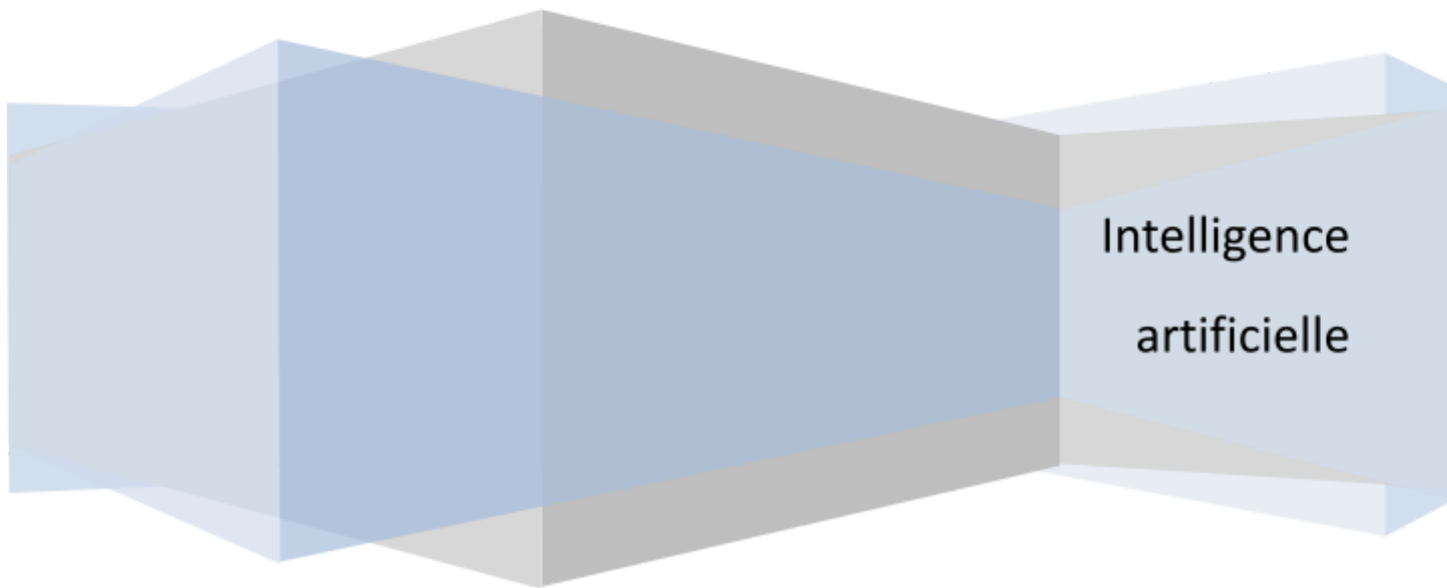


Modèle de Markov caché

Atelier d'application aux séquences biologiques
et discussion

Par Nadine St-Amand



ISBN 978-0-557-02612-8

© 2008 Nadine St-Amand

Montréal, Québec, Canada

Publié et distribué par Lulu



1. Introduction

1.1 Contexte du projet

1.1.1 Contexte bioinformatique de la discussion

En bioinformatique, on étudie, classifie, les séquences biologiques. En effet, l'information biologique contenue dans les séquences moléculaires telles que l'ADN (Acide DésoxyriboNucléique), les ARN (Acide RiboNucléique) et les séquences protéiques prédit en quelque sorte les recettes de fabrication de toutes les molécules du corps vivant.

Lors du processus de l'évolution certains segments des séquences d'ADN subissent des pressions sélectives variables.

La communauté est intéressée à classifier les séquences biologiques: soit à identifier les séquences qui sont évolutivement parentes en utilisant des analyses de similarités entre les séquences. Avec ces résultats, on trace des arbres phylogénétiques entre les molécules de différentes espèces ou encore on étudie les ressemblances entre différentes molécules afin d'en tirer des conclusions sur leurs fonctionnalités.

→ Ceci demeure un problème ouvert.

La communauté est également intéressée à mieux comprendre les relations existant entre la séquence primaire d'une molécule biologique et sa fonction dans le corps. Des relations entre séquences primaires et interprétations biologiques sont difficiles à établir à cause du nombre important de paramètres en jeu : molécules régulatrices, régions excisées, interactions hydrophobes, interactions polaires, etc.

→ Ceci demeure également un problème ouvert.

De nos jours, les modèles de Markov cachés servent à étudier plusieurs variantes de ces problèmes:

Recherche de gènes dans une base de données
(« Genefinding »)

Recherche de profils (« profile searches »)

Alignement de séquences similaires (« multiple
sequence alignment » ou MSA)

Identification de site de régulation

- Une solution : Les modèles de Markov cachés (HMM)

"Hidden Markov models (HMMs) are a formal foundation for making probabilistic models of linear sequence 'labeling' problems." (5)

Il a été identifié par les spécialistes de la bioinformatique, que les séquences biologiques formaient un problème idéalement modélisé par les modèles de Markov cachés, d'une manière similaire à celle des problèmes de reconnaissance vocale. Il affirme même:

"HMMs are the Legos of computational sequence analysis" (5)

Ce qui annonce le rôle central que les chaînes de Markov sont appelées à jouer dans les sciences biologiques au cours des années à venir.

1.1.2 Contexte biologique du projet

Voici quelques mots de vocabulaire qui seront utilisés au cours de la définition du problème. Nous définissons ici les processus principaux affectant les séquences biologiques.

- On dit d'un gène qu'il est **EXPRIMÉ** s'il code pour une protéine, si une telle molécule est instanciée dans le corps. On peut caractériser l'expression en disant qu'un gène est fortement exprimé si beaucoup de copies de la molécule cible sont fabriquées.
- La **TRANSCRIPTION** est le processus qui copie un segment d'ADN en un ARN en tous points complémentaire. (Nous définirons formellement ce qu'est la complémentarité)
- La **TRADUCTION** est le processus qui traduit chaque ARN en protéine selon le code génétique. Avant ce dernier processus, lors de la maturation de l'ARN, certains segments sont tout simplement omis : nous en reparlerons.
- Les protéines sont soit utilisées directement par le corps ou servent d'enzymes ou de régulateur pour d'autres opérations de synthèse

La composition des séquences biologiques.

Les différentes séquences biologiques d'intérêt sont pour nous l'ADN et les protéines. Dans l'ADN, on caractérise les régions, selon leur rôle dans la finalité de la construction de la molécule-cible.

- On parle de régions **CODANTES** et de régions **NON-CODANTES** de l'ADN. Les régions non-codantes sont aussi appelées régions **INTER-GÉNIQUES**.
- Chaque région codante est précédée, à une distance connue, d'un site de liaison pour les ribosomes (complexes moléculaires qui vont traduire l'ARN messager en protéine). Ceci s'abrévie **RBS** pour Ribosome Binding Site.
- Une région codante est totalement transcrite en ARN, mais cet ARN sera redécoupé avant d'être traduit en protéines. Ce découpage s'appelle l'**ÉPISSAGE** et les zones incluses se nomment **EXONS** alors que les zones exclues se nomment **INTRONS**.

Cette information biologique est de la plus grande importance dans les modèles markoviens, tel que nous le verrons plus tard.

Si nous utilisons notre loupe et descendons à un niveau de détail plus fin, informellement, nous dirons que la séquence biologique se compare à une séquence de lettres ou de sons, et par conséquent, se modéliserait très bien telle quelle avec une machine d'états telle qu'utilisée dans les analyseurs syntaxiques (parseurs).

- Dans l'ADN, on a un polymère où chaque position est occupée par un des **4 NUCLÉOTIDES** possibles: A-C-G-T (adénine, cytosine, guanine et thymine). Ceux-ci deviendront éventuellement nos observations.
- Dans l'ARN, on a un polymère où chaque position est occupée par un des 4 nucléotides possibles: A-C-G-U (adénine, cytosine, guanine et uracile)
- La **COMPLÉMENTARITÉ** est définie par le fait que tout C se voit associé à un G alors que tout A se voit associé à un U (T dans l'ADN). Des chaînes complémentaires sont comme des copies carbonées de leur chaîne hôte. La complémentarité existe entre séquences ADN-ADN et ADN-ARN (Voir l'annexe 3).
- À partir du point de départ de la traduction, chaque séquence de 3 nucléotides se traduit en un acide aminé selon le **CODE GÉNÉTIQUE** (en annexe). Puis la séquence des 3 nucléotides suivants, qui lui est juxtaposée, est également traduite et forme l'acide aminé suivant, etc.

- Dans la protéine, chaque position est occupée par un des **20 ACIDES AMINÉS** possibles¹. Ceci s'appelle une chaîne peptidique.

Une autre information biologique d'importance est la conformation tridimensionnelle résultante de la séquence des éléments constitutants.

- La séquence d'acides aminés dans une protéine constitue sa **STRUCTURE PRIMAIRE**
- La chaîne protéique se conforme à 2 **STRUCTURES SECONDAIRES** typiques : l'hélice dite **HÉLICE ALPHA** et le feuillet dit **FEUILLET BÊTA**. Ces structures peuvent être prédites à partir de la séquence.
- La **STRUCTURE TERTIAIRE** des protéines est la conformation 3D résultante d'interactions biochimiques telles que les interactions hydrophiles, hydrophobes et polaires principalement, entre les chaînes libres et les structures secondaires.
- L'ARN messager arbore également quelques structures secondaires. Cette structure secondaire résulte, dans le cas de l'ARN, de liaisons avec les nucléotides complémentaires qui se connectent avec une séquence inversée, comme une image miroir.
- Même l'ADN, par-dessus sa conformation en double hélice, forme une structure tridimensionnelle (la chromatine) qui influence l'expression de ses gènes : si l'ADN est trop condensé et enroulé, les gènes qui sont enfouis ne seront tout simplement pas exprimés.

