THÈSE

de Doctorat de l'Université Paul Sabatier, Toulouse 3
Discipline : **Bioinformatique**

LOCALISATION DE GÈNES ET VARIANTS PAR INTÉGRATION D'INFORMATIONS

Sylvain Foissac 2004

Département de Mathématiques et Informatique Appliquées
Institut National de la Recherche Agronomique

Soutenue le 15 décembre 2004 devant la commission d'examen :

Claude THERMES, Chercheur CNRS ..... Rapporteur
Alain VIARI, Dir. de Recherche INRIA ..... Rapporteur
Gwennaele FICHANT, Professeur UPS ..... Examinatrice
Michel WEBER, Dir. de Recherche UPS ..... Examinateur
Thomas SCHIEX, Dir. de Recherche INRA ..... Directeur de thèse

Table des matières

Merci !

En théorie, cette partie du document est traditionnellement consacrée aux remerciements d'usage. En pratique[1]*, elle offre en outre l'opportunité à l'impétrant de faire part de son ressenti personnel sur l'expérience vécue et d'exprimer sa personnalité plus librement que dans le cadre strict du discours scientifique. C'est pourquoi la première personne que je souhaite remercier n'est autre que l'inventeur (au nom malheureusement tombé dans l'oubli*[2]*) de la partie "Remerciements" des thèses, ne serait-ce que pour le plaisir que j'ai eu à en lire certaines et celui que j'ai à écrire celle-ci, qui j'espère en procurera à d'autres.*

Je remercie Thomas Schiex, mon directeur de thèse, pour tout ce qu'il a fait pour moi et tout ce qu'il m'a apporté. Merci Thomas, merci tout d'abord de m'avoir donné ma chance, moi qui au départ n'avais pas bien plus à proposer dans certains domaines que ma bonne volonté. Merci d'avoir fait preuve (tel un expérimentateur biologiste) de la curiosité scientifique nécessaire à la réalisation de l'expérience qui consistait à plonger un biologiste en cours de virage dans un milieu de mathématiciens et informaticiens sur un sujet de bioinformatique pure pour voir le résultat. Merci de m'avoir fait profiter de tes talents pédagogiques[3]*. Merci pour ton accessibilité et ta disponibilité.*

Concernant la direction scientifique et le "management"*, merci de m'avoir laissé une grande liberté, en préférant te limiter à donner ton avis (toujours volontiers, avec franchise et argumentation à la clef — ce pourquoi je te suis également reconnaissant) sans imposer les choix. Merci pour ton attitude et la liberté laissée vis-à-vis de mes activités "extra-thésiennes" "chronophages" (comme le monitorat, les formations, et certaines activités sportives. . .). Merci pour ta confiance et ta considération.*

Au sujet de la formation professionnelle, je te remercie de m'avoir fait partager ton éthique et pour m'avoir offert la possibilité de participer à un

[1] Et l'on sait que si en théorie, la différence entre la théorie et la pratique est nulle, en pratique, c'est rarement le cas.

[2] voir à ce sujet la dernière citation de ce document (page 136).

[3] *"Si vous n'arrivez pas à expliquer un concept à un enfant de 6 ans, c'est que vous ne maîtrisez pas ce concept."* (A. Einstein)

large pannel des activités du métier de chercheur, quand il est (malheureusement) facile de ne laisser aux thésards que les pénibles besognes. Merci d'avoir encouragé mes initiatives et l'expression de mes idées, réflexions, interrogations, parfois de mes délires. Merci de ne pas avoir tenu rigueur des énormités que cela pouvait parfois engendrer de ma part. Merci pour ton soutien et ton honnêteté. Thomas, je suis conscient de la chance que j'ai eue d'avoir travaillé avec toi durant ces quelques années, qui furent pour moi formidables, tant professionnellement que humainement. Et merci à feu la porte vitrée de l'Aligot pour son indulgence...

Deux autres personnes ont aussi une grande part de responsabilité dans l'aboutissement de mon travail et dans le plaisir que j'ai eu à le réaliser au sein de l'unité BIA ; il s'agit de Régis Sabbadin et de Patricia Thébault, dont l'aide fut cruciale notamment grâce à leurs lectures et corrections des multiples versions du manuscrit, et grâce à leur précieux soutien moral.

Régis, merci de m'avoir fait bénéficier de ton ouverture d'esprit exemplaire (les thèses de bioinformatique auxquelles tu t'es intéressé étant relativement éloignées de ta thématique de recherche sur l'aide à la décision) et de ton judicieux sens critique. Merci également pour tes talents de "coach" *remarquables et d'ailleurs déjà remarqués (Thébault, 2004), qu'il s'agisse de faire parvenir un(e) thésard(e) à la soutenance, l'unité BIA à la victoire des Tolosanes de l'INRA*[4]*, ou une joueuse (voire un joueur) à l'entraînement de rugby.*

Patricia, merci pour tout ce que tu m'as apporté. Je pense avoir beaucoup appris au fil des ans grâce aux discussions que nous avons eues. Merci pour ton écoute, ton aide, ton énergie et ton mental. Courage pour la suite.

Je remercie Claude Thermes et Alain Viari, mes rapporteurs, pour avoir accepté (et accompli !) la tâche de lecture et de critique du manuscrit, malgré la charge de travail qu'ils avaient par ailleurs. Merci également à Gwennaële Fichant, Roderic Guigó et Michel Weber de s'être intéressé à mon travail. Merci à tous de m'avoir fait l'honneur d'un tel jury. Je remercie aussi Pierre Rouzé et Dominique Cellier pour avoir participé à mon comité de thèse INRA.

À l'image de certains sports d'équipe, certaines parties d'une thèse relèvent de l'effort collectif. Je remercie donc chaleureusement l'ensemble des membres de l'unité BIA pour leur accueil, leur disponibilité, le cadre de travail et l'ambiance en général. Merci particulièrement à l'équipe administrative (Pascale, Jacky, Maïthé et Monique), à l'équipe informatique (Abde, Martin, Patrick), et à celles et ceux qui ont fait l'effort de s'intéresser à mon travail (leur aide lors des répétitions de la soutenance a été décisive). Évoquer l'ambiance ne pourrait se faire sans saluer les non-titulaires de l'unité, dont le cru "stagiaires" de l'été 2004 notamment fut d'excellente qualité. Merci à eux pour cette bouffée de bonne humeur et de bon esprit (mention spéciale aux

[4] alors Régis, le doublé pour 2005 ?

photos "conceptuelles" et à la page intranet de Sylvain B. qui est entrée dans les annales[5]*). Merci à Iadine pour ses conseils, Matthias pour ses coups de main. J'en profite pour souhaiter (entre autres) bon courage aux thésards de l'unité à qui je passe le relais (et pour qui je ne me fais pas trop de souci*[6]*) et plus largement à tout(e) lecteur(-trice) désirant soutenir sa thèse un jour et qui ne bénéficie pas des conditions favorables que j'ai eu la chance d'avoir.*

Je souhaite exprimer une reconnaissance particulière envers les membres de l'équipe EuGène*, avec qui j'ai eu beaucoup de plaisir à collaborer et sans le travail desquels je n'aurais pas terminé à l'heure où j'écris ces lignes : merci donc à Philippe (Bardou) pour l'énorme travail de développement, notamment en ce qui concerne la réorganisation de l'architecture logicielle (vivent les* "Plugins" *!) et le site web d'*EuGène'Hom*, à Marie-Jo (-sée Cros) pour avoir pris en charge la suite du processus d'optimisation en achevant son intégration au sein du programme et pour son apport de rigueur ("et la doc ?. . . "), à Jérôme (Gouzy) pour son dynamisme, son efficacité et son regard critique (qui me fait penser à la première note de bas de page au sujet de la différence entre la théorie et la pratique), et à Annick (Moisan) pour sa finesse et sa culture. Merci à tous de ne m'avoir pas tenu rigueur de mes quelques mises en retrait par rapport au projet, surtout la dernière année. Merci aussi à l'équipe "perpignanaise" (Richard Cooke, Anne-Marie Henry, Benoit Piégu).*

Par souci de clarté pluridisciplinaire, le premier chapitre de ce document ("Introduction") fut relu par un grand nombre de courageux volontaires, que je n'énumérerai pas ici mais que je tiens aussi à remercier.

Du côté de la fac, merci aux "drosos-philes" David, Christian ("c'est beau la pédagogie !"), Corinne et Pierre, à Catherine, à Barou ("li biti bitou, li ma biti ma bitou !") ; et merci à Christel, de l'école doctorale, pour son amabilité.

Pour finir avec les équipes, je ne pourrais omettre de remercier celles et ceux qui m'ont permis de partager régulièrement le plaisir du ballon ovale ces trois dernières saisons, Régis en tête bien sûr (avec son esquive rotative et sa désormais célèbre passe entre les jambes[7]*) pour avoir permis tout cela (et surtout pour avoir démontré qu'il est possible d'être de plus mauvaise foi que moi !), mais aussi (en vrac) Patricia (et ses bonnes excuses, dont l'imparable "j'ai oublié mes chaussures"), Anne, Aurélie (merci pour les chevreuils !), Mélanie, Jean-Marc (et le coup de la ceinture toujours trop serrée pour mon tour de ventre), Martin (et sa passe à une main*[8]*), Gérald (profondeur exemplaire en attaque), le trio Stéphane, Arnaud et Baba ("la prochaine fois je viens !"), Catherine, Stef, Laurent, Matthias (au cadrage-débordement fulgurant), Simon, Olivier, Rémi, les Ragondins du Soleil Levant (et même leur arbitre 32), le clan familial (une abondante source de bonheur et de fierté pour moi), et tou(te)s les autres. . . sans oublier bien sûr nos anciens piliers désormais expatriés (mais dont jamais au grand jamais le trou dans la défense n's'est refermé) : Pierre-Cyril (magicien du recrutement) et Øyvind (les tactiques annoncées en norvégien fonctionnent encore !).*

[5] il n'y a point de suspension. . .

[6] attention toutefois à ce que la thèse ne devienne pas un embêtement !

[7] pas toujours très précise toutefois. . . :-)

[8] même remarque que ci-dessus.

Enfin, je remercie ma famille plus ou moins proche (de ma mère pour son soutien et sa "créativité maximale" aux "cousings" d'outre-Atlantique[9]*, en passant par YoB, mon mononc pis mes matantes...) et les amis qui m'ont proposé et/ou apporté de l'aide (JB, là !). Pour terminer, j'associe à la gratitude que j'éprouve (entre autres) pour ma chérie à qui je dois beaucoup sur le plan personnel toutes mes félicitations pour m'avoir supporté pendant ces années, et j'en profite d'ailleurs pour l'encourager à persévérer dans cette voie...*

Merci à ceux qui m'ont aidé et que j'ai oubliés, et aux personnes qui poursuivront la lecture du manuscrit au-delà de cette partie[10] *pour s'attaquer au contenu scientifique, certes moins amusant (pour autant que le fussent ces quelques lignes), mais qui ne s'en trouve pas moins intéressant, du moins je l'espère.*

Il y a plus de courage que de talent dans la plupart des réussites.

— Félix Leclerc

You want to know how I did it? This is how I did it, Anton : I never saved anything for the swim back.

— Vincent

(Bienvenue à Gattaca, 1997)

(Après avoir manifesté sa gratitude et sa joie en s'agenouillant aux pieds de Martin Scorsese et en embrassant les membres du jury) *Mesdames et monsieur!... Mon coeur est époustouflant de ce moment! Et j'ai pas les mots pour vous donner tout mon amour! Et j'ai gagné la palme d'or*— (se tournant brusquement en montrant son trophée) *qu'est-ce que c'est?...*

— Roberto Benigni

(recevant le Grand Prix du Jury au festival de Cannes, 1998)

[9] *"Nous ne voulons pas être une province "pas comme les autres", nous voulons être un pays comme les autres."* (Pierre Bourgault)

[10] les dernières études démontrant que les remerciements constituent la partie d'une thèse qui bénéficie du rapport $\frac{ML}{CS}$ le plus élevé, où ML représente le nombre moyen de mots lus et CS l'importance du contenu scientifique.

Cette thèse est dédiée à ses lectrices et lecteurs.

Chapitre 1

Introduction

I Le vivant à l'échelle moléculaire

1/ Avertissements

Cette partie a pour objectif de présenter les bases élémentaires de biologie moléculaire nécessaires à la compréhension de ce manuscrit pour un non-spécialiste du domaine.

Le premier avertissement s'adresse au lecteur biologiste, qui pourra trouver ce qui suit très réducteur quant à la complexité et la richesse des phénomènes biologiques sommairement décrits. Des contraintes de longueur ont en effet rendu nécessaires certaines simplifications.

Le deuxième avertissement s'adresse au lecteur non familiarisé avec le domaine de la biologie. Il faut savoir que contrairement aux mathématiques et à l'informatique, la biologie est le domaine de l'incertain, où toutes les règles ont des exceptions[1], et où aucune expérience n'est totalement déterministe ni reproductible, ce qui peut déstabiliser certaines personnes habituées aux sciences exactes.

