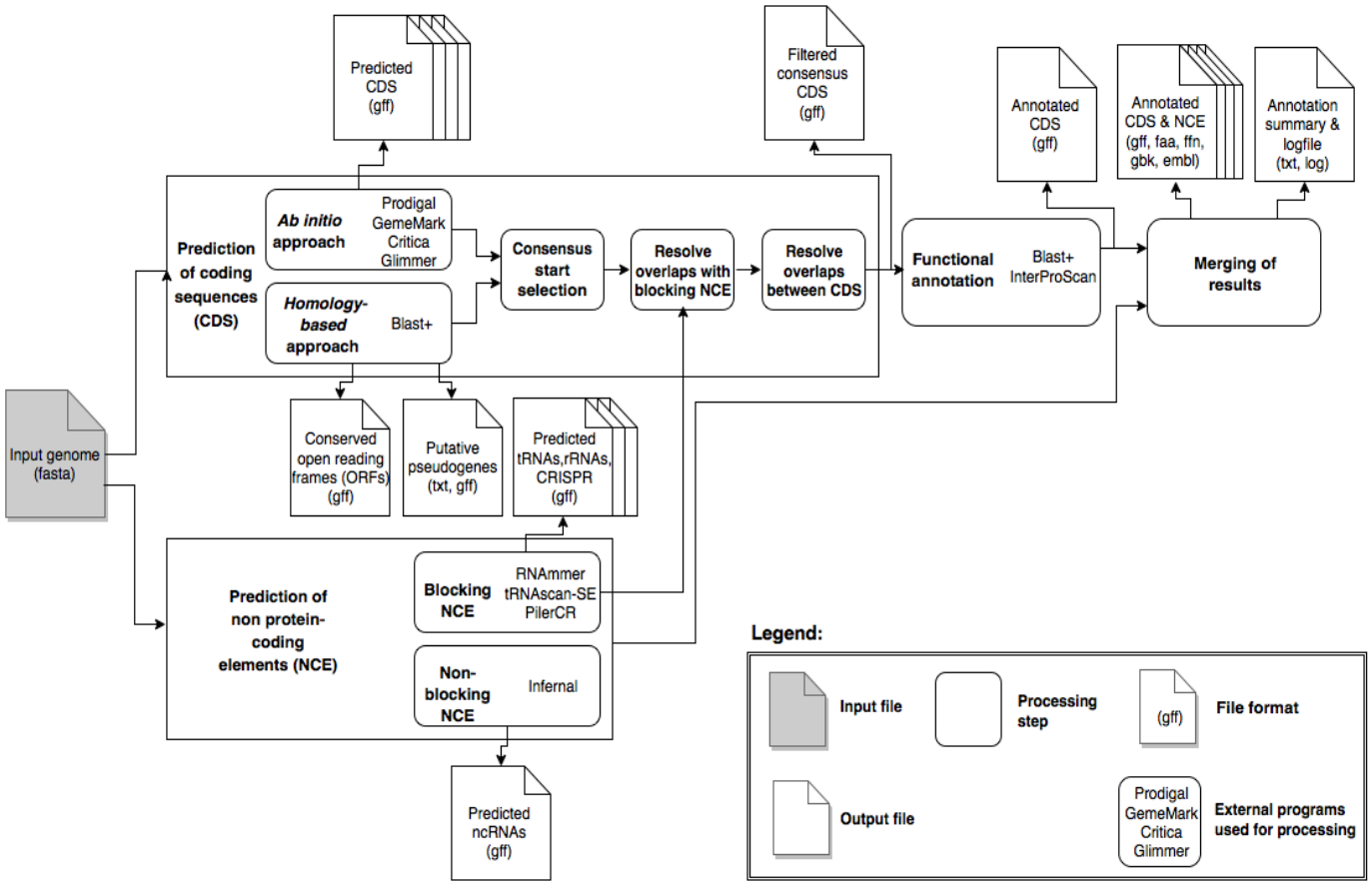
✓ **ConsPred** (Bioinformatics. 2016, 32:3327-29)

DESCRIPTION:

We present ConsPred, a prokaryotic genome annotation framework that performs intrinsic gene predictions, homology searches, predictions of non-coding genes as well as CRISPR repeats and integrates all evidence into a consensus annotation. ConsPred achieves comprehensive, high-quality annotations based on rules and priorities, similar to decision-making in manual curation and avoids conflicting predictions. Parameters controlling the annotation process are configurable by the user.

# Environnements intégrés pour l'annotation des génomes procaryotes

## Exemple de ConsPred (Weinmaier *et al.*, 2016, Bioinformatics, 32:3327-29)




(Extrait de Weinmaier *et al.*)

**Figure S1**. ConsPred workflow

Coding sequences (CDS) are predicted by combining different *ab initio* gene predictions, and conserved open reading frames (ORFs) detected by homology search against the NCBI nr database. Database entries from closely related taxa are excluded to prevent possible misannotations due to low phylogenetic distance. Putative pseudogenes are exported for user inspection. From all predicted non-protein-coding elements (NCE) those that biologically must not overlap with CDS are considered blocking NCE. CDS overlapping with blocking NCE are removed. Filtered consensus CDS are obtained from predicted CDS and conserved ORFs by using predefined weights and rules and subsequent removal of CDS that overlap with blocking NCEs. Filtered consensus CDS are functionally annotated and then merged with the NCE into the final annotation files.