Figure 1. L'hémoglobine est un exemple de molécule protéique comportant beaucoup d'hélices alpha

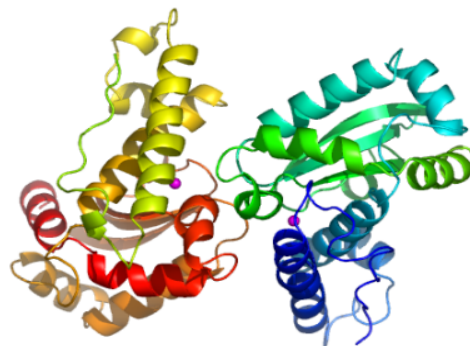
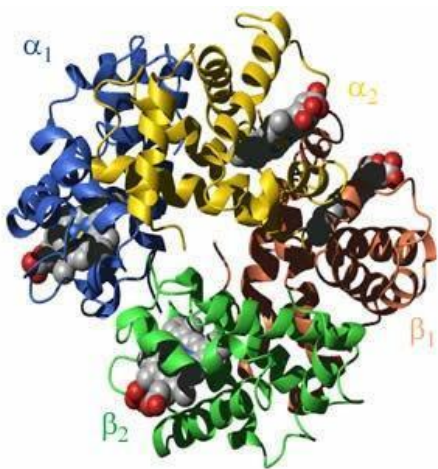


Figure 2. Enzyme intervenant dans la réplication de l'ADN
http://dbpedia.org/page/DNA_clamp

<http://cathwww.biochem.ucl.ac.uk/latest/class.html> & <http://www.bioinf2.leeds.ac.uk/msc2004/cathscop.gif>

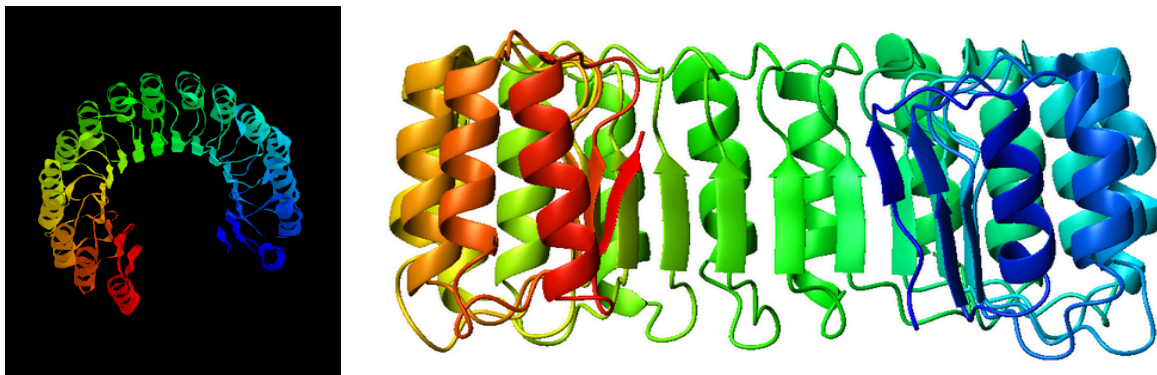
¹ Possible dans le sens que l'ADN code pour 20 acides aminés différents, mais il en existe d'autres.

N.B. : Ne pas confondre avec la dénomination des sous-unités de l'hémoglobine : ces chaînes alpha (1 et 2) et bêta (1 et 2) ne se rapportent pas à la structure secondaire mais à l'énumération des sous-unités peptidiques. Ceci prête à confusion mais ces noms sont consacrés.

Les structures secondaires sont un sujet d'étude si important qu'on a mis en place certaines classifications des structures protéiques telles que CATH (<http://cathwww.biochem.ucl.ac.uk/latest/class.html>) et SCOP (<http://www.bioinf2.leeds.ac.uk/msc2004/cathscop.gif>).

Ces classifications tiennent compte du fait que les structures secondaires peuvent être combinées en des structures plus complexes. Par exemple, la structure de cette molécule répète les hélices alpha et les feuillets bêta à l'extrême.

Figure 3. Inhibiteur de ribonucléase placentaire



Source de l'illustration: http://en.wikipedia.org/wiki/Ribonuclease_inhibitor

La prédiction des structures secondaires à partir de la séquence demeure un problème ouvert. Cependant, ce qui intéresse vraiment les scientifiques, c'est la structure tertiaire qui est la vraie conformation 3D formée de cette composition de structures secondaires. C'est la structure tertiaire d'une protéine qui lui dicte vraiment sa fonction biologique.

Certaines protéines servent de briques de construction car elles sont solides et uniformes, d'autres protéines sont plus complexes et leur conformation tridimensionnelle joue un rôle de clé-serrure dans des interactions de régulations des plus sensibles et délicates, avec des portées variables sur le métabolisme humain.

L'évolution des séquences biologiques.

Différentes séquences biologiques subiront différentes pressions sélectives. D'une manière générale, il est vrai de dire qu'une séquence d'ADN ne sera exposée à la pression sélective que si elle est exprimée, donc exposée aux transcriptases et transformée en ARN messager puis en protéine.

Le but de ce travail n'est pas de recenser toutes les relations possibles entre la nature d'une séquence, sa fonction et/ou son évolution mais de se donner des outils afin de démontrer informellement qu'une telle relation existe en créant un modèle capable de détecter ou interpréter une séquence et ses variantes.

Ceci nous sous tendra le cadre théorique et expliquera notre besoin d'un modèle qui exprime les probabilités de certains événements. Voici donc quelques éléments concrets qui pourront être éventuellement incorporés dans la modélisation :

- Nous savons que certaines mutations de l'ADN seront plus probablement conservées que d'autres.
- Nous savons que certaines suites de nucléotides codent pour le même acide aminé et sont donc fonctionnellement équivalentes.
- Nous savons que certains acides aminés se « ressemblent » dans leur fonction alors que d'autres sont très différents. Des groupes ou familles d'acides aminés ont été identifiés dans la littérature (voir annexe).
- Il existe des caractéristiques statistiques affectant les distributions d'acides aminés selon les types de protéines.
- Il existe des caractéristiques statistiques affectant la distribution des nucléotides selon l'espèce ; par exemple, le rapport GC/AT permet d'identifier des espèces de bactéries. Cette différence est due à un biais mutationnel des ADN polymérases des différentes espèces.
- Il existe de même des biais dans l'utilisation des codons synonymes (équivalents) : une espèce préfère l'un alors qu'une autre espèce préfère l'autre. Ceci peut être caractérisé statistiquement.

Nous savons aussi que certaines régions de l'ADN résultent en des chaînes protéiques et d'autres non, en vertu de la présence du codon stop ou du phénomène de l'épissage lors de la maturation de l'ARN pré-messager avant sa traduction. D'autres régions ne sont tout simplement pas exposées à la transcription à cause de facteurs épigénétiques ou de complexes protéiques qui en masquent l'accès. Pour l'instant, nous tiendrons compte d'un modèle simple : dans la séquence de nucléotides qui composent un gène eucaryote, nous distinguons 3 types d'états : une région codante conservée lors de la maturation (l'exon), une région non codante excisée lors de l'épissage (l'intron) et le site d'épissage 5' (5' splice site) marquant la frontière entre l'exon et l'intron.

1.2 Définition explicite de la problématique

1.2.1 Objectif à réaliser dans cet atelier

Mon premier objectif sera de préparer le HMM-jouet qui est expliqué dans l'article "What is a hidden Markov Model" (5). Dans cet article, on explique un modèle simple dénommé "A toy HMM for 5' splice site recognition" (5).

Figure 6. HMM jouet pour la reconnaissance de site de coupure dans l'ADN. Tiré de Eddy (5)

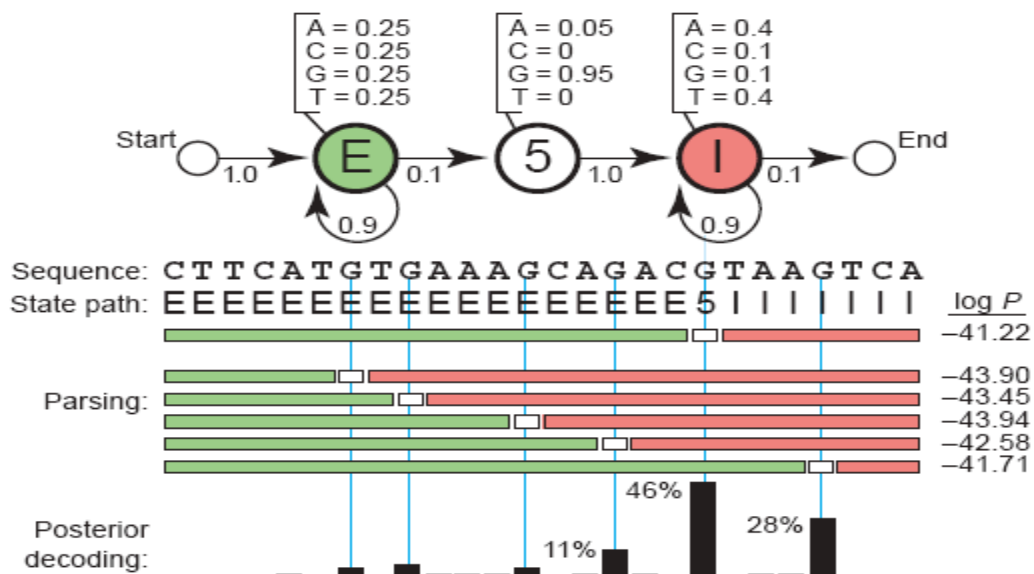


Figure 1 A toy HMM for 5' splice site recognition. See text for explanation.