Enfin, une dernière précision : dans ce chapitre est présentée la relation entre le domaine du vivant et les mécanismes moléculaires de l'expression génique. Il ne s'agit là bien entendu que d'une façon de mettre en valeur l'intêrêt suscité par la discipline, car il serait indéniablement réducteur de considérer le vivant uniquement d'un point de vue "tout-génétique". La nature de tout être vivant est le fruit de l'interaction permanente entre l'inné et l'acquis, qui ne sauraient être dissociés.

[1] celle-ci comprise....

2/ L'ADN génomique

Tout être vivant est constitué d'une ou plusieurs cellules. La cellule, unité de base de la vie, contient l'ensemble de l'information génétique appelé génome qui caractérise l'individu et son espèce. La présence d'un noyau, compartiment spécifique renfermant le génome, permet de distinguer dans la classification du vivant les organismes eucaryotes, à cellule(s) nucléée(s), des organismes procaryotes qui en sont dépourvus. Le support moléculaire de l'information génétique est l'acide désoxyribonucléique, ou ADN.

L'ADN est une macromolécule dont la structure est connue depuis 1953, grâce à la célèbre publication de J. Watson et F. Crick basée notamment sur les travaux de R. Franklin et M. Wilkins (Watson et Crick, 1953). L'ADN génomique se présente sous la forme d'une hélice composée de deux brins (Fig. 1.1). Chaque brin est constitué d'un enchaînement linéaire et orienté d'éléments appelés nucléotides, dont la formule chimique se compose de trois parties : un acide phosphorique, un sucre (un 2'-désoxy-D-ribose) et une base azotée ou base (Fig. 1.2 ci-contre). On distingue pour l'ADN quatre types de nucléotides représentés par les lettres A, C, G et T qui ne diffèrent que par leur base, respectivement une adénine, une cytosine, une guanine et une thymine. La taille d'un brin étant fonction du nombre de nucléotides et donc de bases présents dans la chaîne, l'unité de longueur de la molécule d'ADN est la paire de base (pb) ou plus simplement la base (b).

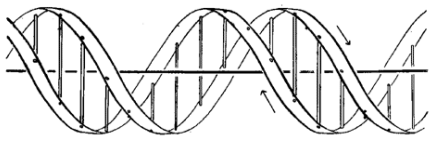

FIG. 1.1 – Structure schématique de la double hélice de l'ADN. Les chaînes nucléotidiques sont représentées par les deux rubans et les appariements entre les bases par les barreaux verticaux, le trait horizontal représentant l'axe de l'hélice. Image originale de la publication de Watson et Crick (1953).

Dans un brin d'ADN, un nucléotide est lié au précédent par son atome de carbone appelé 5' (d'après la numérotation conventionnelle des atomes de carbone composant le squelette du sucre) et au suivant par son carbone 3'. La liaison internucléotidique est une liaison covalente[2] appelée phosphodiester[3]. Le premier nucléotide de la chaîne présente donc un carbone 5' qui n'est pas engagé dans une telle liaison. L'extrémité correspondante est appelée 5' et l'autre l'extrémité 3', ce qui définit un sens pour tout brin d'ADN.

Les deux brins composant l'hélice de l'ADN génomique sont "antiparallèles", car ils sont parallèles mais en orientations 5' → 3' opposées (Fig. 1.1). Leur association est rendue possible grâce aux appariements entre leurs bases respectives, chaque base de l'un pouvant s'associer à la base lui faisant face, et ce par des liaisons de type hydrogène, plus faibles que les liaisons cova-

[2]type de liaison chimique forte.

[3]en raison des deux liaisons nécessaires de type ester reliant l'acide phosphorique et le sucre, avec élimination d'une molécule d'eau.

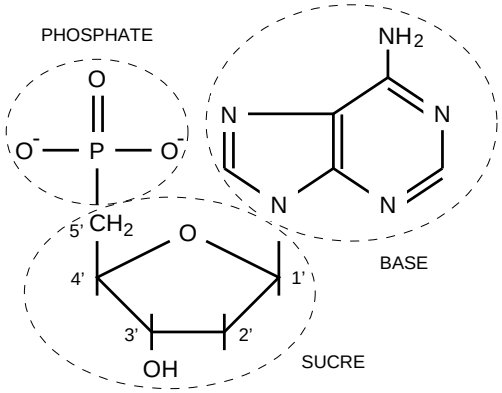

FIG. 1.2 – Structure schématique d'un nuléotide isolé (ici une désoxy-adénosine 5'-phosphate). Les pointillés indiquent les 3 composants le constituant, à savoir le sucre ou ose en bas, le groupement phosphate en haut à gauche, et la base à droite (ici une adénine). Les chiffres $1' \ldots 5'$ numérotent les atomes de carbone du sucre. À noter l'absence de groupement hydroxyle `OH` en 2', présent en 3'.

lentes. Les paires de bases ainsi formées impliquent majoritairement soit une adénine et une thymine (paire A-T), soit une guanine et une cytosine (paire G-C) : ce type d'appariement appelé "Watson-Crick" définit la complémentarité entre deux bases ou deux brins.

Par conséquent, on peut représenter l'information génétique contenue dans la totalité ou une partie d'une molécule d'ADN donnée par une succession ou séquence de lettres `A`, `C`, `G` et `T` identifiant les bases successives de l'un des brins de la double hélice. Par convention, une séquence d'ADN représente les bases de 5' vers 3' dans le sens de lecture. La séquence de l'autre brin se déduit sans ambiguïté par complémentarité (et inversion). Par exemple, la séquence génomique `GATTACA` fait face à la séquence `TGTAATC` qui est dite inverse-complémentaire.

Remarque 1 *Lorsque l'on représente l'information d'une molécule d'ADN par une séquence de lettres {`A`,`C`,`G`,`T`}, on dit que le brin contenant les bases indiquées est le brin sens (*"forward"*), l'inverse-complémentaire étant l'anti-sens (*"reverse"*).*

3/ Gène et expression génique

La fonction fondamentale de tout génome est de maintenir et transmettre l'information génétique, portée par certaines régions appelées gènes. Le gène est l'unité fonctionnelle universelle de l'information héréditaire. Concrètement, il s'agit d'une partie d'une molécule d'ADN génomique nécessaire à la synthèse dans la cellule d'un produit biologique fonctionnel, généralement[4] une protéine. Sans chercher à minimiser l'importance des autres catégories de gènes, nous ne considèrerons dans le cadre de ce travail que les gènes codant pour des protéines.

Les protéines sont les acteurs essentiels de tout processus biologique complexe ; elles interviennent aussi bien au niveau des structures fondamentales que des principales activités biochimiques du vivant (on peut citer par exemple les enzymes, les anticorps, ou certains pigments comme la chlorophylle ou l'hémoglobine). Comme l'ADN, une protéine est polymérique,

[4] si l'on considère le nombre de gènes identifiés à ce jour.

c'est-à-dire qu'elle est composée d'un assemblage de plusieurs éléments, qui sont ici des acides aminés. La nature d'une protéine, et par conséquent sa fonction, est déterminée par sa séquence d'acides aminés, elle-même étant déterminée par la séquence nucléotidique du gène correspondant. Le processus cellulaire de production d'une protéine à partir de son gène s'appelle l'expression génique.

L'expression génique est le mécanisme moléculaire que l'on trouve à la base de toute vie connue à ce jour. Tous les organismes vivants ont en commun ce phénomène, qui assure la synthèse protéique à partir de l'information génétique, avec quelques variations selon les espèces. Ici ne sera considérée que l'expression génique eucaryote, qui concerne tous les êtres vivants à l'exception des bactéries.

Chez les eucaryotes, l'essentiel de la fabrication des protéines a lieu dans le cytoplasme (à l'extérieur du noyau cellulaire). Par conséquent, l'information génétique portée par l'ADN doit y être acheminée. Le vecteur moléculaire utilisé à cette fin s'appelle l'ARN messager. L'expression génique se décompose en trois étapes majeures comprenant la synthèse d'un précurseur d'ARN messager à partir de l'ADN génomique (transcription), sa maturation en ARN messager, et la construction de la protéine (traduction) à partir du messager (Fig. 1.3 ci-contre). Ces étapes de l'expression génique, brièvement décrites ci-dessous, constituent le dogme central de la biologie moléculaire.

a. La transcription

La transcription correspond à l'étape de lecture et de copie de l'information génétique contenue sur une partie de la molécule d'ADN génomique par synthèse d'une molécule d'acide ribonucléique ou ARN.

Dans un premier temps, le complexe de transcription, ensemble de molécules comprenant l'enzyme dénommée d'ARN polymérase, se fixe sur l'ADN grâce à des interactions biochimiques impliquant le promoteur, une région génomique située dans la partie initiale du gène. Puis, ce complexe se déplace le long du gène en suivant un des deux brins de la double hélice. Au fur et à mesure de sa progression, il construit une molécule formée d'un brin d'acide ribonucléique ou ARN par polymérisation, en assemblant des nucléotides libres les uns à la suite des autres.

L'ARN diffère de l'ADN par la nature de ses nucléotides dont le sucre (le D-ribose) possède un groupement hydroxyle `-OH` complet en position 2' au lieu d'un simple atome d'hydrogène comme dans celui de l'ADN (le 2'-désoxy-D-ribose). En outre, si l'ARN contient également comme bases l'adénine (A), la cytosine (C) et la guanine (G), l'uracile (U) remplace en revanche la thymine (T) de l'ADN.

La polymérisation de la molécule d'ARN par le complexe de transcription s'effectue de façon à ce que la séquence des bases du brin d'ARN synthétisé soit la copie de la séquence d'un des brins d'ADN (aux substitutions `T`→`U` près). Arrivée à la fin du gène (dont la taille peut aller de quelques centaines à quelques millions de bases), l'ARN polymérase se décroche de l'ADN génomique, le gène est alors transcrit. La molécule d'ARN obtenue est appelée

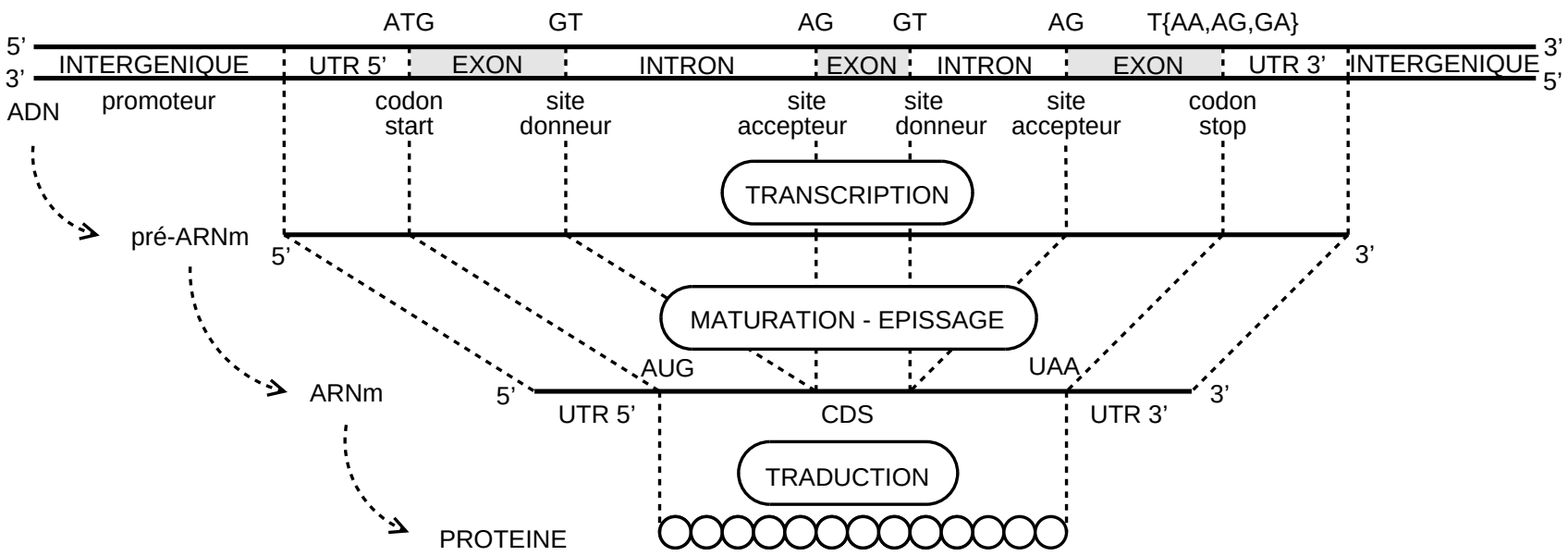

FIG. 1.3 – Schéma de l'expression génique eucaryote. Les lignes épaisses représentent les brins d'ADN (en haut) et d'ARN (dessous). Sont indiqués à leurs extrémités les repères pour l'orientation, au dessus la séquence de certains motifs fonctionnels, et dessous le nom de ces motifs ou des éléments qu'ils délimitent. L'enchaînement de cercles (en bas) symbolise la chaîne d'acides aminés constituant la protéine. CDS = séquence codante (dont la longueur est multiple de 3). À noter que les molécules sont représentées linéaires mais qu'aucune ne l'est *in vivo* et que seules les parties codantes sont considérées comme appartenant aux exons (comme défini à la page 9). Ne pas tenir compte des proportions.

transcrit primaire, précurseur d'ARN messager, ou encore pré-ARNm. Elle passe par une étape de maturation avant de sortir du noyau.

Remarque 2 *Le brin dont la séquence est identique (avec T=U) à celle du pré-ARNm définit le sens du gène. S'il s'agit du brin sens, pour une séquence génomique donnée (toujours présentée de 5' vers 3' par convention), le gène est transcrit de gauche à droite. Dans le cas contraire, on dit que le gène est sur le brin anti-sens et la séquence transcrite est l'inverse-complémentaire de la génomique (avec T→U).*

b. La maturation

La maturation correspond à l'étape de modification du précurseur d'ARN messager pour obtenir un ARN messager mature.

Diverses interventions sont réalisées sur le pré-ARNm. Citons par exemple le coiffage (ou *"capping"*) et la poly-adénylation, qui protègent d'une dégradation précoce par certains enzymes de la cellule le brin d'ARN en ajoutant respectivement une coiffe[5] à l'extrémité 5' et une queue poly-A[6] à l'extrémité 3'. Dans le cadre de notre problématique, le phénomène qui nous intéresse plus particulièrement s'appelle l'épissage.

L'épissage (ou *"splicing"*) est le processus qui consiste à exciser certaines parties du pré-ARNm appelées introns en conservant et raboutant entre elles les autres parties appelées exons (Fig. 1.3 page précédente). Cette opération est réalisée par le spliceosome, un complexe moléculaire d'épissage composé de protéines et de petits ARN, qui interagit de façon précise avec les régions séparant les exons des introns : les sites d'épissage. On en distingue deux sortes : entre la fin d'un exon et le début d'un intron (dans l'orientation conventionnelle de la molécule de 5' vers 3') se trouve le site donneur d'épissage, et entre la fin d'un intron et le début de l'exon suivant se trouve le site accepteur d'épissage. Ils sont caractérisés par une séquence de bases plus ou moins conservée selon la position dans le site et l'espèce considérée. Ainsi, la séquence d'un intron standard commence par `GT` côté 5' (site donneur) et se termine par `AG` côté 3' (site accepteur).

Remarque 3 *Cependant, tous les gènes ne disposent pas d'une structure exon-intron unique. En effet, un mécanisme permet à certains de produire plusieurs types d'ARNm différents à partir d'un même type de précurseur : il s'agit de l'épissage alternatif. En fonction de divers facteurs de régulation, le complexe d'épissage peut alterner le choix d'un ou plusieurs sites donneurs et/ou accepteurs. Il résulte de cette variation une augmentation de la diversité des produits des gènes. Ce phénomène complexe, autrefois considéré comme exceptionnel, apparaît aujourd'hui comme un acteur majeur de la ré-*

[5] il s'agit exactement d'un nucléotide qui porte une guanine méthylée sur son azote en 7 et qui se lie au premier nucléotide du pré-ARNm par une liaison anhydride d'acide atypique 5'→5'.

[6] constituée simplement d'une succession de nucléotides à adénine.

1re position (5')	2e position U		2e position C		2e position A		2e position G		3e position (3')
U	UUU	PHE	UCU	SER	UAU	TYR	UGU	CYS	U
	UUC		UCC		UAC		UGC		C
	UUA	LEU	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG		UGG	TRP	G
C	CUU	LEU	CCU	PRO	CAU	HIS	CGU	ARG	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	GLN	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	ILE	ACU	THR	AAU	ASN	AGU	SER	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	LYS	AGA	ARG	A
	AUG	MET	ACG		AAG		AGG		G
G	GUU	VAL	GCU	ALA	GAU	ASP	GGU	GLY	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	GLU	GGA		A
	GUG		GCG		GAG		GGG		G

TAB. 1.1 – Le code génétique. ALA : alanine, ARG : arginine, ASN : asparagine, ASP : aspartate, CYS : cystéine, GLN : glutamine, GLU : glutamate, GLY : glycine, HIS : histidine, ILE : isoleucine, LEU : leucine, LYS : lysine, MET : méthionine et codon start, PHE : phénylalanine, PRO : proline, SER : sérine, STOP : codon stop, THR : thréonine, TRP : tryptophane, TYR : tyrosine, VAL : valine.

gulation de l'expression génique, ce qui pose certains problèmes qui feront l'objet du chapitre 6 de ce document.

Pour conclure sur la maturation, on peut considérer que l'ARNm ou transcrit (mature) est le résultat de l'assemblage bout à bout des exons du pré-ARNm. Il traverse alors la membrane séparant le noyau du reste de la cellule pour atteindre le cytoplasme, siège de la traduction.

c. La traduction

La traduction est l'étape de fabrication de la protéine à partir de l'information portée par l'ARN messager.

Comme décrit plus haut, une protéine est un polymère d'acides aminés dont la séquence est déterminée par la séquence de nucléotides de l'ARNm. Comme il existe vingt acides aminés différents et seulement quatre nucléotides possibles, la traduction repose sur un code qui associe un acide aminé à chaque triplet de nucléotides successifs de la séquence. Ce code est appelé code génétique (Tab. 1.1) et les triplets de nucléotides sont les codons.

Le code génétique possède des propriétés intéressantes. Il est universel, c'est-à-dire que toutes les espèces vivantes utilisent le même (à quelques variations près). De plus, il est déterministe, car à un codon donné ne correspond qu'un seul acide aminé sans ambiguïté. Enfin, il est dégénéré ; en effet, l'utilisation de triplets avec un alphabet de 4 lettres permet 64 codons différents, et ceci pour seulement 20 acides aminés, ce qui autorise une redondance. Plusieurs triplets peuvent donc coder pour le même acide aminé (on parle de codons synonymes), et pour une espèce donnée on peut observer

une utilisation préférentielle de certains au détriment d'autres. Ce phénomène s'appelle le biais d'usage du code et sera abordé ultérieurement lors de l'étude statistique des exons. L'observation du tableau indique que les codons synonymes diffèrent souvent par leur 3^{e} base.

Le complexe moléculaire responsable de la traduction porte le nom de ribosome, il est constitué d'un ensemble de protéines et d'ARN. À partir de l'extrémité 5' de l'ARNm, il parcourt la molécule jusqu'à ce qu'il atteigne une position portant le signal de démarrage de traduction. Celui-ci contient un codon particulier appelé le codon start : `AUG`. À partir de là, le ribosome progresse par décalages successifs sans chevauchement de triplets en triplets, en associant à chaque codon l'acide aminé correspondant, et en assemblant ce dernier au précédent par une liaison covalente peptidique. La chaîne protéique est ainsi constituée, jusqu'à l'occurrence d'un des trois codons particuliers signalant la fin de la traduction appelés codons stop : `UAA`, `UAG` et `UGA`[7]. La partie de l'ARNm comprise entre les codons start et stop est appelée la CDS (pour *"Coding (DNA) Sequence"*), et les régions flanquantes de part et d'autre les UTR5' (avant le start) et UTR3' (après le stop) car elles sont transcrites mais non traduites (UTR pour *"Untranslated (Terminal) Region"*). Les codons stop sont les seuls à n'avoir pas d'acide aminé associé[8] (Tab. 1.1 page précédente) et provoquent le décrochement du complexe de traduction et donc la libération de la protéine ainsi constituée.

Remarque 4 *Pour une séquence génomique quelconque, il est aisé de procéder à une traduction virtuelle. À partir d'une position initiale i, il suffit de lire la séquence triplet par triplet et d'associer à chaque codon l'acide aminé indiqué par le code génétique, ce qui donne une séquence d'acides aminés. Deux autres séquences peuvent être obtenues en décalant la position initiale (pour* $i+1$ *et* $i+2$*), qui détermine la phase de lecture. Enfin, on peut générer trois autres séquences protéiques en traduisant dans le sens opposé la séquence de l'autre brin, ce qui permet en tout 6 phases de lecture possibles pour une séquence d'ADN génomique. Dans la suite du document, nous emploierons le terme de traduction aussi bien pour désigner la traduction* in vivo *que pour la traduction virtuelle. Il est également à noter que la CDS étant constituée de triplets, sa longueur est naturellement multiple de trois, ce qui n'est pas forcément le cas des exons la composant.*

d. Conclusion

Au niveau de l'ADN génomique, on peut donc considérer la structure d'un gène eucaryote comme une mosaïque d'exons intercalés d'introns, le tout flanqué par des UTR (5' et 3') et séparé des gènes voisins de part et d'autre par des régions intergéniques. Comme nous nous intéressons aux parties codantes du gène (constituant la CDS), nous utiliserons par la suite le terme d'exon uniquement pour faire référence aux parties codantes des

[7] appelés aussi ocre, ambre et opale, respectivement.

[8] sauf de rares exceptions comme la sélénocystéine et la pyrrolysine, pour lesquelles il a été montré qu'elles pouvaient s'associer aux codon `UGA` et `UAG`, respectivement.

exons (Fig. 1.3 page 5). Cette nuance par rapport à la définition biologique implique que le premier exon d'un gène débute par un codon start (au lieu du premier nucléotide transcrit) et se termine soit par un codon stop (gène sans intron) soit par un site donneur d'épissage. Quant à l'exon terminal, il s'achève par un des trois codons stop.

Par conséquent, si l'on dispose d'une séquence génomique contenant un gène dont on voudrait identifier la protéine pour laquelle il code, il suffit de déterminer la position de début et de fin de tous les exons composant ce gène. Une simple lecture de ces régions codantes donne la CDS, qu'il est trivial de traduire en protéine en utilisant le code génétique.

II Contexte historique et motivations

1/ De la génétique à la post-génomique

Ce n'est que près d'un siècle après la découverte en 1866 des premières lois de la génétique par Gregor Mendel que le dogme central de la biologie moléculaire fut établi. Mais le formidable essor de la biologie moléculaire ne vit le jour que dans les années 1970 grâce à des innovations d'ordre technologique. En 1977 notamment, Frédéric Sanger met au point une technique de séquençage de l'ADN[9]. Elle permet d'identifier la séquence de bases portée par la molécule d'ADN, et donc d'accéder à l'information génétique. Bien qu'améliorée depuis, c'est encore cette technique qui est utilisée aujourd'hui dans les projets internationaux de séquençage.

L'intérêt suscité par la possibilité d'accéder à l'information génétique est évident : l'expression génique étant le mécanisme à la base même de la vie, la compréhension d'un organisme vivant n'est pas envisageable sans la connaissance de son organisation génétique. Le séquençage d'un grand nombre de génomes est alors entrepris, ce qui donne naissance dans les années 1980 à une nouvelle discipline, consacrée au séquençage, à la cartographie et à l'analyse des génomes : la génomique.

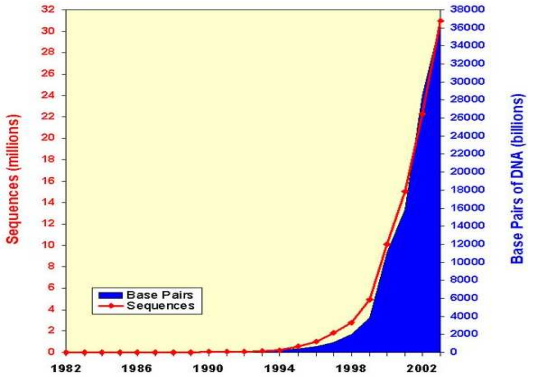

Fig. 1.4 – Croissance de la base de données publique Genbank (Benson *et al.*, 2004). La courbe et la zone foncée représentent respectivement le nombre de séquences déposées (échelle de gauche) et la quantité correspondante en nombre de paires de bases (échelle de droite).

[9] basée sur la capacité de didéoxynucléotides à interrompre une polymérisation d'ADN.

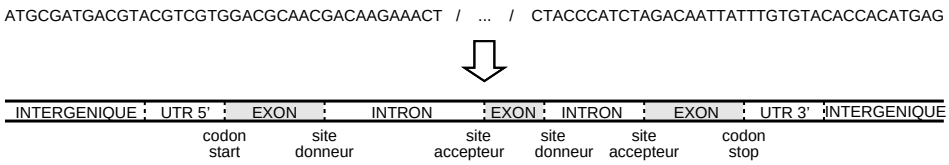

FIG. 1.5 – Objectif de la prédiction de gènes : à partir d'une séquence génomique brute (en haut), identifier sa structure génique (en bas).

Or, la quantité de données découvertes rend très vite incontournable l'utilisation de l'outil informatique, qu'il s'agisse de leur stockage, actualisation et partage avec l'apparition des bases de données publiques accessibles par l'internet ou de leur analyse avec les premiers algorithmes de recherche d'alignements locaux entre séquences. C'est ainsi au cœur de l'ère génomique que la bioinformatique fait son apparition.