Dans ce modèle, on pose l'hypothèse que les régions conservées, les exons, ont une distribution équiprobable des 4 bases de nucléotides, donc 0,25 de probabilité pour chacune dans l'état EXON étiqueté « E » sur la figure. On pose également l'hypothèse que les régions excisées, les introns, sont plus riches en A et T qu'en C et G, à cause des biais de composition propres aux régions introniques. Les probabilités sont donc de 0,4 pour A et 0,4 pour T (et 0,1 pour C et G) dans la région des INTRONS étiquetée « I » sur la figure. Enfin, on pose la dernière hypothèse que le site d'épissage 5' (site donneur à la jonction exon-intron) est très majoritairement occupé par une guanine (G) ou, plus rarement, par une adénine (A) : la probabilité d'émission est donc de 0,95 pour G et 0,05 pour A dans l'état du site d'épissage 5' étiqueté « 5 » (ou « S ») sur la figure.

Je vais implémenter ce modèle et le tester sur différentes séquences, simulées et réelles, pour l'identification des transitions exon-intron dans l'ADN.

Mon second objectif est de tenter de préparer un modèle de Markov concret applicable à une séquence biologique que j'aurai choisie et de l'utiliser afin de mesurer des similarités entre les séquences, de l'entraîner avec une technique d'optimisation de réestimation et éventuellement l'utiliser afin de faire une recherche de séquences similaires dans une base de données génétique.

Enfin, si le temps le permet, cet algorithme sera adapté à une recherche de séquences dans une base de données de séquences d'ADN, dans laquelle, bien sûr, les régions ne seront pas déjà découpées.

1.2.2 Méthode(s) à mettre en œuvre dans l'atelier

- L'implémentation du HMM-jouet
 - Création d'un simulateur de séquences biologiques correspondant à des caractéristiques statistiques particulières
 - Choix de paramètres en fonction des valeurs fournies par Eddy
 - Implémentation de l'algorithme de Viterbi, et estimation de la séquence d'états optimale pour une séquence d'observations donnée
 - Détermination du critère de qualité
- Adaptation du HMM-jouet avec la réestimation des paramètres du modèle précédent
- La création d'un nouveau modèle adapté à des séquences caractérisées biologiquement et son utilisation sur une base de données de séquences réelles

1.2.3 Sources de données biologiques

PhyloFacts::DB Search by accession, taxonomy, biochemistry, localization, process...

http://phylogenomics.berkeley.edu/query_db/query_db.php

SPdb – a signal peptide database

<http://www.biomedcentral.com/1471-2105/6/249>

The GDB Human Genome Database

<http://www.gdb.org/>

The NCBI Entrez Gene

<http://www.ncbi.nlm.nih.gov/sites/entrez?db=unigene>

Gene Cards

<http://www.genecards.org/>

1.3 Références bibliographiques relatives au problème traité

Le sujet du traitement de l'information biologique avec des chaînes de Markov est relativement jeune puisque les techniques mathématiques les sous-tendant ne sont pas traditionnellement enseignées aux biologistes ni aux biochimistes. Avec l'éclosion du pluridisciplinaire, nous avons vu les horizons de la biologie s'élargir mais l'utilisation de cette technique n'est encore qu'à ses premiers balbutiements. Aussi la littérature scientifique est encore succincte et si elle regorge d'expériences scientifiques, il n'y a pas encore beaucoup de revues de cette littérature spécialisée.

Plusieurs scientifiques qui publient dans le domaine s'inspirent des recherches et de la revue faite par Rabiner, celui-ci publie dans le domaine de l'analyse et de la reconnaissance vocale, Un problème similaire s'il en est puisque dans la reconnaissance vocale, on a un lexique et une interprétation grammaticale.

Dans les séquences biologiques, le lexique est le code génétique et, en guise de grammaire, on a différentes interprétations biologiques de la séquence primaire. Lorsque la recherche scientifique sera plus avancée, on pourra définir des grammaires formelles basées sur les motifs.

La référence pour le cadre méthodologique de la technique de Markov est basée sur les travaux de Rabiner:

- 1) Lawrence R. Rabiner, [A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition](#). *Proceedings of the IEEE*, 77 (2), p. 257–286, February 1989.

Voici d'autres références de la littérature scientifique beaucoup plus directement en lien avec le problème traité :

- 2) Eddy SR. Profile hidden Markov models. *Bioinformatics*. 1998;14(9):755-63.

Quelques conférences et mémoires publiés sur le Web sont également des sources éloquentes et pédagogiques :

- 3) Pierre NICOLAS [Prédiction de gènes et chaînes de Markov cachées](#), pierre.nicolas@jouy.inra.fr
<http://genome.jouy.inra.fr/download/mig/M2-MBI-P7/genes-hmm.pdf.gz>
- 4) Arnaud Fontaine, Détection d'ARN non-codant mémoire pour le stage de Master Recherche de l'Université des Sciences et Technologies de Lille (spécialité informatique) 4 juillet 2005
<http://www2.lifl.fr/~fontaina/master/rapport.pdf>

Ces sources d'informations expliquent comment appliquer le modèle markovien à la séquence biologique.

- 5) Sean R Eddy, What is a hidden Markov model? **NATURE BIOTECHNOLOGY** VOLUME 22 NUMBER 10 OCTOBER 2004 1315

Cette dernière source d'information est un article de vulgarisation grand public, mais rédigé par le même auteur que «Profile hidden Markov model», il est donc un leader dans la publication de techniques de Markov appliquées aux données biologiques.

Enfin, ce tutorial est une précieuse source d'informations pour la partie pratique du projet.

- 6) [Hidden Markov Models University of Leeds](http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/viterbi_algorithm/s1_pg1.html)
http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/viterbi_algorithm/s1_pg1.html

2. Méthode(s) proposée(s) pour résoudre le problème

2.1 Notations et hypothèses

Nous posons l'hypothèse que le système peut être modélisé avec les étapes discrètes d'une machine d'état. Nous supposons qu'à un instant discret t , le système est dans un état donné $x(t)$. Le système n'est pas déterministe. À chaque instant le système a une probabilité donnée de transitionner à chaque état: c'est le modèle probabiliste de la machine d'état.

Nous posons également l'hypothèse que le modèle probabiliste convient à notre problème concret.

Des hypothèses plus précises sont formulées dans la section de la définition des objectifs de mon projet, après l'introduction des techniques et du modèle markovien.

Notation des observations effectuées par la lecture des éléments de séquence

- T = longueur de la séquence des observations (ensemble d'entraînement)
- O_t variable qui dénote l'observation à un temps t (variable de sortie)
- $\sigma = o_1, \dots, o_T$ (séquence des observations actuelles)

Notation des observations possibles

- M = nombre d'observations possibles (de l'ensemble d'entraînement)
- $\Omega(O) = \{v_1, \dots, v_M\}$ (ensemble des observations possibles)

Notation des états proposés par le modèle et observés

- N = nombre d'états
- $\Omega(X) = \{q_1, \dots, q_N\}$ (ensemble des états possibles)
- x_t variable qui dénote l'état à un temps t (variable d'état)

L'instant discret dans le cas qui nous occupe, soit la séquence biologique, se trouve à être la lecture d'un nucléotide supplémentaire, comme lors de l'activité de parser un texte. Donc dans notre machine d'état, chaque transition correspond à l'arrivée de l'élément suivant dans la lecture de chaîne de nucléotides.

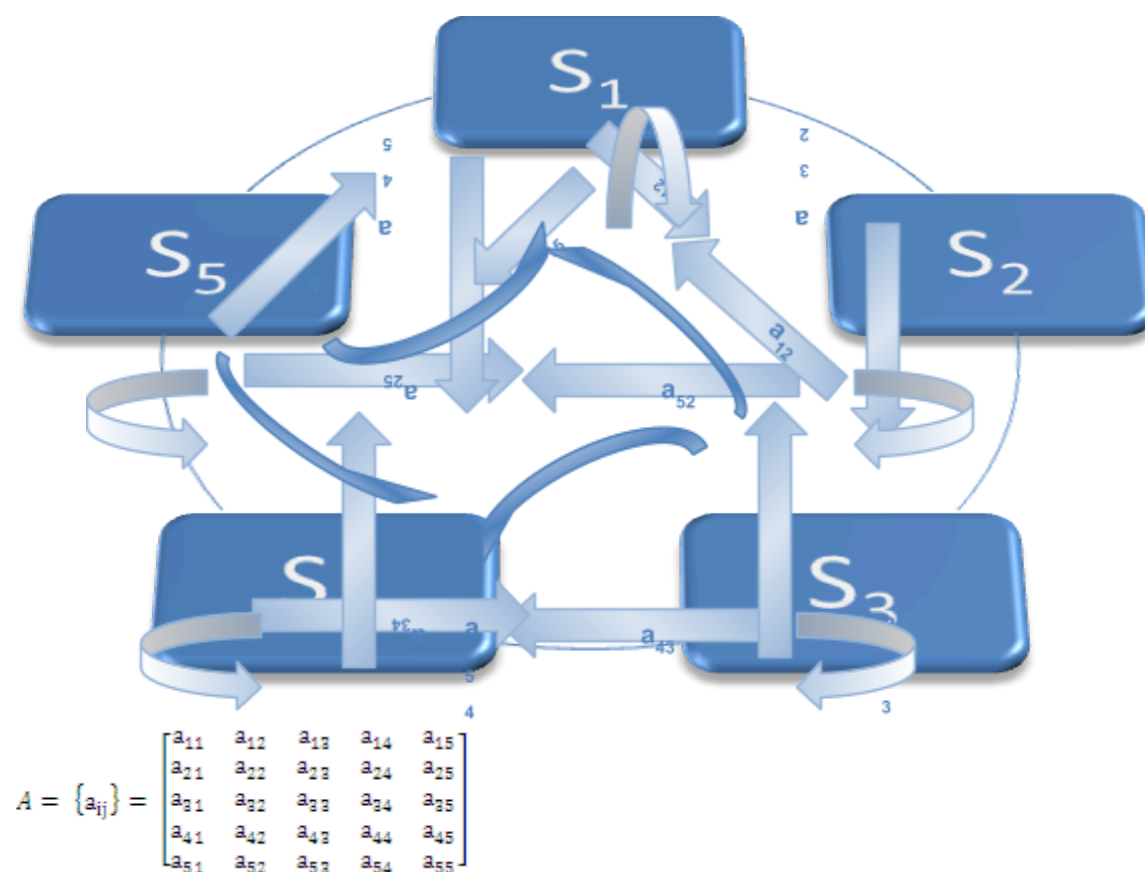
2.2 Présentation synthétique de la méthode, mettant en évidence les choix effectués par les auteurs