La bioinformatique est une discipline à la frontière entre la biologie, les mathématiques, et l'informatique, qui a pour objectif de proposer des méthodes et des outils pour la gestion, l'analyse et l'exploitation des données issues de la biologie afin de produire de nouvelles connaissances. Puisqu'elle s'adresse à des problématiques spécifiques, elle définit une nouvelle branche de la biologie : après l'approche du vivant *in situ* (dans son milieu naturel), *in vivo* (dans l'organisme), et *in vitro* (en éprouvette), elle propose l'approche *in silico* (dans l'ordinateur[10]).

Avec la diffusion des premières séquences génomiques complètes dans les années 1990 se pose le problème de l'exploitation des données génomiques au point de faire apparaître le terme d'ère post-génomique. Aujourd'hui, avec plus de 200 génomes entièrement séquencés disponibles, le défi majeur auquel doit répondre la bioinformatique réside dans l'exploitation des données qui sont issues de la génomique et dont la quantité et la variété ne cessent d'augmenter (Fig. 1.4 page précédente).

2/ Du séquençage à l'annotation : la détection de gènes

Le séquençage du génome d'une espèce ne fournit qu'une information brute, soit une succession de lettres A, C, G et T. L'étape succédant au séquençage consiste à localiser et à caractériser les éléments fonctionnels présents dans le génome, il s'agit du processus d'annotation. Une des premières étapes de l'annotation est la localisation des gènes (Fig. 1.5).

Déterminer la structure des gènes, c'est-à-dire localiser la position de leurs éléments et surtout de leurs exons, permet d'identifier à partir de la séquence d'un génome l'ensemble des protéines potentiellement synthétisées par l'espèce considérée. Des logiciels de détection (ou prédiction) de gènes ont été créés pour répondre à ce besoin mais leur précision et leur capacité à intégrer les nouvelles données produites restent à améliorer.

[10] la plupart des puces d'ordinateurs étant à base de silicium.

III Objectif de la thèse

L'exploitation des données issues des grands projets de séquençage des génomes représente un enjeu majeur de la biologie moderne. L'objectif du travail présenté ici consiste à proposer et à mettre en application de nouvelles méthodes d'intégration des données aujourd'hui disponibles afin d'augmenter les performances et le champ d'application des logiciels de détection de gènes dans les séquences d'ADN génomiques. Cet objectif large sera détaillé dans les différents chapitres de la thèse :

Dans un premier temps, un état de l'art du domaine de la détection de gènes est proposé dans le chapitre 2. Les outils existants, les méthodes utilisées et les performances correspondantes y sont présentés, ainsi que les évolutions et tendances constatées dans le domaine et les principales limitations subsistantes. C'est de cette analyse que découle l'approche entreprise dans cette thèse, qui consiste à développer des méthodes d'intégration de données.

Le chapitre 3 présente le logiciel EuGène qui a servi de base pour toute la mise en application informatique des méthodes développées.

Le chapitre 4 décrit les améliorations apportées au logiciel qui ont permis d'envisager les différentes intégrations de données. Ces améliorations concernent le processus d'ajustement automatisé de paramètres par optimisation.

Le chapitre 5 présente la méthode d'intégration de données génomiques et sa mise en œuvre dans le logiciel. Cette partie décrit pourquoi et comment exploiter les séquences des génomes disponibles par le biais d'homologies inter- et intra-génomiques.

Enfin, le chapitre 6 est consacré à l'intégration des données issues des séquences transcrites. Un exemple concret concernant la prise en compte de nouveaux types de séquences est présenté, ainsi que la remise en question du modèle classique de gène provoquée par la considération du phénomène de l'épissage alternatif, devenu incontournable dans les projets d'annotation.

Une tentative de pénalité, c'est... trois points en suspension.

— Philippe Guillard
(Petits bruits de couloir, 1999)

J'ai toujours été persuadé — je le suis encore — que les diplômes sont fait pour les gens qui n'ont pas de talent. Malheureusement, il ne suffit pas de ne pas avoir de diplômes pour avoir du talent.

— Pierre Desproges
(Chroniques de la haine ordinaire, inédit)

Chapitre 2

État de l'art de la prédiction de gènes

La prédiction de gènes, appelée aussi détection de gènes, localisation de gènes ou identification de structure de gènes, a pour objectif d'identifier l'ensemble des protéines potentiellement produites à partir d'une séquence d'ADN génomique (nous n'aborderons pas ici le cas des gènes d'ARN non-codants). La séquence étant représentée par une succession ordonnée de lettres {A,C,G,T}, l'objectif est de déterminer pour chaque gène de la séquence la position de début et de fin de chacun de ses exons (Fig. 1.5 page 10).

Or, le problème est d'une complexité imposante. En effet, le nombre de gènes potentiellement contenus dans une séquence augmente de façon exponentielle avec sa taille. Depuis une vingtaine d'année environ, des approches ont été explorées pour détecter les régions codantes dans les génomes via l'outil informatique. Des publications proposent régulièrement une mise à jour des connaissances du domaine (Fickett et Tung, 1992; Claverie, 1997; Reese *et al.*, 2000b; Mathé *et al.*, 2002; Zhang, 2002; Wang *et al.*, 2003; Brent et Guigó, 2004). Cependant, malgré la diversité des méthodes développées et le nombre impressionnant de logiciels créés, le problème reste toujours d'actualité.

L'objectif de ce chapitre, au travers d'une présentation de l'état de l'art du domaine de la détection et localisation de gènes *in silico*, est de mettre en évidence les limitations qui subsistent dans les méthodes développées jusqu'à présent, et de dégager un axe de travail à suivre afin d'apporter une contribution originale et efficace à cette problématique.

I Structure générale d'un prédicteur de gènes

En général, un logiciel de prédiction de gènes standard présente la structure schématisée en Fig. 2.1 page suivante. Il prend en entrée une séquence

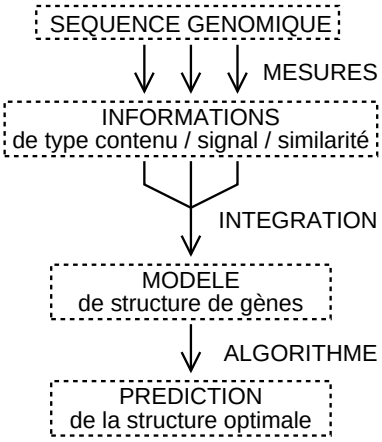

FIG. 2.1 – Fonctionnement schématique d'un logiciel de prédiction de gènes, décrivant les principales étapes du processus. À partir de la séquence génomique à annoter, des analyses sont réalisées pour mesurer diverses informations. Puis, ces informations sont intégrées au sein d'un modèle, probabiliste ou non, qui représente l'ensemble des structures de gènes possibles. Enfin, un algorithme, en général de programmation dynamique, permet d'identifier la meilleure structure de gènes possible.

d'ADN génomique (un fichier contenant une succession de lettres {A,C,G,T}), et produit en sortie une annotation, c'est-à-dire une structure de gènes (les positions sur la séquence génomique des exons prédits). Ce processus comprend trois étapes essentielles :

- mesures d'informations relatives à la séquence génomique. Les informations utilisées peuvent être de type *contenu, signal* ou *similarité*, comme nous allons le voir.
- intégration de ces informations dans un modèle de structure de gènes. Ce modèle, qui peut être probabiliste, a pour objectif de représenter l'ensemble des annotations possibles pour la séquence considérée.
- identification de la meilleure prédiction possible par un algorithme, en général de programmation dynamique.

Nous allons dans un premier temps présenter les différentes informations qui sont utilisées par les logiciels existants, puis dans un second temps les méthodes et les algorithmes associés. Enfin, nous proposerons une analyse générale de l'existant pour mettre en évidence les progrès nécessaires à ce domaine de recherche.

II LES INFORMATIONS UTILISÉES

1/ Les informations de type *contenu*

Les informations de type *contenu* permettent de refléter les différences statistiques de composition en bases entre les contenus des différents éléments géniques (exons, introns...), l'objectif étant de caractériser le plus précisément possible une "signature codante" pour les exons que l'on cherche à détecter.

Les premières tentatives de détection des régions codantes basées sur leurs propriétés statistiques datent du début des années 80, et exploitent deux caractéristiques principales des exons : la structure fondamentale périodique en codons (Fickett, 1982) de la séquence, et l'usage du code génétique (Staden et McLachlan, 1982) (pour un rappel sur le tableau du code génétique et ses

propriétés, voir page 7). Depuis, de nombreux travaux ont mis en évidence diverses informations statistiques dont certaines d'entre elles, utilisées par les logiciels de détection de gènes, sont présentées ci-dessous. Pour en avoir un aperçu exhaustif, le lecteur peut se référer aux publications de Fickett et Tung (1992) et de Guigó (1999)[1] ainsi qu'à la bibliographie régulièrement mise à jour par Wentian Li[2].

a. Première génération

i) Préférence en acides aminés

On ne retrouve pas les vingt acides aminés avec la même proportion dans les protéines. À partir d'un ensemble de protéines connues, on peut donc calculer des probabilités d'occurrences des différents acides aminés, ou des différents oligopeptides constitués par plusieurs acides aminés (McCaldon et Argos, 1988). Sur une séquence nucléotidique donnée, il suffit ensuite de procéder à une traduction virtuelle (voir la remarque 4 page 8) pour utiliser ces probabilités et obtenir ainsi une mesure du caractère codant.

ii) Préférence en codons

Nous avons vu que la structure du code génétique (Tab. 1.1 page 7) permet à plusieurs codons dits synonymes de coder pour le même acide aminé. Or, pour une espèce donnée, certains codons tendent à être "favorisés" par rapport à leurs synonymes. Il est donc possible à partir d'un jeu de données de gènes connus de l'espèce considérée de calculer les probabilités relatives de chaque codon synonyme, et d'utiliser cette information pour refléter le caractère codant d'une séquence donnée (Gribskov *et al.*, 1984).

iii) L'usage du code ou biais d'usage du code

Mis en évidence par Grantham *et al.* (1980), il se traduit par une préférence d'utilisation de certains codons dans les régions codantes par rapport aux régions non-codantes. Il peut être considéré comme la combinaison des 2 facteurs précédents, soient la préférence en acides aminés des protéines favorisant certains groupes de codons synonymes, et la préférence en codons qui pour un acide aminé donné favorise certains codons parmi les synonymes.

L'usage du code est utilisé pour la détection des régions codantes pour la première fois par Staden et McLachlan (1982), repris dans RECSTA (Fichant et Gautier, 1987) avec une méthode d'analyse des correspondances (AFC) puis dans les premiers logiciels de prédiction de gènes, comme GENEMODELER (Fields et Soderlund, 1990), SORFIND (Hutchinson et Hayden, 1992) ou GENEPARSER (Snyder et Stormo, 1995).

iv) L'asymétrie de composition

L'asymétrie de composition est une mesure basique reposant sur le fait que la composition en bases des séquences codantes varie selon la position

[1] http ://www.cbi.pku.edu.cn/mirror/gene/guigo99.pdf
[2] http ://www.nslij-genetics.org/dnacorr/

que l'on considère dans le codon. Ce biais de composition entre les positions des codons est une propriété fondamentale du codant.

Initialement exploitée dans l'algorithme TESTCODE (Fickett, 1982), l'asymétrie de composition est également reprise dans des prédicteurs de la première génération comme la méthode de Gelfand (1990) ou GRAIL (Uberbacher et Mural, 1991).

v) Fréquences de mots

Les différences de fréquences des mots de longueur k (ou k-uplets) entre les régions codantes et non-codantes peuvent également servir à détecter les exons. On peut considérer cette information comme une généralisation de l'usage du code, qui en est un cas particulier pour $k = 3$. L'algorithme TESTCODE, qui utilise le pourcentage en lettres `A,C,G,T`, correspond au cas particulier $k = 1$. L'utilisation des k-uplets pour la caractérisation du codant fut proposée la première fois par Claverie et Bougueleret (1986). En 1992, une étude comparative la considère comme la mesure la plus discriminante entre codant et non-codant (Fickett et Tung, 1992). Elle est généralement utilisée dans les prédicteurs de gènes avec $k = 6$ pour des fréquences d'hexamères, comme dans GRAIL (Uberbacher et Mural, 1991), GENEPARSER (Snyder et Stormo, 1995) ou FGENEH (Solovyev *et al.*, 1994).

b. L'avènement des chaînes et modèles de Markov

En 1993, une révolution se produit, initialement dans le domaine de prédiction de gènes procaryotes, grâce aux créateurs du logiciel GENEMARK (Borodovsky et McIninch, 1993) qui proposent des modèles de chaîne de Markov (ou plus simplement modèles de Markov) pour caractériser les contenus statistiques des différentes régions. Depuis, la majorité des prédicteurs de gènes reprennent le concept, avec parfois quelques modifications. Les modèles de Markov sont présentés notamment par Rabiner (1989) et par Durbin *et al.* (1999). Nous allons ici dans un premier temps définir les modèles de Markov, et présenter par la suite dans le cadre de l'étude des séquences nucléotidiques certaines de leurs applications, caractéristiques et problématiques intéressantes.

i) Définition

Définition 1

Une chaîne de Markov d'ordre k est une suite ordonnée $(x_i)_{i\in\mathbb{N}}$ de variables aléatoires à valeurs discrètes (dont l'ensemble est fini ou dénombrable) qui vérifie la propriété de Markov

$$P(x_i|x_{i-1}, x_{i-2}, \cdots, x_1) = P(x_i|x_{i-1}, x_{i-2}, \cdots, x_{i-k})$$

Cela signifie que la probabilité associée à la valeur de chaque variable ne dépend que de la valeur des k variables précédentes, k étant l'ordre de la chaîne.