Nous allons d'abord introduire la technique de Markov, ses principaux problèmes et leurs solutions tels qu'exposés par Rabiner (1), puis j'en dériverai l'application biologique.

2.2.1 Chaînes de Markov du premier ordre

Selon la revue théorique effectuée par Rabiner, on commence notre étude avec un modèle simple, lequel modélise une suite d'événements observés, par une machine d'état. Dans cette machine d'état, chaque transition entre un état à un autre est modélisée par une flèche à laquelle est associée une probabilité. Une matrice $A = \{a_{ij}\}$ vient donc synthétiser la liste des probabilités interreliant tous les états possibles du modèle.

Figure 4. Illustration d'une machine d'état comportant 5 états.



Bien sûr, puisque ce sont des coefficients de probabilité qui caractérisent les arcs de transition, la somme des probabilités des arcs sortant d'un état donné est égale à 1 et toutes ces probabilités sont positives ou nulles.

$$a_{ij} \geq 0$$

$$\sum a_{ij} = 1$$

Ce modèle est dit du premier ordre si l'on pose l'hypothèse markovienne que la probabilité de l'état courant ne dépend que de l'état immédiatement précédent.

À partir d'un tel modèle, il est possible de considérer une séquence d'états et de calculer la probabilité d'occurrence d'une telle séquence étant donné le modèle.

$$P(O|\text{modele}) = \text{probabilité d'occurrence de la première observation} \\ * \text{les probabilités de transitions aux états suivants}$$

Il existe des variantes du modèle de base de Markov : par exemple, par opposition au modèle dit pleinement connecté (ergodique / fully connected), on trouve le modèle gauche-droite (left-to-right) pour lequel aucune transition d'un état d'indice supérieur vers un état d'indice inférieur n'est permise. Ce type de modèle progresse uniquement vers l'avant. Dans notre expérience, nous représenterons la matrice générale à N états.

2.2.2 Extension aux chaînes de Markov cachées

Le modèle précédent, pour lequel chaque état du modèle correspond à une observation est trop restrictif et ne permet pas de modéliser correctement plusieurs problèmes qui nous intéressent en traitement de l'information biologique et en traitement du signal.

Nous postulons ici que l'observation est une fonction probabiliste de l'état. On considère qu'il y a un processus aléatoire embarqué dans un autre processus aléatoire (comme des poupées gigognes). Un des deux processus aléatoires n'est pas directement observable, il est dit caché.

Un exemple similaire à notre application est celui des urnes et des balles. Dans cet exemple, un artiste choisit un panier parmi trois, puis une balle de laine dans l'urne choisie. À chaque fois, la couleur de la balle est enregistrée.

Dans ce modèle, chacun des N états définit une distribution de probabilité sur les M observations possibles.

Par exemple, dans notre cas, sur une séquence biologique donnée, nous voulons inférer la séquence d'états cachés. Il y a potentiellement plusieurs chemins d'états qui pourraient avoir généré la même séquence biologique, mais nous voulons trouver celui qui possède la plus grande probabilité (5).

La raison pour laquelle plusieurs chemins d'états restent plausibles est que les distributions de nucléotides entre régions se chevauchent partiellement et que, selon le cadre de lecture ou la position de

départ, différentes segmentations locales sont possibles. Au cours de l'évolution, l'ADN a été remanié et recombéné, de sorte que de nombreux fragments de gènes présentent des similarités ou des motifs inversés pouvant former des structures secondaires dans l'ARN.

Selon Eddy (5), le qualificatif « caché » des modèles de Markov cachés (HMM) vient du fait que même si seule la séquence de symboles biologiques est directement observée (nucléotides ou acides aminés), la séquence sous-jacente des états fonctionnels de la machine (exon, site d'épissage, intron) demeure cachée et doit être inférée.

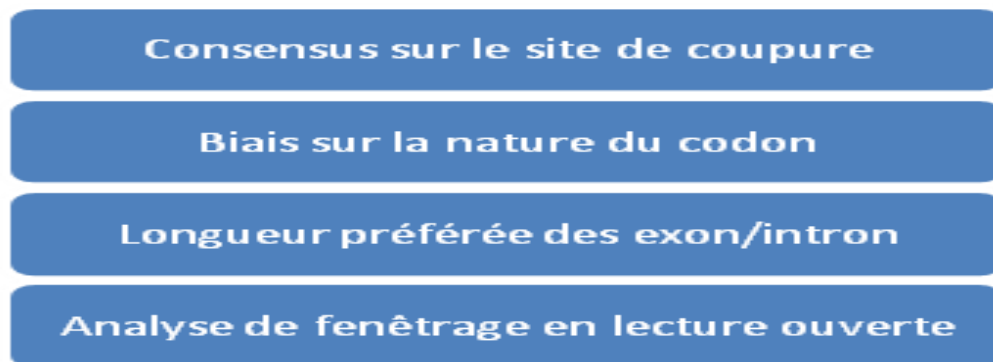
Le modèle probabiliste est donc astucieusement appliqué afin de mettre en relation des caractéristiques qualitatives biologiques avec des séquences polymériques dont les éléments sont déterminés.

Par exemple, dans sa version du modèle de Markov caché, Nicolas (3) décrit le modèle de Markov utilisé ainsi :

La **séquence cachée** $S = (S_1, S_2, \dots, S_n)$ décrit l'alternance des régions (ex: non codant, RBS, codant). Elle est modélisée par une chaîne de Markov.

La **séquence observée** $X = (X_1, X_2, \dots, X_n)$ correspond à la séquence d'ADN. Elle est modélisée conditionnellement à la séquence cachée.

Les informations biologiques qui peuvent être interprétées sont de diverses natures. Les différents paramètres qui peuvent être intégrés dans un système de scores pour un détecteur de gènes sont, selon Eddy (5) :



2.3 Algorithmes(s) utilisé(s)

2.3.1 Les 3 problèmes à résoudre des modèles de Markov (selon Rabiner (1))

Leurs solutions constituent en fait la liste des algorithmes qui sont utilisés de nos jours en collaboration pour travailler avec les modèles de Markov.

1 - Étant donné une séquence d'observations $O = O_1 O_2 \dots O_T$ et un modèle $\lambda = (A, B, \pi)$, comment calculer efficacement la probabilité totale d'observer cette séquence $P(O | \lambda)$?

2 - Étant donné une séquence d'observations $O = O_1 O_2 \dots O_T$ et un modèle λ , comment déterminer la séquence d'états cachés $Q = q_1 q_2 \dots q_T$ optimale, c'est-à-dire celle qui explique le mieux les observations au sens du maximum de probabilité conjointe $P(Q, O | \lambda)$?

3 - Comment ajuster les paramètres du modèle $\lambda = (A, B, \pi)$ afin de maximiser la vraisemblance $P(O | \lambda)$?

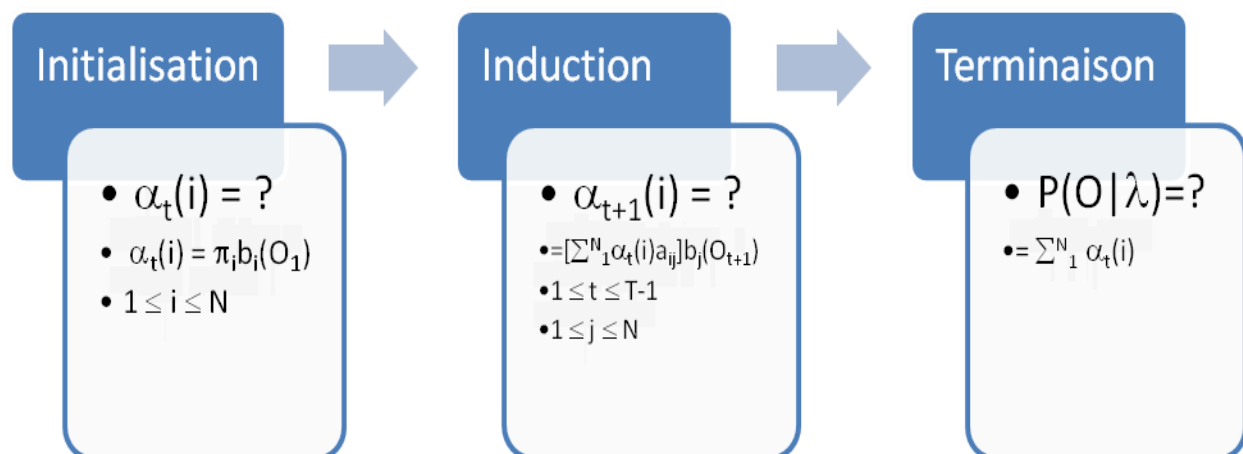
Le problème 1 - L'évaluation

1 - Étant donné une séquence d'observations $O = O_1 O_2 \dots O_T$ et un modèle $\lambda = (A, B, \pi)$, comment calculer efficacement la probabilité de la séquence d'observations $P(O | \lambda)$?

Le calcul direct de $P(O | \lambda)$ en sommant les probabilités de tous les chemins d'états possibles de longueur T exigerait d'énumérer N^T séquences d'états, ce qui entraîne une explosion exponentielle dès que la séquence s'allonge. Pour résoudre ce problème exactement en temps polynomial $O(T \cdot N^2)$, on utilise la programmation dynamique via la procédure AVANT-ARRIÈRE (Forward-Backward).

On remarque en effet qu'une énumération naïve repasserait sans cesse par les mêmes états aux mêmes instants et recalculerait inutilement les mêmes sous-chemins. L'algorithme AVANT (Forward) exploite cette structure en factorisant les calculs partiels par récurrence : on calcule les probabilités partielles $\alpha_{t+1}(j)$ à l'instant $t+1$ directement à partir des valeurs $\alpha_t(i)$ mémorisées à l'instant t .

Posons $\alpha_t(i) = P(O_1, O_2 \dots O_t, q_t = S | \lambda)$



Le problème 2 - Découverte de la partie cachée du modèle

2 - Étant donné une séquence d'observations $O = O_1 O_2 \dots O_T$ et un modèle λ , comment choisir la séquence d'états $Q = q_1 q_2 \dots q_T$ optimale qui explique le mieux les observations ?

On veut trouver la meilleure séquence d'états Q pour une séquence d'observations O donnée. Nous utilisons δ_t en tant que meilleur score (la probabilité la plus grande). La meilleure solution selon Rabiner (1) est l'Algorithme de Viterbi.

Par induction nous avons $\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] * b_j(O_{t+1})$

Pour retrouver la séquence d'états, nous devons garder la trace de l'argument qui a maximisé notre score pour chaque t et chaque j. Cette mémorisation est faite avec le vecteur $\psi_t(j)$

1. Initialisation

$$\delta_t(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N$$

$$\psi_t(i) = 0$$

2. Récursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$$

$$2 \leq t \leq T, 1 \leq j \leq N$$

$$\psi_t(i) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

$$2 \leq t \leq T, 1 \leq j \leq N$$

3. Terminaison

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)]$$

4. Chemin (séquence d'état) en parcours inverse

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

$$t = T - 1, T - 2, \dots, 1$$

$$\delta_t(i) = \pi_i b_i(O_t) \quad 1 \leq i \leq N$$

Le problème 3 - Optimisation du modèle

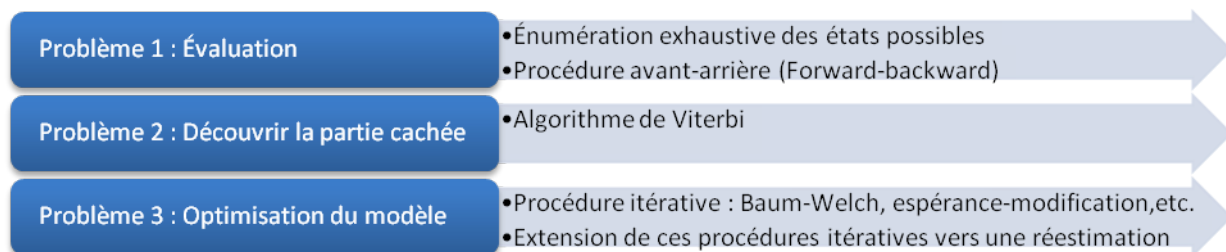
3 - Comment ajuster les paramètres du modèle $\lambda = (A, B, \pi)$ afin de maximiser $P(O | \lambda)$?

Il n'existe pas de formule analytique fermée permettant de calculer directement les paramètres $\lambda = (A, B, \pi)$ qui maximisent la vraisemblance $P(O | \lambda)$ lorsque les états sont cachés. On recourt donc à des procédures d'optimisation itératives telles que l'algorithme de Baum-Welch, qui est l'application aux chaînes de Markov cachées de l'algorithme d'espérance-maximisation (EM, Expectation-Maximization), ou à des méthodes de montée de gradient.

Cette procédure de réestimation opère de manière inductive en alternant deux étapes : l'étape d'espérance (E), qui utilise les variables avant (α) et arrière (β) pour calculer le nombre attendu de transitions de l'état i vers l'état j et d'émissions de chaque symbole, puis l'étape de maximisation (M), qui met à jour la probabilité de transition a_{ij} par le ratio entre l'espérance du nombre de transitions de i vers j et l'espérance du nombre total de transitions partant de i . Lorsque les états réels sont connus sur un jeu d'entraînement supervisé, ces probabilités s'estiment directement par comptage des fréquences empiriques.

Plus tard, on utilisera ce modèle avec des paramètres fixes afin d'analyser de nouvelles données dont les caractéristiques ne sont pas aussi bien connues que celles de l'ensemble d'entraînement.

Figure 5. Résumé des problèmes et solutions au modèle de Markov selon Rabiner (1)



3. Résultats

3.1 Rappel des objectifs poursuivis

Rappelons que le premier objectif consiste à implémenter le modèle simple (modèle jouet) et à utiliser l'algorithme de Viterbi afin d'analyser des séquences d'observations.

3.2 Conditions expérimentales

Les expériences ont été réalisées sur Matlab 5.0, version éducationnelle.

Le code utilisé pour implémenter la recherche de séquences est fourni en fichier joint (Annexe 4).

Les données utilisées ont été simulées avec un générateur de séquences dont le code est également annexé (Annexe 4). Les caractéristiques statistiques des segments ont été introduites dans un modèle de simulation.

Dans le cas des séquences simulées, le critère de qualité est la détection du site d'épissage 5' (état S) qui marque la frontière entre un exon et le départ d'un intron dans chaque séquence fournie. Un vrai site S est présent par séquence, et l'on mesure la distance en nucléotides entre la transition détectée et le site réel, ainsi que le nombre de faux positifs.

Voici les paramètres utilisés :

La machine d'états possède 3 états qui représentent des régions fonctionnelles de l'ADN : EXON (E), INTRON (I) et SITE D'ÉPISSAGE 5' (S), donc $N = 3$.

Le modèle présente 4 observations possibles pour l'ADN : 'A', 'C', 'G' et 'T' donc $M = 4$

Les séquences observées (simulées) sont longues de 26 nucléotides, donc nous avons $T = 26$

- Le paramètre PI représente les probabilités initiales de se trouver dans telle ou telle région
- Le paramètre A représente les probabilités dans une région donnée (état) de transitionner dans l'autre région.
- Le paramètre B représente les probabilités dans une région donnée (état) d'avoir tel événement (nucléotide)

```
parametre_PI = [probabilite_exon probabilite_intron probabilite_site] = [0.66 0.33 0.01];
```

```
parametre_A = [0.7 0.0 0.3; % region_exon aux autres
```

```
0.3 0.7 0.0; % region_intron aux autres
```

```
0.0 1.0 0.0]; % region_site aux autres
```

```
parametre_B = [distribution_exon' distribution_intron' distribution_site'];
```

```
= [[0.25 0.25 0.25 0.25]' [0.4 0.1 0.1 0.4]' [0.05 0 0.95 0]'];
```

3.3 Résultats

Voici le résultat d'une exécution moyennée sur 100 échantillons simulés, où chaque échantillon est une séquence de 26 nucléotides structurée selon les 4 segments suivants : 10 E, 1 S, 10 I, 5 E (soit T = 26 nucléotides au total).

L'expérience vise à mesurer si le site S est détecté et si oui, à quelle distance du site réel la machine change d'état. Cette mesure est similaire à celle utilisée dans la littérature qui consiste à vérifier si les exons sont détectés ou non.

Indice de qualité	Valeur
Moyenne faux positif	3.0 occurrences
Moyenne erreur	2.0 nucléotides de distance

Identifications	TOTAL	VRAIS	FAUX
Positifs	400	VP = 100	FP = 300
Négatifs	0	0	FN = 0

Sensibilité (Rappel) S_n = nombre de vrais sites S détectés / nombre total de vrais sites S = $VP / (VP + FN) = 100 / (100 + 0) = 100 \%$

Spécificité (au sens bioinformatique de la précision / valeur prédictive positive, cf. Burge & Karlin 1997)
 S_p = nombre de vrais sites S détectés / nombre total de sites S prédits = $VP / (VP + FP) = 100 / (100 + 300) = 25 \%$

Deux versions de l'implémentation de Viterbi ont été réalisées, toutes deux équivalentes sur le plan algorithmique : la première est basée sur une formulation récursive et la seconde sur une boucle itérative dynamique. La version récursive initiale se heurtait à une limite matérielle de pile d'appels (erreur « Stack overflow » au-delà de 256 nucléotides, la profondeur de pile procédurale par défaut de Matlab 5.0 étant limitée à 256 appels). Comme les séquences génomiques réelles atteignent plusieurs milliers de nucléotides, la seconde version, purement itérative, a été implémentée pour lever cette limitation.