Considérer une séquence nucléotidique (ou un ensemble de séquences) comme une chaîne de Markov revient à définir un modèle mathématique

$\mathbb{N}$	ensemble des entiers naturels
$\boldsymbol{X}$	ensemble des bases nucléotidiques tel que $X = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$
$\boldsymbol{x}$	une base telle que $x \in X$
$\boldsymbol{x_i}$	base portée par le nucléotide à la position i dans une séquence d'ADN
$\boldsymbol{W^k}$	ensemble des mots possibles de longueur k avec l'alphabet X
$\boldsymbol{w^k}$	un mot de longueur k tel que $w^k \in W^k$
$\boldsymbol{w_i^k}$	mot formé par les bases $(x_i, x_{i+1}, \cdots, x_{i+k-1})$ d'une séquence, soit le mot de longueur k dont la première lettre correspond à la base de la position i. Par extension, si $k = 1$ alors $w_i^k = x_i$
$\boldsymbol{(w^k x)}$ ..	mot w^k immédiatement suivi de la base x, avec $(w^k x) \in W^{k+1}$
$\boldsymbol{\#(w^k)}$ ..	nombre d'occurrences observées du mot w^k dans une ou plusieurs séquences données d'ADN
$\boldsymbol{P(w_i^k)}$ ..	probabilité d'observer w_i^k, le mot w^k à la position i d'une séquence
$\boldsymbol{P(x\|w^k)}$	probabilité conditionnelle d'observer la base x après tout mot w^k dans une séquence d'ADN

TAB. 2.1 – Notations utilisées

qui la caractérise, que nous appellerons modèle de Markov ou modèle markovien, pour lequel les variables aléatoires x_i prennent leur valeur dans $X = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. L'ordre du modèle permet la capture d'interactions à distance bornée.

Définition 2
Un modèle de Markov d'ordre k appliqué aux séquences d'ADN est entièrement défini par l'ensemble des probabilités

$$\begin{cases} P_0(w_1^k) & \textit{pour tout } w_1^k \in W^k \\ P(x|w^k) & \textit{pour tout } x \in X \textit{ et tout } w^k \in W^k \end{cases}$$

où w^k est un mot de longueur k (voir Tab. 2.1).

Ce qui signifie que le modèle est caractérisé par la probabilité initiale de chaque mot, et par la probabilité conditionnelle de chaque lettre en fonction du mot précédent. Le nombre total de probabilités définissant un modèle est donc de $4^k + 4^{k+1}$ (bien qu'en pratique, en raison des degrés de liberté, le nombre de paramètres soit inférieur à cette valeur). À noter que nous considérons uniquement le cas des modèles de Markov homogènes, où les probabilités sont les mêmes quelle que soit la position de la séquence.

Bien entendu, il ne s'agit pas ici d'affirmer que les séquences génomiques sont le produit de l'application d'un quelconque modèle mathématique, mais il s'avère que ce mode de représentation permet de caractériser de façon très satisfaisante l'information statistique qu'elles contiennent.

ii) Emission d'une séquence par un modèle

À partir d'un modèle markovien donné, il est possible d'émettre une séquence en suivant les probabilités du modèle. Il suffit de commencer par

choisir aléatoirement un premier mot selon les probabilités initiales, puis de générer chaque lettre en fonction du mot précédant et de la probabilité associée.

iii) Vraisemblance d'une séquence selon un modèle

VRAISEMBLANCE La ressemblance entre le contenu statistique d'une séquence et un modèle de Markov donné peut être quantifiée par une mesure appelée vraisemblance, qui correspond à la probabilité que la séquence soit émise par le modèle.

La vraisemblance d'une section S de la séquence comprise entre les positions i_1 et i_2 en fonction du modèle $M1$ s'exprime par

$$\boxed{P_{M1}(S) = P_{M1}(w_{i_1}^k) \prod_{i=i_1+k}^{i_2} P_{M1}(x_i | w_{i-k}^k)} \tag{2.1}$$

Par exemple, si l'on considère un modèle représentant le contenu statistique des exons d'une espèce, cette valeur calculée sur une séquence donnée est une approximation de la probabilité qu'elle soit codante.

RAPPORT DE VRAISEMBLANCE Pour mesurer la vraisemblance d'une séquence selon un modèle comparativement à un autre, on calcule en général le logarithme d'un rapport de vraisemblance. Pour une séquence ou section de séquence S et deux modèles M_1 et M_0, il s'agit de

$$Q(S) = \log \frac{P_{M1}(S)}{P_{M0}(S)} \tag{2.2}$$

cette valeur étant nulle si la séquence a une probabilité égale d'être émise par l'un ou l'autre des modèles, positive si la probabilité d'émission selon le modèle $M1$ est supérieure à celle qui correspond à $M2$, et négative dans le cas contraire.

Par exemple, si le modèle $M1$ caractérise les régions codantes du génome et $M0$ les autres, la valeur de $Q(S)$ est d'autant plus grande qu'il est préférable de considérer la séquence S comme codante plutôt que non-codante.

iv) Construction d'un modèle

Pour caractériser l'information statistique contenue dans un ensemble de séquences, on peut donc construire un modèle markovien. Pour cela, il faut disposer d'un jeu de données d'apprentissage, constitué de l'ensemble des séquences que le modèle doit représenter, et estimer les probabilités définissant ce dernier en appliquant la formule

$$P_0(w_1^k) = \frac{\#(w^k)}{\sum_{w' \in W^k} \#(w')} \tag{2.3}$$

où le numérateur $\#(w^k)$ est le nombre d'occurrences observées du mot w^k dans le jeu de données (Tab. 2.1), et le dénominateur désigne simplement le nombre total de mots de longueur k, et par la formule

$$P(x|w^k) = \frac{\#(w^k x)}{\sum_{x' \in X}(\#(w^k x'))} \quad \text{(en ignorant } w^k \text{ si } k = 0) \qquad (2.4)$$

où le numérateur $\#(w^k x)$ représente le nombre d'occurrences du mot w^k suivi de la base x, et le dénominateur le nombre d'occurrences du même mot suivi par une base quelconque.

Ces formules permettent de déterminer les valeurs des paramètres du modèle qui maximisent la vraisemblance des données observées dans le jeu d'apprentissage. Ainsi, on garantit que le modèle construit représente le plus fidèlement possible les séquences d'intérêt.

v) L'ordre

L'ordre est un paramètre capital d'un modèle de Markov, car son influence est primordiale aussi bien sur la finesse (et donc la précision du modèle) que sur la quantité de données nécessaire à une estimation convenable des paramètres.

Un modèle d'ordre $k = 0$ par exemple ne reflète rien de plus que les fréquences des quatres bases. Malgré sa simplicité, il contient toutefois une information non négligeable pour la détection des régions codantes. En effet, chez les eucaryotes supérieurs, il est connu que la somme des pourcentages des bases `G` et `C` (cette mesure s'appelle le $GC\%$) qui composent les exons est plus élevée que dans le reste du génome[3]. D'ailleurs, bien que la notion de chaîne de Markov n'y soit pas explicite, le programme TESTCODE (Fickett, 1982) exploite déjà les fréquences des 4 bases.

Des ordres supérieurs permettent de capter une plus grande dépendance entre les nucléotides successifs, et donc de modéliser plus d'information. On peut faire un parallèle avec la méthode basée sur la fréquence de mots. En effet, un modèle markovien d'ordre 2 pour le codant est comparable à la méthode basée sur l'usage du code, ou sur les fréquences de mots de longueur 3. Lors de l'introduction des modèles de Markov dans la détection de gènes (Borodovsky et McIninch, 1993), l'ordre retenu est de $k = 5$, ce qui offre une précision similaire à celle de la méthode de fréquences d'hexamères, considérée comme une des plus performantes (Fickett et Tung, 1992). Depuis, de nombreux logiciels utilisent des modèles de Markov pour caractériser le codant, majoritairement avec un ordre de 5 comme GENSCAN (Burge et Karlin, 1997), GENEID (Parra *et al.*, 2000) ou DAGGER (Chuang et Roth, 2001), ou de 4 comme HMMGENE (Krogh, 1997), SNAP (Korf, 2004) ou AUGUSTUS (Stanke et Waack, 2003).

Enfin, on peut noter que le bénéfice d'un ordre élevé apparaît moins évident pour les régions non-codantes, du fait des moindres contraintes exer-

[3] notons que du fait de la complémentarité entre les brins d'ADN qui assure par appariement le même nombre de base `G` que `C`, la mesure du $GC\%$ est indépendante du brin considéré.

cées sur leur séquence. On trouve par exemple pour les modèles intergéniques des ordres allant de 0 pour GeneID (Parra *et al.*, 2000) à 5 pour GenScan (Burge et Karlin, 1997), en passant par 3 dans HMMgene (Krogh, 1997) et 4 avec Augustus (Stanke et Waack, 2003).

vi) Intégration de la périodicité dans le modèle codant

Dans une chaîne de Markov dite homogène les probabilités sont les mêmes quelle que soit la position. Or, nous avons vu avec l'asymétrie de composition qu'il existait un biais de composition selon la position considérée dans les codons des exons. Afin de prendre en compte cette propriété des régions codantes, on peut utiliser des modèles markoviens hétérogènes 3-périodiques, comme introduits par Borodovsky et McIninch (1993) dans GeneMark. Cela revient à construire 3 modèles différents, un par position dans le codon, et à les appliquer de façon périodique pour une phase de lecture donnée. Ceci permet de prendre en compte de façon spécifique les différences entre les positions des codons qui sont conséquentes ; par exemple, la structure du code génétique fait que la deuxième base du codon ne dépend que de la nature de l'acide aminé correspondant, qui la détermine totalement (hormis pour la sérine, qui laisse deux possibilités), alors que la troisième base, plus libre, est classiquement corrélée au $GC\%$ la région génomique considérée. Actuellement, tous les prédicteurs de gènes exploitant les chaînes de Markov utilisent des modèles 3-périodiques pour le codant.

vii) Le problème du modèle intergénique

Les génomes contiennent plusieurs types d'éléments non-codants comme les introns, les UTR et les régions intergéniques. Si pour les deux premiers les données ne manquent pas pour peu que l'on considère un organisme ayant bénéficié d'un certain intérêt de la part de la communauté scientifique[4] et dont un nombre conséquent de gènes est déjà bien caractérisé par des méthodes expérimentales, les séquences intergéniques sont par nature difficiles à localiser. En effet, du fait même que tous les gènes ne sont pas connus, on ne peut affirmer sans risque d'erreur qu'une région située entre deux gènes n'en contient pas un troisième qui ne serait pas encore découvert !

Comment construit-on alors un modèle pour l'intergénique ? Si les publications décrivant les logiciels ne sont pas toujours explicites sur le sujet, trois méthodes principales sont envisageables :

- Estimer "en ligne" les paramètres d'un modèle d'ordre 0 directement sur la séquence à annoter. Si l'on fait par exemple un rapport de vraisemblance avec un modèle exonique, cela revient à comparer la séquence à un modèle codant par rapport à un modèle neutre, où la distribution des nucléotides est aléatoire mais en accord avec leurs proportions observées dans la séquence. Cette solution simple a l'avantage de n'exiger aucun jeu d'apprentissage, mais elle fait l'hypothèse que l'intergénique ne contient pas d'information exploitable sur des corrélations entre nucléotides adjacents. De plus, dès lors que la séquence contient un autre

[4] comme la mouche *Drosophila melanogaster* par exemple, ou le primate *Homo sapiens*.

type d'élément que de l'intergénique, le modèle ne peut être considéré comme représentant correctement de l'intergénique.

- Estimer les paramètres à partir des séquences intraniques. Cette possibilité pose moins de problèmes de quantité de données mais implique l'hypothèse que les introns sont similaires en composition et en fréquence de mots avec les séquences intergéniques (ce qui peut surprendre d'un point de vue biologique au vu des différences entre les éléments fonctionnels présents dans les deux types de région). On trouve cette stratégie par exemple dans GeneID (Parra *et al.*, 2000) et dans Augustus (Stanke et Waack, 2003).

- Estimer les paramètres à partir des parties supposées intergéniques des séquences répertoriées dans les bases de données.

  On peut les obtenir en extrayant les régions situées entre deux gènes adjacents connus. Ainsi, on obtient des séquences intergéniques "authentiques" et complètes. Toutefois, réduire le risque d'y inclure un gène complet non encore découvert (risque d'autant plus grand que les gènes sont éloignés) tend à inclure des séquences intergéniques plutôt courtes et dont on peut mettre en doute la représentativité. De plus, la quantité de gènes connus dont l'éloignement satisfait la contrainte de distance choisie peut être insuffisante.