Pour des séquences courtes (T = 26), le temps d'exécution apparaît quasi instantané quelle que soit l'implémentation choisie ; le temps de calcul n'a donc pas été un facteur limitant dans cette expérience.

Sur le plan théorique, l'utilisation de la programmation dynamique (algorithme de Viterbi) réduit drastiquement la complexité algorithmique en remplaçant une énumération exhaustive en $O(T \cdot N^T)$ par un parcours matriciel en $O(T \cdot N^2)$. Dans notre expérience (N = 3 et T = 26), nous remplaçons ainsi un parcours de 26×3^{26} ($\approx 6,6 \times 10^{13}$ opérations) par seulement $26 \times 3^2 = 234$ opérations élémentaires. Cet écart d'ordre de grandeur devient encore plus décisif sur des séquences biologiques réelles comptant des milliers de nucléotides.

4. Discussion

4.1 Analyse critique des résultats

4.1.1 Analyse des résultats de l'expérience

Nous observons une excellente sensibilité (100 % de détection des transitions réelles, avec une erreur moyenne de positionnement de seulement 2,0 nucléotides), mais une faible précision prédictive (beaucoup de faux positifs : 3 fausses détections en moyenne pour chaque détection véritable, soit une valeur prédictive positive / spécificité au sens bioinformatique de 25 %).

Ceci s'explique par la grande simplicité du modèle jouet à 3 états ainsi que par l'absence d'entraînement supervisé sur un corpus étendu, les probabilités de transition ayant été fixées manuellement à titre illustratif.

En résumé, nos résultats présentent une sensibilité maximale (100 %) mais une spécificité prédictive réduite (25 %) : toutes les vraies jonctions sont captées, mais le modèle déclenche également des transitions parasites dès qu'un court motif riche en A/T est précédé d'un G dans un exon. En pratique, un tel détecteur peut servir de premier filtre de criblage rapide, à condition d'être suivi d'une étape de validation ou d'être enrichi par un modèle d'ordre supérieur.

4.1.2 Résultats d'expériences similaires par la communauté

On voit que les modèles élaborés par la communauté, qui sont plus complexes et mieux entraînés, nous donnent une meilleure spécificité. La sensibilité plus faible de leur résultat s'explique sans doute par les objectifs de détection qui sont plus sévères que les miens : parfois des alignements devaient être réalisés, et parfois des modélisations de gènes avec des régions beaucoup plus complexes étaient réalisées. Peut-être qu'un biais de détection a pu être introduit par un surentraînement également.

Programs	# of seq	Nucleotide accuracy				Exon accuracy							
		<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>CC</i>	<i>ESn</i>	<i>ESp</i>	$(ESn+ESp)/2$	<i>ME</i>	<i>WE</i>	<i>PCa</i>	<i>PCp</i>	<i>OL</i>
FGENES	195(5)	0.86	0.88	0.84	0.83	0.67	0.67	0.69	0.12	0.09	0.20	0.17	0.02
GeneMark	195(0)	0.87	0.89	0.84	0.83	0.53	0.54	0.54	0.13	0.11	0.29	0.27	0.09
Genie	195(15)	0.91	0.90	0.89	0.88	0.71	0.70	0.71	0.19	0.11	0.15	0.15	0.02
Genscan	195(3)	0.95	0.90	0.91	0.91	0.70	0.70	0.71	0.08	0.09	0.21	0.19	0.02
HMMgene	195(5)	0.93	0.93	0.91	0.91	0.76	0.77	0.76	0.12	0.07	0.14	0.14	0.02
Morgan	127(0)	0.75	0.74	0.70	0.69	0.46	0.41	0.43	0.20	0.28	0.28	0.25	0.07
MZEF	119(8)	0.70	0.73	0.68	0.66	0.58	0.59	0.59	0.32	0.23	0.08	0.16	0.01

"Evaluation of gene finding programs" S. Rogic, A. K. Mackworth and B. F. F. Ouellette. Genome Research, 11: 817-832 (2001).

Table 1. Performance comparison for Burset/Guigó set of 570 vertebrate genes
A Comparison of GENSCAN with other gene prediction programs

Program	Sequences	Sn	Accuracy per nucleotide			Sn	Accuracy per exon			
			Sp	AC	CC		Sp	Avg.	ME	WE
GENSCAN	570 (8)	0.93	0.93	0.91	0.92	0.78	0.81	0.80	0.09	0.05
FGENEH	569 (22)	0.77	0.88	0.78	0.80	0.61	0.64	0.64	0.15	0.12
GeneID	570 (2)	0.63	0.81	0.67	0.65	0.44	0.46	0.45	0.28	0.24
Genie	570 (0)	0.76	0.77	0.72	n/a	0.55	0.48	0.51	0.17	0.33
GenLang	570 (30)	0.72	0.79	0.69	0.71	0.51	0.52	0.52	0.21	0.22
GeneParser2	562 (0)	0.66	0.79	0.67	0.65	0.35	0.40	0.37	0.34	0.17
GRAIL2	570 (23)	0.72	0.87	0.75	0.76	0.36	0.43	0.40	0.25	0.11
SORFIND	561 (0)	0.71	0.85	0.73	0.72	0.42	0.47	0.45	0.24	0.14
Xpound	570 (28)	0.61	0.87	0.68	0.69	0.15	0.18	0.17	0.33	0.13
GeneID+	478 (1)	0.91	0.91	0.88	0.88	0.73	0.70	0.71	0.07	0.13
GeneParser3	478 (1)	0.86	0.91	0.86	0.85	0.56	0.58	0.57	0.14	0.09

Source : Chris Burge* and Samuel Karlin Prediction of Complete Gene Structures in Human Genomic DNA J. Mol. Biol. (1997) 268, 78±94

N.B.: Il existe d'autres frameworks qui auraient pu être comparés mais ceci s'inscrit en dehors du scope de ce projet: hmms (code GNU sous linux), HMM Matlab Toolbox (contribuée) et Biojava (librairie Java), EasyGene, etc.

4.1.3 Explication des résultats

Un premier facteur expliquant ces résultats réside dans le paramétrage a priori du modèle : dans un problème réel où les caractéristiques statistiques fines des séquences varient selon les familles de gènes, un modèle non réestimé demeure trop générique et génère des faux positifs.

L'autre facteur est la simplicité naïve du modèle utilisé qui ne donne pas assez d'informations sur les données à modéliser. Dans cette situation, il existe trop de bonnes solutions pour l'algorithme de Viterbi qui a le loisir de cartographier presque n'importe quelle région du génome à n'importe quel état sans pour autant avoir un mauvais score.

Cette simplicité découle en partie du fait que physiquement, nous manquons d'informations sur la nature exacte des sites d'épissage 5' dans un modèle d'ordre 1. Ceux-ci n'y sont représentés que par un seul nucléotide dont l'occurrence isolée n'est pas exclusive à la jonction exon-intron, ce qui explique les problèmes de spécificité observés.

Cependant, ce manque d'information pourrait être compensé par d'autres informations en complexifiant le modèle et en décrivant plusieurs sous-types de régions, ce qui ajouterait davantage d'états à la machine d'états.

La sensibilité obtenue, qui semble supérieure, est sans doute due aux caractéristiques statistiques idéales de nos échantillons simulés.

4.1.4 Suggestion d'améliorations à la procédure

Longueur des régions

La plupart du temps, les sections des introns ont des longueurs importantes ; il est donc peu probable que les sites d'épissage S soient suivis de régions INTRON de longueur inférieure à un certain seuil, comme c'est le cas dans mes résultats. Ceci ne fait pas partie du modèle de Markov utilisé dans l'expérience mais pourrait éventuellement être inclus dans un algorithme modifié. En effet, plus la séquence est depuis longtemps dans le même état INTRON, plus elle a de chances de transitionner vers l'état EXON, ce qui n'est pas le cas de l'état EXON.

Note rétrospective (2026) - Formulation indépendante du principe des modèles semi-markoviens cachés (HSMM)

En 2008, en analysant uniquement les faux positifs de ce banc d'essai basé sur le modèle d'ordre 1 de Rabiner (1989) et d'Eddy (2004), j'avais déduit de manière indépendante que la probabilité de sortie d'un état devait dépendre du temps déjà écoulé dans cet état (« plus la séquence est depuis longtemps dans l'état INTRON, plus elle a de chances de transitionner vers l'état EXON »). Dans une chaîne de Markov standard du 1er ordre, la durée de séjour dans un état est en effet « sans mémoire » (loi géométrique). Conditionner explicitement la transition à la durée parcourue constitue le fondement mathématique des modèles semi-markoviens cachés (Hidden Semi-Markov Models, HSMM), qui sont au cœur des prédicteurs de gènes avancés (tels que GENSCAN) et de la modélisation temporelle moderne.