  On peut également extraire les régions flanquantes des gènes connus en se limitant à une certaine longueur, ce qui pose deux problèmes : le premier concerne le côté de la frontière avec le gène, et découle de l'imprécision de la détermination des limites des régions transcrites des gènes connus. En effet, elle sont définies expérimentalement (ainsi que les frontières entre exons et introns) à partir des séquences des ARNm que l'on a réussi à isoler, séquencer, et aligner sur la séquence génomique. Etant donné que les procédés expérimentaux impliqués permettent rarement d'obtenir la totalité de la séquence d'un transcrit, on s'attend à ce que les séquences flanquant les gènes ainsi répertoriés contiennent une partie d'UTR, et donc que le modèle intergénique correspondant ne soit pas correct. Le deuxième problème des régions flanquantes concerne leur côté distal par rapport au gène, et donc le choix de leur longueur : jusqu'où peut-on considérer que l'on a uniquement affaire à de l'intergénique ? Le choix de cette limite de distance se base sur la distribution de longueur théorique des séquences intergéniques de l'espèce considérée. Un compromis se fait entre le risque d'inclure une partie de gène voisin et le défaut de représentativité de telles régions flanquantes qui par construction correspondent davantage à un modèle "bordure d'intergénique" qu'un modèle "intergénique" à proprement parler. Les logiciels Snap (Korf, 2004) et HMMGene (Krogh, 1997) par exemple utilisent des séquences intergéniques issues de bases de données.

La réalisation d'un modèle intergénique fiable reste un problème ouvert et malheureusement peu abordé dans la littérature. De plus, la diversité des

types d'éléments contenus dans les régions intergéniques (séquences répétées, promoteurs, régulateurs...) laisse entrevoir la nécessité de considérer plusieurs modèles distincts pour caractériser les différents contenus statistiques, ce qui n'a pas encore été réalisé à notre connaissance.

viii) Gestion de l'insuffisance de données

En fonction de l'ordre du modèle, la quantité de données disponibles pour estimer les paramètres peut s'avérer insuffisante. On sait en effet que le nombre de paramètres à estimer pour un modèle markovien croît de façon exponentielle avec l'ordre. Pour donner une idée, à partir d'un même jeu de données et donc de la même quantité d'information, un modèle d'ordre 3 par exemple contient 320 paramètres alors que 5120 caractérisent un modèle d'ordre 5. Or, chaque paramètre s'obtient par une division entre des nombres d'occurrences de mots, dont la longueur est fonction de l'ordre. Il faut donc que la quantité de séquences dont on dispose pour l'estimation permette l'observation d'un effectif suffisant en nombre de mots pour refléter les probabilités du modèle. Sinon, on se trouve dans un cas d'*"overfitting"*, où le modèle construit est trop spécifique du peu de données disponibles pour représenter l'ensemble du modèle idéal, et ne s'applique pas correctement à d'autres séquences. Plusieurs procédés existent pour dépasser cette limitation :

- La première possibilité, classique en estimation de probabilités, est d'introduire de l'*a priori*. Pour cela, on peut par exemple augmenter artificiellement la quantité de données observées par des pseudo-comptes. La façon la plus simple est d'ajouter uniformément un certain effectif à toutes les occurrences de mots observés. Le cas où cet ajout est égal à 1 correspond à la loi de Laplace, pratique pour éviter les occurrences nulles qui posent problème dans les calculs. Une méthode plus subtile consiste à ajouter une quantité dont l'influence s'ajuste en fonction de la quantité de données observées, ce qui à partir de la formule (2.4) de la page 19 donne

$$P(x|w^k) = \frac{\#(w^k x) + \Psi \cdot \psi_{(x|w^k)}}{\sum_{x' \in X}(\#(w^k x')) + \Psi}$$

  où le nombre Ψ est le poids du pseudo-compte et $\psi_{(x|w^k)}$ la proportion choisie vers laquelle on veut faire tendre $P(x|w^k)$ lorsque la quantité de données observées est faible par rapport à Ψ. L'inconvénient de cette méthode concerne précisément ce cas, car le modèle estimé ainsi tend alors vers un modèle artificiel, déterminé *a priori* de façon arbitraire. Les pseudo-comptes sont utilisés par exemple dans HMMGENE (Krogh, 1997) et dans AUGUSTUS (Stanke et Waack, 2003).

- Une autre possibilité est de restreindre le calcul des paramètres aux données dont on juge la quantité suffisante pour permettre une estimation robuste. Pour cela, il suffit de rendre variable l'ordre du modèle markovien, et de définir une valeur seuil d'observations nécessaires κ en

deçà de laquelle on utilise l'ordre inférieur. Formellement, la probabilité de trouver la base x juste après le mot w^k de longueur k est

$$P(x|w^k) = \begin{cases} \frac{\#(w^k x)}{\sum_{x' \in X}(\#(w^k x'))} & \text{si } \sum_{x' \in X}(\#(w^k x')) > \kappa, \\ P(x|w^{k-1}) & \text{dans le cas contraire.} \end{cases}$$

Plus élégante que la précédente, cette méthode ne dépend que d'un paramètre (la valeur seuil κ), n'affecte aucunement les probabilités estimées à partir de quantités suffisantes, et garantit une "protection" automatique contre le faible échantillonnage. Cette garantie est d'autant plus nécessaire que le modèle est hétérogène dans ses ditributions de probabilités, car dans ce cas le risque que certains mots apparaissent avec une fréquence faible voire nulle devient considérable. Cette approche, appelée modèles de Markov tronqués ou modèles de Markov de longueur variable (VLMM pour *"Variable Length Markov Models"*) représente un bon compromis entre la précision et la simplicité. Le logiciel AUGUSTUS (Stanke et Waack, 2003) utilise ce système de modèle de Markov à ordre variable, dont une variante est évoquée aussi par Bejerano (2004).

- Les modèles de Markov interpolés, ou IMM (pour *"Interpolated Markov Models"*), représentent une généralisation des modèles à ordre variable. Toujours à partir de la formule (2.4) de la page 19, un IMM définit la probabilité $P(x|w^k)$ comme une combinaison linéaire des probabilités associées aux mots de tailles inférieures à k contenus dans w_k par la formule récursive

$$P(x|w^k) = \zeta_{w^k} \cdot \frac{\#(w^k x)}{\sum_{x' \in X}(\#(w^k x'))} + (1 - \zeta_{w^k}) \cdot P(x|w^{k-1})$$

où ζ_{w^k} est le paramètre d'interpolation, avec $0 \leq \zeta_{w^k} \leq 1$. La détermination de ce paramètre peut se faire par diverses méthodes (Salzberg *et al.*, 1998b; Azad et Borodovsky, 2004). Le modèle précédant à ordre variable est un cas particulier d'IMM où ζ_{w^k} vaut 1 si $\#(w^k X) > \kappa$ et 0 sinon. La première application de modèles de Markov interpolés pour la détection de gènes fut destinée aux procaryotes avec le logiciel GLIMMER (Salzberg *et al.*, 1998b; Delcher *et al.*, 1999).

D'autres procédés proposent des solutions pour pallier le manque de données, comme les *"Mixture Transition Distribution"* MTD (Raftery et Berchtold, 2002) ou les modèles de Markov parcimonieux (Bourguignon et Robelin, 2004). Cependant, ces méthodes n'ont pas encore fait l'objet d'applications pratiques au sein de logiciels de prédiction de gènes.

ix) Les classes de GC%

Certains génomes présentent une composition en bases très hétérogène selon la région génomique considérée. Chez *Homo sapiens* par exemple, les variations du *GC%* mesuré dans une fenêtre glissante le long du génome

sont telles que l'on emploie le terme d'isochores pour désigner les régions homogènes en distribution nucléotidique.

Cette composition en mosaïque du génome humain pose des problèmes concernant l'application des modèles de Markov. En effet, il est difficile pour un modèle unique construit à partir de séquences issues de régions génomiques à $GC\%$ très différents de refléter les caractéristiques propres de chaque région, qu'il s'agisse d'un modèle du codant ou du non-codant. Fusionner des distributions fortement dissemblables implique une perte d'information et se fait donc au détriment de la précision.

C'est pourquoi les prédicteurs de gènes dédiés à *H. sapiens* adoptèrent très tôt une stratégie de type "diviser pour mieux régner", à l'image du logiciel GeneParser (Snyder et Stormo, 1995). Ainsi, il est courant de séparer les séquences du jeu de données d'apprentissage en différentes classes en fonction de leur $GC\%$, afin de construire autant de modèle de chaque type (exon, intron,...) qu'il y a de classes. Puis, pour mesurer l'affinité d'une séquence avec un type de modèle (comme avec la formule (2.1) page 18), il suffit d'utiliser celui qui appartient à la classe dont l'intervalle de $GC\%$ comprend le $GC\%$ de la séquence.

Bien que cette façon de faire soit courante (Snyder et Stormo, 1995; Salamov et Solovyev, 2000; Burge et Karlin, 1997), elle comporte certains points délicats. Tout d'abord se pose la question de la détermination du nombre et des valeurs frontières des classes de $GC\%$. Ensuite, partager les données entre diverses classes implique de réduire d'autant la quantité d'information disponible pour la construction de chaque modèle. On peut également s'interroger sur la pertinence de mesurer le $GC\%$ global d'une séquence, que ce soit à l'estimation des paramètres aussi bien qu'à l'évaluation de l'affinité, car les longueurs des séquences peuvent être considérablement différentes. Enfin, la distinction d'un nombre fini et généralement petit de classes (par exemple 4 pour GenScan) pose un problème de manque de continuité entre les modèles.

Quelques approches tentent de pallier ces problèmes. Fickett *et al.* (1992) proposent pour l'estimation de conserver un cloisonnement des données en classes, mais pour la prédiction d'appliquer pour la séquence à annoter un modèle construit par régression linéaire des deux qui appartiennent aux classes les plus proches du $GC\%$ de la séquence. Pour le logiciel Augustus, Stanke et Waack (2003) augmentent le nombre de classes (de 4 à 10) mais utilisent pour l'estimation d'un modèle appartenant à l'une d'entre elles l'ensemble des séquences, en les pondérant en fonction de la distance entre leur $GC\%$ et celui de la classe considérée. Cependant les longueurs des séquences ne sont apparemment pas prises en compte ni pour l'estimation, ni pour la prédiction. Apparemment, le seul logiciel ayant intégré des dépendances de modèles probabilistes envers le $GC\%$ par une approche continue est Grpl, qui utilise des rapports de vraisemblances suivant une fonction linéaire du $GC\%$ mesuré sur une fenêtre glissante (Hooper *et al.*, 2000).

c. Les longueurs

Les différents éléments constituant les gènes ne présentent pas les mêmes distributions de longueur. Celles-ci peuvent être estimées à partir de jeux de données, et servir ensuite à améliorer la qualité des prédictions. Nous reprendrons ce type d'information dans la partie algorithmique page 38. À noter que ces distributions sont très dépendantes de l'espèce considérée.

d. Périodicité du codant

D'autres méthodes se basent exclusivement sur la propriété de périodicité de structure des séquences codantes, qui se traduit par une corrélation périodique entre positions nucléotidiques. Sa mesure principale est la composition en bases déterminée sur les nucléotides qui ont la même position dans le codon pour une phase donnée (information capturée par modèles de Markov 3-périodiques). Certaines approches sont décrites par Fickett et Tung (1992) et par Guigó (1999).

Parmi les méthodes développées, on peut citer l'approche qui utilise la transformée de Fourier sur les séquences d'ADN, proposée par Fickett et Tung (1992) et appliquée à la prédiction de gènes procaryotes par Tiwari *et al.* (1997). Un développement de cette approche, basée sur une transformée de Fourier discrète, porte le nom de mesure de rotation spectrale ou *"Spectral Rotation Measure"* (Kotlar et Lavner, 2003) et semble être la mesure du codant la plus efficace parmi celles exploitant la seule information de périodicité des régions codantes. Malheureusement, aucun comparatif avec la méthode utilisant les modèles de chaînes de Markov n'est disponible.

Une autre approche semble fournir des résultats intéressants. Il s'agit de la méthode Zcurve (Zhang et Zhang, 1994). Brièvement, il s'agit d'une mesure dépendant de la composition en nucléotides (ou di- ou tri-nucléotides, ou de diverses combinaisons) de chaque position des codons potentiels mesurée sur une partie d'une séquence d'ADN. Cette mesure se traduit par des coordonnées de points dans un espace à trois dimensions. Un entraînement peut se faire sur un jeu de données d'apprentissage comportant des séquences codantes et non-codantes grâce à un algorithme de discrimination (de type équation linéaire de Fischer) qui a pour but de distinguer au mieux les deux catégories. Ainsi, cette méthode permet ensuite d'attribuer une valeur du caractère codant à une partie de séquence donnée, ou au nucléotide placé au centre d'une fenêtre glissante.

Dans une récente publication, Gao et Zhang (2004) réalisent une étude comparative entre les performances de la méthode Zcurve et celles des modèles de Markov (entre autres) sur des régions codantes de courtes tailles. En effet, les petits exons restent actuellement difficiles à localiser par les modèles markoviens du fait de la faible quantité d'information statistique qu'ils contiennent, et les méthodes basées sur la périodicité pourraient apporter de l'information dans ce cas de figure (Kotlar et Lavner, 2003), ce que les résultats de l'étude semblent confirmer (Gao et Zhang, 2004). Malheureusement, on peut se demander si cette analyse exploite les modèles de Markov au maximum de leur potentiel, et des doutes subsistent quant au bénéfice

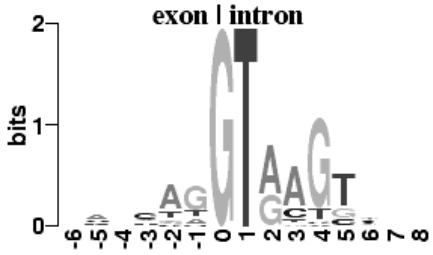

FIG. 2.2 – Représentation graphique d'un alignement de séquences nucléiques, ici pour des sites donneurs d'épissage chez *H. sapiens*. La position 0 en abscisse correspond au début de l'intron, la taille des lettres reflète leur prédominance à la position correspondante de l'alignement (information capturée par les PWM, voir le texte). Image tirée de http ://weblogo.berkeley.edu/examples.html.

réel apporté par cette approche.

2/ Les informations de type *signal*

Des signaux biologiques séparent les différents éléments des gènes (voir Fig. 1.3 page 5). En général, les signaux d'un même type peuvent être caractérisés par une séquence nucléotidique plus ou moins conservée qu'ils ont en commun. En effet, d'un point de vue biologique, leur fonction lors de l'expression génique est de recruter les complexes moléculaires adéquats, par exemple le *"spliceosome"* pour l'épissage, ou le ribosome pour la traduction. Ces complexes interagissent de façon spécifique et précise (au nucléotide près) avec ces signaux grâce à l'information contenue dans la séquence des bases nucléiques. Pour un même type d'interaction moléculaire et donc de signal, il est naturel d'espérer des séquences nucléiques similaires, même si ce n'est pas toujours le cas. À partir d'un alignement de ces séquences, il est donc possible de caractériser des profils statistiques pour ces signaux et de les modéliser par diverses méthodes mathématiques. Ces modèles sont ensuite utilisés sur des séquences à annoter pour aider à la localisation des gènes. Cette partie présente les principales méthodes de modélisation et de détection de ces signaux.

À noter que les techniques de gestion de l'insuffisance de données décrites à partir de la page 22 (pseudo-comptes, VLMM...) sont applicables aux modèles markoviens utilisés dans les méthodes ci-dessous.

a. Les PWM

Considérons un alignement de N séquences de longueur L, constituant un jeu de données d'apprentissage d'un type de signal que l'on souhaite caractériser (codon start, site donneur, site accepteur,...). Ces séquences sont caractérisées par la présence d'un mot particulier à une position de l'alignement (`ATG` pour le start ou `GT` pour le site donneur d'épissage, par exemple) flanqué par un certain contexte de quelques bases que l'on suppose informatives (Fig. 2.2). Un modèle appelé *"Positional Weight Matrices"* (PWM) ou *"Weight Matrix Model"* (WMM) (Staden, 1984a) peut être construit à partir de cet alignement en attribuant pour chaque base possible x et pour chaque position i ($1 \leq i \leq L$) une probabilité d'apparition $P_i(x) = \frac{\#(x_i)}{N}$, où $\#(x_i)$ est le nombre d'occurrences de la base x à la position i dans les séquences

composant l'alignement. On peut considérer un PWM comme un modèle de Markov non homogène d'ordre 0, c'est-à-dire comme une série de modèles markoviens d'ordre 0, un par position. Une fois les paramètres du modèle estimés, il est possible comme vu précédemment (formule (2.1) page 18) de calculer la vraisemblance d'un signal potentiel donné S (dont la première position par rapport aux séquences de l'alignement est i_0) selon le modèle M_1 par la probabilité

$$P_{M1}(S) = \prod_{i=i_0}^{i_0+L-1} P_i(x)$$

De la même façon, on peut construire un modèle M_0 de sites faux positifs, par exemple en extrayant les mots caractéristiques du signal considéré (`ATG`, `GT`...) et leur contexte à des positions qui ne correspondent pas à des signaux fonctionnels. Ainsi, il est possible d'attribuer un score à tout signal potentiel dans une séquence génomique en calculant comme avec la formule (2.2) le logarithme d'un rapport de vraisemblance par

$$Q(S) = \log \frac{P_{M1}(S)}{P_{M0}(S)}$$

Remarque 5 *L'extraction de sites faux positifs se fait à partir de jeux de données de séquences dont on n'est jamais certain de disposer de toute l'information les concernant. Par conséquent, le risque est toujours présent d'inclure dans le modèle des sites fonctionnels non identifiés (d'une façon similaire à ce que nous avons évoqué pour les séquences intergéniques, page 20). Cette éventualité, bien qu'elle limite la perfection du modèle théorique, est simplement ignorée en pratique car on considère sa fréquence négligeable.*

Un PWM demande un nombre relativement restreint de paramètres (seulement $4L$), mais ne permet pas de prendre en compte d'éventuelles dépendances entre nucléotides, et fait l'hypothèse d'une longueur fixe pour le signal. Par exemple, GeneID (Parra *et al.*, 2000) utilise des PWM pour les sites d'épissage notamment.

b. Les WAM

Un *"Weight Array Model"* (WAM) (Zhang et Marr, 1993) est une extension d'un PWM, dans le sens où il équivaut à un modèle de Markov non homogène d'un ordre $k > 0$. Pour le construire, il suffit d'estimer pour chaque position de l'alignement comme ci-dessus une probabilité pour chaque base en fonction du contexte en amont, en appliquant la formule (2.4). La vraisemblance d'un motif donné selon le modèle se calcule comme pour les PWM, en remplaçant simplement les $P_i(x)$ par $P_i(x|x_{i-1}\cdots x_{i-k})$. Un WAM exige l'estimation de $4^{k+1}L$ paramètres et impose aussi un signal d'une longueur constante, mais permet de prendre en compte certaines dépendances entre positions adjacentes. L'estimation complémentaire d'un modèle de faux positifs construit à partir de contextes extraits de signaux non-fonctionels permet de calculer pour un signal donné un logarithme de rapport de vraisemblance

(formule (2.2)). Cette approche est fréquente dans le domaine (Salzberg, 1997; Burge et Karlin, 1997; Korf, 2004; Stanke et Waack, 2003).

c. WWAM, MDD et autres

Le *"Windowed Weighted Array Model"* de Burge et Karlin (1997) ou WWAM fut introduit dans GENSCAN pour disposer d'un nombre suffisant de données pour estimer un WAM d'ordre 2. La différence par rapport à ci-dessus est que pour calculer chaque probabilité conditionnelle correspondant à une position donnée de l'alignement, les fréquences utilisées sont les fréquences moyennes observées sur la région comprenant la position d'intérêt plus les positions voisines. Ainsi, le modèle autorise un peu plus de "souplesse" au niveau de la décomposition du signal en positions spécifiques.

Le *"Maximal Dependence Decomposition"* ou MDD fut également développé par Burge et Karlin (1997) dans GENSCAN. Le méthode a pour but de permettre la prise en compte de dépendances entre nucléotides trop distants pour une approche par WAM. Brièvement, elle revient à construire plusieurs PWM, chacun à partir d'un sous-ensemble des données d'apprentissage, à l'image des classes de $GC\%$ pour les modèles markoviens évoqués plus haut. Ici, la séparation s'effectue en fonction des bases portées par les positions du signal qui présentent le plus de dépendance envers les autres. L'importance de ces corrélations est mesurée par des statistiques de type χ^2. Dans GENSCAN, les MDD servent à modéliser les sites donneurs d'épissage.

Le *"Comparative Weight Array Model"* fut introduit par Zhang *et al.* (2003), pour modéliser les signaux dans le cadre d'une approche comparative du génome de l'homme et de la souris. L'objectif est de capter l'information de conservation entre les signaux présents dans les deux génomes. Il faut pour cela constituer un jeu d'apprentissage constitué d'alignements de paires de signaux (des deux espèces). Un recodage est réalisé à chaque position du signal pour passer de l'alphabet nucléique classique à un alphabet de booléens 0 ou 1, 1 signifiant que le nucléotide de la position considérée du signal est le même dans les deux séquences de l'alignement (0 pour une différence). Ensuite, un modèle WAM d'ordre 1 est estimé sur l'ensemble des séries de 0 et de 1. Le calcul de la vraisemblance s'effectue donc pour tout signal potentiel pour lequel on dispose d'un alignement avec son homologue de l'autre espèce, en fonction des conservations nucléotidiques entre les séquences (Zhang *et al.*, 2003).

D'autres approches encore sont possibles, intégrées ou non dans des prédicteurs de gènes, et dont la place manque ici pour les détailler toutes : réseau de neurones (Reese *et al.*, 1997; Cai et Bork, 1998), réseau bayésien (Chen *et al.*, 2004; Castelo et Guigó, 2004), classification par des fonctions discriminantes linéaires ou non (Salamov et Solovyev, 2000), SVM *"Support Vector Machine"* (Degroeve *et al.*, 2002; Saeys *et al.*, 2004) ou Séparateur à Vaste Marge (proche de l'analyse discriminante linéaire), modèle de régression (Hooper *et al.*, 2000), maximum d'entropie (Yeo et Burge, 2004), ... ou bien sûr toute combinaison de plusieurs méthodes.

3/ Les informations de type *similarité*

Parallèlement aux propriétés intrinsèques de la séquence à annoter, comme nous venons de voir avec son contenu statistique et ses signaux, il existe une toute autre catégorie d'informations d'intérêt majeur, à savoir des informations de type *similarité* ou extrinsèque. Plus précisément, ce sont des similarités détectées par comparaison de la séquence avec d'autres qui peuvent refléter la présence de gènes et donner des indications sur leur structure. Ces similarités concernent plusieurs familles de séquences, celles qui proviennent de données d'expression génique (ARNm et protéine) et celles qui sont issues d'ADN génomique comme la séquence à annoter (pour un rappel sur l'expression génique, voir la Fig. 1.3 page 5).

a. Similarités avec des séquences exprimées

i) Principe

Les séquences exprimées[5] caractérisent des molécules de nature ribonucléique (ARN) ou protéique. L'intérêt qu'elles représentent pour détecter les gènes est évident, car elles en sont directement issues et leur séquence dépend de celle du gène source. En général, lorsqu'un gène est identifié, sa séquence et celle de ses produits ARNm et protéine sont déposées dans les bases de données publiques, et ces informations peuvent alors être exploitées.

ii) Les transcrits

En premier lieu, voyons comment se présente ce type d'information. L'ensemble des transcrits (dont les molécules d'ARNm) qui sont présents dans une cellule à un moment donné définit le transcriptome, représentant les gènes en cours d'expression[6]. Il se trouve qu'à partir d'un ensemble de cellules, il est possible expérimentalement de procéder à une extraction des ARNm contenus et à la synthèse de deux types de molécules importantes pour l'étude du transcriptome, qui peuvent être séquencées et déposées dans les bases de données : les ADNc et les EST (Fig. 2.3 page suivante).

Les ADNc (pour "ADN complémentaire") sont des molécules d'ADN simple brin synthétisées à partir d'ARN. Il est possible de les produire *in vitro* à partir d'extraits cellulaires d'ARNm. Ils portent ce nom car leur séquence est identique à celle de l'ARNm correspondant (aux substitutions U→T près). Puisqu'un ADNc provient d'une molécule transcrite et épissée, sa séquence fournit une information précieuse pour aider à la localisation des exons. Le but de la production d'ADNc est d'obtenir la séquence complète de l'ARNm correspondant. Cependant, les techniques expérimentales permettent rarement de garantir l'intégra-

[5] il arrive fréquemment que l'on utilise ainsi le terme de séquence pour mentionner directement la molécule caractérisée par la séquence proprement dit (séquence exprimée, séquence transcrite).

[6] par extension, le transcriptome d'une espèce désigne l'ensemble des transcrits potentiellement générés à partir du génome considéré ; de la même façon, on appelle protéome l'ensemble des protéines.