Complexité des modèles (nombre d'états)

Des modèles plus complexes iront modéliser d'autres types de régions :

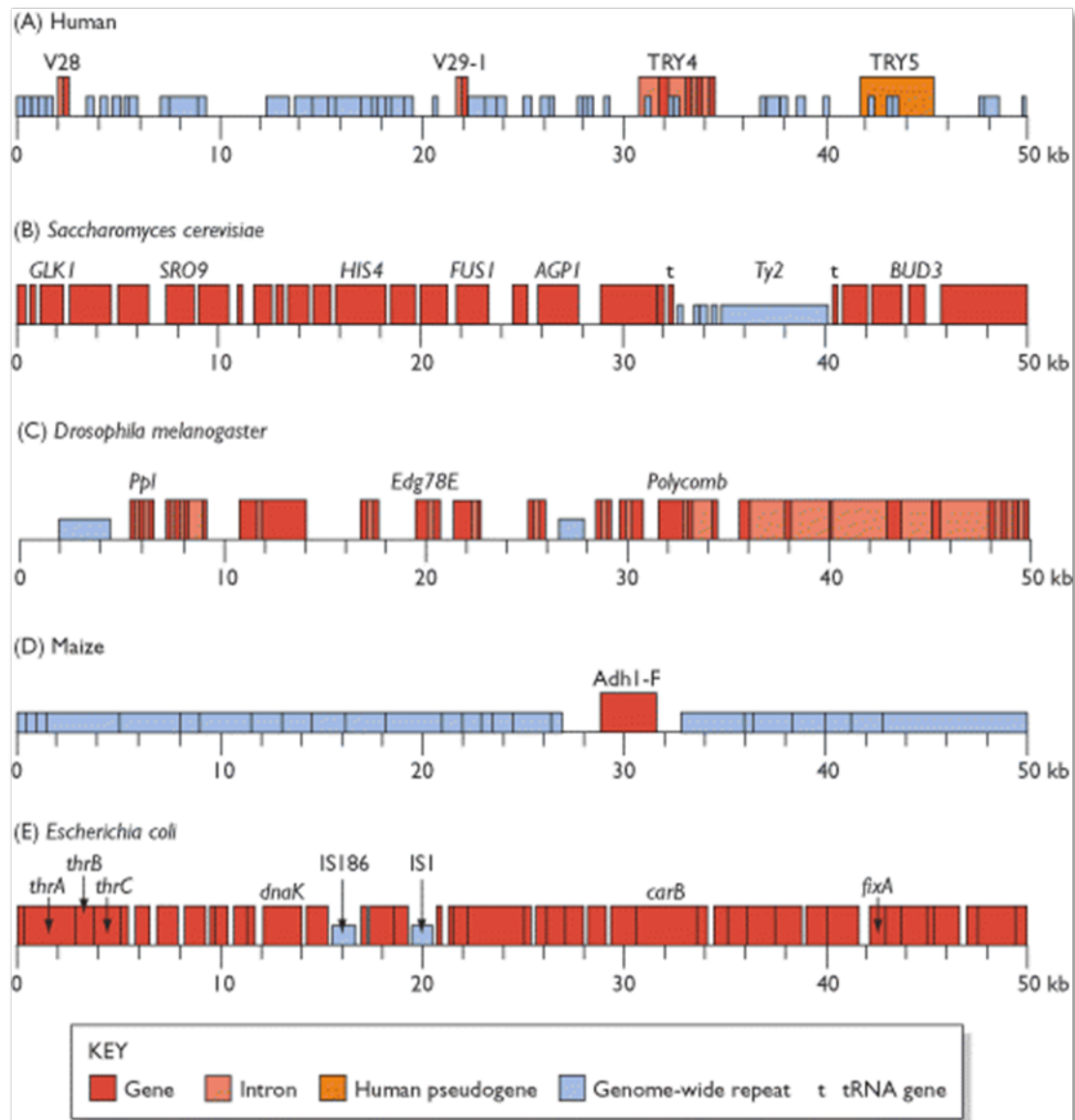
- Premièrement, ces modèles tiendront compte du début du gène et de la fin du gène dans l'ADN.
- Ensuite, ces modèles tiendront compte de régions codantes avec des distributions statistiques différentes.
- Enfin, les sites de début de transcription, de début de traduction et de coupure pour l'épissage font l'objet de modèles plus précis qui tiennent compte des versions alternatives de ces sites et des positions des nucléotides qui donnent le « signal ».

Les dépendances de nucléotides adjacents

Non seulement des modèles de Markov cachés plus riches peuvent être dérivés, mais il existe aussi une variante de l'algorithme pour ce qui concerne les dépendances entre nucléotides adjacents ou non adjacents, comme par exemple dans les sites de coupure ou de début de transcription/traduction. Une telle modélisation des dépendances entre nucléotides permettra de modéliser les répétitions de séquences et de séquences inversées, comme on en observe dans les structures secondaires.

Cette technique se nomme : **Maximal Dependency Decomposition (MDD)**.

Cette illustration montre comment la complexité générale d'un modèle génétique dépend de l'espèce étudiée. Tel qu'expliqué précédemment dans le contexte biologique, les fonctions des protéines codées ont aussi une incidence sur les proportions d'acides aminés et donc sur les nucléotides les codant.



Source : Brown Fig 2.2 BIOS Scientific Publishers Ltd. 1999

Cette illustration montre un modèle plus complexe élaboré par une équipe tout spécialement pour détecter les gènes d'E. coli.

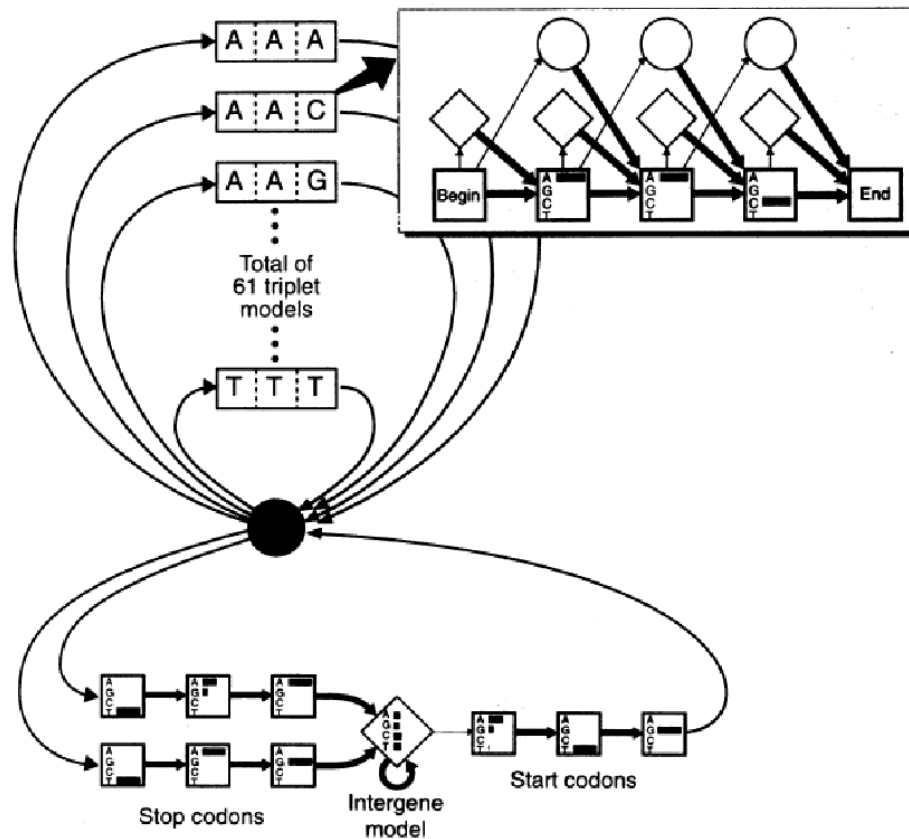


Figure 1. HMM architecture for a parser for *E. coli* DNA with a simple intergenic model. The central state (shaded circle), generates no nucleotides and is used to connect all the models. The 61 triplet or codon models above the central state all have identical structures, shown in detail for the codon AAC. Squares represent main states; diamonds denote a state where a nucleotide can be inserted between consecutive codon nucleotides whereas circles generate no nucleotide and can be used to delete one of the three nucleotides. The thickness of the arrows indicate the fraction of sequences making the given transition. The insert state in the middle of the intergenic model (diamond) produces random sequences from a base distribution estimated from the actual distribution of bases in the intergenic regions of the training set. The four bases have almost the same frequency.

Source : Krogh et al. A hidden Markov model that finds genes in *E. coli* DNA 4768-4778 Nucleic Acids Research, 1994, Vol. 22, No. 22

4.2 Comparaison avec les résultats fournis par d'autres méthodes

Historique ou techniques alternatives utilisées jusqu'à présent

Différentes techniques ont été utilisées jusqu'à présent par la communauté scientifique afin de réaliser des alignements ou des détections de séquences:

La programmation dynamique

Cette technique mathématique exacte (dont l'algorithme de Viterbi et les alignements de paires sont des applications directes) permet de trouver le maximum global de l'espace de recherche en temps polynomial en mémorisant les sous-solutions optimales. Toutefois, son application directe à l'alignement multiple simultané d'un grand nombre de séquences mène à une explosion combinatoire (problème NP-complet), ce qui motive le recours aux heuristiques ci-dessous.

La construction d'alignements dynamiques

Cette heuristique utilise un critère de similarité et des comparaisons de séquences voisines afin de déterminer une base de travail, soit un premier alignement partiel, puis construit à partir de celui-ci l'alignement du reste de la séquence.

Les méthodes itératives

À la manière des techniques de descente de gradient, ces techniques répètent la construction d'alignements dynamiques en choisissant parfois de réaligner les séquences initiales. Aucune séquence n'est définitivement alignée avant la convergence.

Les algorithmes génétiques

Originellement, le problème à analyser devient ici une source d'inspiration pour la technique qui l'optimisera. Il devient ici une source de modifications aléatoires de la solution combinée avec un algorithme de descente de gradient, donc une astucieuse combinaison d'un processus systématique et déterministe avec un processus non-déterministe qui explore l'espace de solution.

Bien que l'algorithme génétique puisse être appliqué aux paramètres de notre modèle de Markov et non pas directement aux séquences, il existe aussi une branche de la bioinformatique qui l'applique directement aux séquences.

4.3 Mise en évidence de faits saillants

Voici un recensement des causes d'erreur de cette méthode selon la littérature bioinformatique. Ce recensement met en lumière les facteurs et les caractéristiques exceptionnelles de certaines séquences qui les rendent difficiles à détecter par nos techniques automatiques basées sur un modèle homogène de premier ordre :

- Décalage de fenêtrage : quelques séquences stockées dans une base de données peuvent contenir des décalages résultant d'erreurs de séquençage.
- De très longs gènes : le gène de la dystrophine humaine se compose de 79 exons s'étendant sur 2,3 Mb.
- De très longs introns : le gène de la dystrophine possède quelques introns > 100 kb représentant > 99 % du gène.
- Des introns très conservés entre les séquences différentes.
- De très courts exons : par exemple seulement 3 bases. Ils sont difficiles à détecter, spécialement s'ils sont bordés par de larges introns ou si leur longueur est un multiple de 3 et n'introduit pas de changement dans le fenêtrage.
- Gènes superposés : ils sont très rares mais cela existe, ainsi que les arrangements polycistroniques : 1 gène -> 1 ARNm -> 2+ protéines.
- Introns dans la région du site UTR : il arrive que la région 5'- et/ou 3'-UTR dans l'ARN pré-messager soit interrompue par un ou plusieurs introns.
- Départ de transcription alternatif : par exemple la dystrophine en possède 3, découpage alternatif, polyadénylation alternative, initiation alternative de la traduction (trouver le bon initiateur AUG ou son alternative telle que parfois ACG dans Arabidopsis ou CUG chez l'humain).

4.4 Relation avec les technologies de l'estimation

J'aimerais commencer en citant Eddy (5)

"An HMM is a full probabilistic model - the model parameters and the overall sequence 'scores' are all probabilities. Therefore, we can use Bayesian probability theory to manipulate these numbers in standard, powerful ways, including optimizing parameters and interpreting the significance of scores."

Donc plusieurs liens peuvent être établis avec les concepts de la théorie de l'estimation.

Premièrement, le modèle de Markov introduit un formalisme probabiliste discret au sein d'une machine d'états, là où de nombreuses manipulations classiques d'estimation en traitement du signal sont généralement réalisées sous des hypothèses linéaires et gaussiennes. Autant la mise en place du modèle que l'interprétation a posteriori des résultats peuvent ainsi tirer parti du cadre de l'inférence bayésienne.

Deuxièmement, comme dans la plupart des techniques d'estimation en présence de variables cachées, il est question d'optimiser les paramètres du modèle. Ces estimations par maximum de vraisemblance (approche fréquentiste) ou maximum a posteriori (approche bayésienne) ne peuvent cependant pas être résolues par une formule analytique directe (fermée) dès que le modèle comporte plusieurs états cachés, car la vraisemblance globale somme une multitude de chemins possibles. C'est pourquoi on applique des procédures de réestimation itératives (comme l'algorithme d'espérance-maximisation de Baum-Welch) ou des méthodes d'échantillonnage stochastique.

5. Conclusion

5.1 Remarques générales

L'approche utilisée par les auteurs me semble très judicieuse. Les pas effectués dans le domaine de la reconnaissance vocale sont mis à profit en empruntant les algorithmes qui y sont développés.

La solution au problème numéro 2, la découverte de la partie cachée, dépend également de la modélisation effectuée par le chercheur et non pas seulement de l'algorithme, en vérité c'est très subjectif et ne possède pas de solution exacte.

Cependant, pour les questions du problème 1 (évaluation de la probabilité par le modèle) et du problème 3 (entraînement du modèle par estimation des paramètres), l'espace de solutions est impossible à énumérer par force brute et il est judicieux de la part des auteurs de mettre en place des algorithmes basés sur l'induction (programmation dynamique) et la réestimation itérative.

Je formulerais seulement un bémol par rapport à cette approche : comme le mentionne lui-même l'auteur (Rabiner), ces techniques d'optimisation finissent par converger sur un minimum ou maximum local, ce qui peut généralement être acceptable mais pas toujours. La raison pour laquelle cette convergence vers un optimum local se produit est que l'algorithme de réestimation est purement déterministe, comparativement aux métaheuristiques qui ont la propriété de mélanger une part de hasard (pour une exploration plus complète) à une approche systématique. On parle souvent de « refroidissement progressif » d'un tel système lorsqu'on réduit graduellement la part de l'aléatoire par rapport au processus déterministe. Ces approches hybrides ont ainsi la propriété de favoriser la découverte du minimum ou maximum global et de ne pas se contenter du premier optimum local venu. C'est ainsi que certains scientifiques ont d'ores et déjà commencé à utiliser les algorithmes génétiques ou le recuit simulé (simulated annealing) pour résoudre ces problèmes d'estimation des paramètres du modèle (problème no 3). L'objet de ma critique est seulement de souligner que ces techniques existent et qu'elles m'apparaissent encore plus prometteuses que l'approche purement déterministe.

Encore qu'il faudrait démontrer en pratique qu'une implémentation est réalisable. Ces techniques ne sont pas encore bien caractérisées en application au modèle Markovien.

Note rétrospective (2026) - De l'hybridation stochastique (2008) à l'échantillonnage de chemins lumineux en rendu 3D (ReSTIR / MCMC)

Cette réflexion formulée en conclusion en 2008 - coupler une chaîne de Markov avec une exploration stochastique contrôlée et un « refroidissement progressif » afin d'échapper aux optima locaux et de mieux parcourir l'espace des solutions - trouve un écho direct dans les méthodes contemporaines de traitement du signal et de rendu graphique en temps réel. En particulier, les algorithmes modernes d'illumination globale et de débruitage spéculaire (tels que ReSTIR et ses extensions par mutations MCMC) reposent exactement sur cette alliance entre l'échantillonnage de Monte-Carlo et les chaînes de Markov pour guider et réutiliser les trajectoires lumineuses les plus porteuses sans introduire de biais.

Annexe 1 - Code génétique

Tableau A1.1 - Code génétique

	U	C	A	G
U	UUU Phénylalanine UUC UUA Leucine UUG	UCU UCC Sérine UCA UCG	UAU Tyrosine UAC UAA Stop UAG Stop	UGU Cystéine UGC UGA Stop UGG Tryptophane
C	CUU CUC Leucine CUA CUG	CCU CCC Proline CCA CCG	CAU Histidine CAC CAA Glutamine CAG	CGU CGC Arginine CGA CGG
A	AUU Isoleucine AUC AUA Méthionine/Start AUG	ACU ACC Thréonine ACA ACG	AAU Asparagine AAC AAA Lysine AAG	AGU Sérine AGC AGA Arginine AGG
G	GUU GUC Valine GUA GUG	GCU GCC Alanine GCA GCG	GAU Acide GAC aspartique GAA Acide GAG glutamique	GGU GGC Glycine GGA GGG

Tableau A1.2 - Code génétique - Tableau inverse

Ala	GCU, GCC, GCA, GCG	Leu	UUA, UUG, CUU, CUC, CUA, CUG
Arg	CGU, CGC, CGA, CCG, AGA, AGG	Lys	AAA, AAG
Asn	AAU, AAC	Met	AUG
Asp	GAU, GAC	Phe	UUU, UUC
Cys	UGU, UGC	Pro	CCU, CCC, CCA, CCG
Gln	CAA, CAG	Ser	UCU, UCC, UCA, UCG, AGU, AGC
Glu	GAA, GAG	Thr	ACU, ACC, ACA, ACG
Gly	GGU, GGC, GGA, GGG	Trp	UGG
His	CAU, CAC	Tyr	UAU, UAC
Ile	AUU, AUC, AUA	Val	GUU, GUC, GUA, GUG
START	AUG	STOP	UAG, UGA, UAA

Annexe 2 - Classification des acides aminés

Tableau A2.1 - Classes d'acides aminés

les acides aminés aliphatiques	•Glycine •Alanine •Valine •Leucine •Isoleucine •Proline
les acides aminés aromatiques	•Phénylalanine •Tryptophane •Tyrosine
les acides aminés hydrophiles	•Sérine •Thrénine
les acides aminés hydrophobes	•Glycine •Alanine •Valine •Leucine •Isoleucine •Proline •Phénylalanine •Tryptophane
les acides aminés alcools	•Sérine •Thrénine •Tyrosine
les acides aminés soufrés	•Cystéine •Méthionine
les acides aminés azotés ou amides	•Asparagine •Glutamine
les acides aminés basiques	•lysine •arginine •histidine
les acides aminés acides	•aspartate •glutamate

Annexe 3 - Abréviations des éléments de séquence biologique

Tableau A3.1 - Nomenclature des acides aminés

Nom de l'acide aminé	Code de trois lettres	Code d'une lettre
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cystéine	Cys	C
Glutamate	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Méthionine	Met	M
Phénylalanine	Phe	F
Proline	Pro	P
Sérine	Ser	S
Thréonine	Thr	T
Tryptophane	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Tableau A3.2 - Nomenclature des nucléotides de l'ARN

Nom du nucléotide	Code d'une lettre	Complémentaire (ADN)	Complémentaire (ARN)
Adénosine	A	T	U
Cytosine	C	G	G
Guanine	G	C	C
Uracile	U	A	A

Tableau A3.3 - Nomenclature des nucléotides de l'ADN

Nom du nucléotide	Code d'une lettre	Complémentaire (ADN)	Complémentaire (ARN)
Adénosine	A	T	U
Cytosine	C	G	G
Guanine	G	C	C
Thymine	T	A	A