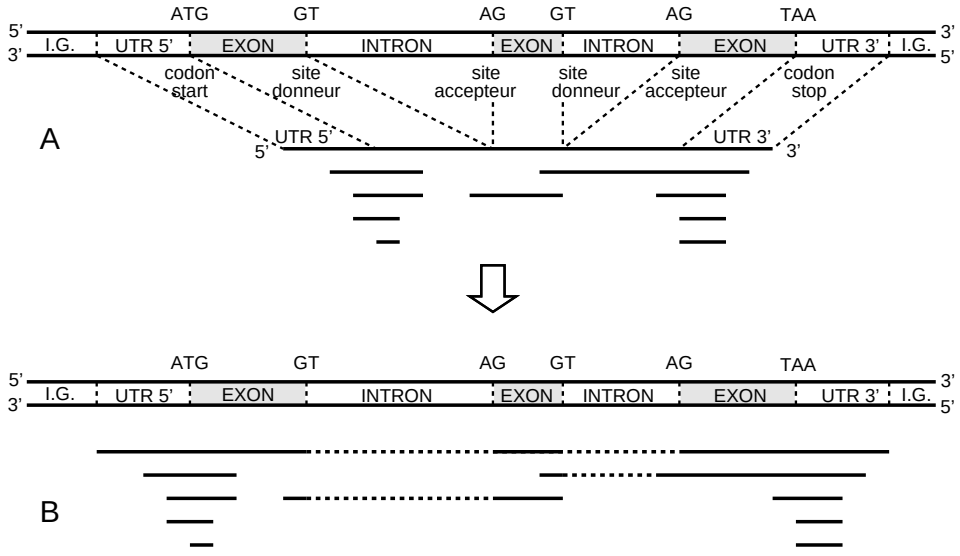

FIG. 2.3 – Représentation de données de séquences transcrites. Le gène est schématisé comme en Fig. 1.3, les traits horizontaux représentent les séquences des transcrits. A : le trait plein horizontal le plus long représente l'ARNm du gène, dont la séquence peut figurer dans les bases de données sous la forme d'ADNc. Les autres montrent des EST, qui sont des fragments de cette séquence. B : alignements des séquences transcrites du dessus sur la séquence génomique. Les traits pleins, régions d'appariements, correspondent aux parties transcrites du gène ; les introns ne faisant pas partie des ARNm, les alignements présentent des régions de brèches.

lité de la copie, et même lorsque c'est le cas (on parle alors d'ADNc de type "pleine longueur", pour *"full-length cDNA"*, ou d'ADNc complet), cette garantie n'est pas totale. Nous reviendrons plus longuement sur les ADNc complets dans le chapitre 6.

Les EST (pour *"Expressed Sequence Tag"*, ou "étiquette de séquence exprimée") correspondent à des fragments d'ADNc, dont ils sont obtenus par séquençage partiel. Il faut savoir à leur sujet qu'ils correspondent en général aux parties terminales des ADNc (plus souvent aux extrémités 3' pour des raisons expérimentales), sont relativement courts (de quelques dizaines à quelques centaines de bases). Par conséquent, il sont aussi moins coûteux en terme de séquençage que des ADNc et donc beaucoup plus abondants dans les bases de données, mais souffrent d'une fréquence élevée d'erreurs dans leur séquence.

Les SAGE et les ORESTES Citons également deux autres classes intéressantes de données de transcrits que les prédicteurs actuels n'ont pas encore intégrées spécifiquement à notre connaissance : les SAGE (pour *"Serial Analysis of Gene Expression"*) qui sont de courts fragments (une douzaine de pb) de parties terminales de transcrits (Velculescu *et al.*, 1995; Boheler et Stern, 2003; Tuteja et Tuteja, 2004), et les ORESTES (pour *"Open reading frame expressed sequence tags"*, construits de façon à ce que leur séquence correspondent préférentiellement à des régions centrales de transcrits, dans la CDS (Dias Neto *et al.*, 2000; Camargo *et al.*, 2001).

Pourquoi exploiter les informations de type transcrit ? Les bases de données publiques contiennent un grand nombre de séquences issues d'ARNm, de type EST et ADNc. Or, la séquence d'un transcrit contient l'assemblage des parties de la séquence génomique qui caractérisent les exons du gène. En d'autres termes, la séquence de chaque exon est parfaitement identique à une section de la séquence du transcrit[7], ce qui peut bien sûr faciliter grandement leur localisation. Par conséquent, si dans la séquence à annoter figure un gène dont l'ADNc complet a déjà été séquencé, il serait particulièrement fâcheux de le rater.

Plus généralement, l'intérêt de l'information d'une séquence transcrite dépasse le cadre de la seule localisation du gène source. En effet, puisque la fonction biochimique d'une protéine dépend de sa structure tridimensionnelle qui elle-même dépend de sa séquence, il est fréquent que des protéines partageant des fonctions communes partagent également certaines parties de leur séquence. Étant donné que la structure du code génétique impose que les codons synonymes ont souvent deux nucléotides sur trois en commun, les ARNm de deux protéines similaires sont en général similaires, ce qui se répercute sur les parties génomiques codantes des deux gènes. En clair, si deux gènes ont des fonctions biochimiques semblables, l'un étant connu et l'autre présent dans une séquence à annoter, la séquence de l'ARNm du premier

[7] dans la pratique, les séquences transcrites sont directement déposées dans les bases de données sous forme d'ADNc, c'est-à-dire avec un alphabet {A,C,G,T}.

peut présenter des similarités avec les parties codantes du second et donc aider à la localisation de ses exons.

Cette stratégie implique des besoins algorithmiques en termes de recherche de similarités et alignements entre séquences, thèmes abordés dans la partie III page 38. De nombreux logiciels exploitent ce type d'information, entre autres ou exclusivement. Citons par exemple GRAIL (Xu et Uberbacher, 1997), HMMGENE (Krogh, 2000), EUGÈNE (Schiex *et al.*, 2001), GENIE (Reese *et al.*, 2000a), GENESEQER (Brendel *et al.*, 2004), PASA (Haas *et al.*, 2003) et CLUSTERMERGE (Eyras *et al.*, 2004).

iii) Les protéines

Le principe est sensiblement le même concernant l'utilisation de séquences protéiques, avec quelques différences par rapport aux transcrits.

Tout d'abord, le séquençage expérimental des protéines est beaucoup plus difficile que celui des acides nucléiques. La plupart des séquences protéiques des bases de données sont obtenues *in silico* par traduction de séquences transcrites connues. On peut alors se demander pourquoi s'intéresser aux protéines, puisque l'information est déjà présente dans les transcrits. D'autant plus que la dégénérescence du code génétique rend ambiguë la détermination de la séquence génomique à partir de la séquence d'acides aminés et donc ne rend pas *a priori* la localisation des régions codantes plus facile qu'en utilisant un ADNc, de séquence identique aux exons.

C'est justement cette dégénérescence qui apporte l'intérêt spécifique des séquences protéiques. En effet, deux gènes ayant en commun certaines fonctions biochimiques et présentant des similarités entre séquences protéiques peuvent diverger suffisamment au niveau ARNm pour que ces similarités ne soit pas détectables en comparant les séquences des transcrits. Le niveau protéique, en concentrant directement l'intérêt sur l'aspect fonctionnel, permet de compenser cette "perte de signal" des intermédiaires et augmente le potentiel des algorithmes de recherche de similarités et d'alignement.

L'utilisation des similarités au niveau protéique demande également des algorithmes de recherche d'alignements. Bien sûr, il faut disposer de séquences d'acides aminés, ce qui est possible même à partir d'une séquence génomique à annoter en la traduisant au préalable dans les six phases possibles[8]. De nombreux logiciels exploitent cette source d'information, comme GENEPARSER (Snyder et Stormo, 1993), le premier à intégrer des informations du type *similarité*, PROCRUSTES (Gelfand *et al.*, 1996), DYNAMITE (Birney et Durbin, 1997), GENIE (Kulp *et al.*, 1997), FGENESH+ (Salamov et Solovyev, 2000), EUGÈNE (Schiex *et al.*, 2001), HMMGENE (Krogh, 2000), GENOMESCAN (Yeh *et al.*, 2001), ou GENEWISE (Birney *et al.*, 2004).

b. Similarités avec des séquences génomiques

L'autre catégorie d'informations de similarité concerne le niveau se situant en amont des séquences exprimées, soit le niveau de l'ADN géno-

[8] 3 phases par brin (voir page 8).

mique. Bien plus récente, l'exploitation de ce type de similarités se base sur une comparaison (par alignement) de séquences génomiques. Avec l'analyse de génomes d'espèces différentes, l'idée sous-jacente est que les séquences fonctionnelles et par conséquent les séquences codantes ont tendance à être davantage conservées que les non-codantes à travers le processus d'évolution des espèces. Ces conservations entre génomes sont appelées des homologies, et peuvent se détecter par des recherches d'alignements entre les séquences d'ADN.

Ce type d'information, son intérêt et son intégration dans le cadre de cette thèse font l'objet du chapitre 5, et sera donc repris plus en détails.

4/ Autres types d'information

Pour terminer ce vaste tour d'horizon des informations exploitées par les logiciels de détection de gènes, nous ne manquerons pas d'évoquer un type d'information d'intérêt particulier : les prédictions produites par les logiciels eux-mêmes. En effet, il s'avère que les prédicteurs de gènes, en raison notamment de la diversité des informations et des méthodes utilisées, produisent des résultats variables (Burset et Guigó, 1996; Rogic *et al.*, 2002; Allen *et al.*, 2004), ce qui se répercute dans les grands projets d'annotation (Hogenesch *et al.*, 2001)[9]. Il est donc naturel de penser que combiner les prédictions de différents logiciels est un moyen d'exploiter les forces et de pallier les défauts de chacun. Cette approche fut évoquée lors de la première évaluation à grande échelle de prédicteurs de gènes (Burset et Guigó, 1996), et développée par Murakami et Takagi (1998); Rogic *et al.* (2002); Pavlovic *et al.* (2002); Tech et Merkl (2003); Yada *et al.* (2003); Zhang *et al.* (2003); Allen *et al.* (2004). Il faut cependant garder à l'esprit qu'une prédiction de gènes, qui résulte d'une décision prise après une intégration d'informations, ne peut contenir toute ces informations d'origine et n'est par conséquent pas aussi riche.

D'autres informations encore peuvent être utilisées, comme des logiciels de prédiction de signaux biologiques (sites d'épissage, codon start...), ou plus largement tout type d'information apportée par l'utilisateur. Enfin, certains systèmes sont conçus pour permettre une incorporation souple d'informations de différents types, comme EuGène (Schiex *et al.*, 2001), Gaze (Howe *et al.*, 2002) ou Combiner (Allen *et al.*, 2004).

5/ Conclusion : l'évolution des sources d'information

i) Que retenir de tout ceci ?

Si l'on prend un peu de recul sur cette partie plutôt technique consacrée aux sources d'informations utilisées, il se dégage d'un point de vue historique un constat fondamental : du fait de la production massive de données

[9] cette étude comparative entre les annotations produites par Celera et Ensembl révélait que 80% des nouveaux transcrits n'étaient prédits que par un seul des deux groupes.

issues du domaine de la génomique, de nouvelles sources d'informations n'ont cessé d'apparaître dans les logiciels d'annotation au cours des 20 dernières années. En outre, il est notable que l'évolution générale des prédicteurs de gènes tend à toujours se diriger vers l'intégration de ces différentes sources d'informations. La figure 2.4-2.5 des pages 36-37 illustre cette dynamique, exposée ci-dessous.

ii) Dynamique des intégrations

En effet, au début de la problématique de détection des régions codantes, deux approches s'opposaient, issues chacune de l'étude des séquences de gènes connus : l'approche *par contenu* ou globale et l'approche *par signal* ou locale (Staden, 1984b; Gelfand, 1990), distinguant les outils qui utilisaient respectivement les informations de type *contenu* et de type *signal* (Fig. 2.4 A). Puis, ces deux sources d'information furent intégrées (Fig. 2.4 B) dans les logiciels développés par la suite (Fields et Soderlund, 1990; Gelfand, 1990; Guigó *et al.*, 1992, ...). Le logiciel GenScan (Burge et Karlin, 1997), par exemple, appartient à cette catégorie.

Ensuite, l'augmentation des données d'expression disponibles (transcrits, protéines) a rendu possible l'émergence d'une nouvelle approche : l'approche extrinsèque qui utilise les similarités avec des séquences exprimées, qui s'oppose avec la précédente, appelée intrinsèque ou *ab initio* (Borodovsky *et al.*, 1994; Fickett, 1995). Ces deux classes distinctes d'informations (Fig. 2.4 C) furent alors intégrées (Fig. 2.4 D) dans de nombreux prédicteurs (Snyder et Stormo, 1995; Kulp *et al.*, 1997, ...). Le logiciel GenomeScan (Yeh *et al.*, 2001), par exemple, résulte de l'intégration de données protéiques dans GenScan.

Puis, la génomique poursuivant son évolution, de nombreux génomes commencèrent à être séquencés, permettant ainsi le recours à un nouveau type d'information. La notion de conservation entre séquences génomiques fut exploitée, en combinaison seulement avec des informations de type *signal* plutôt rudimentaires (Bafna et Huson, 2000; Novichkov *et al.*, 2001; Wiehe *et al.*, 2001)[10]. Cette approche se base sur l'étude génomique comparative (Fig. 2.4 E). À nouveau (Korf *et al.*, 2001; Parra *et al.*, 2003), cette information fut intégrée dans des logiciels de type *ab initio* (Fig. 2.5 F). Par exemple, Twinscan (Korf *et al.*, 2001) résulte de l'intégration de la génomique comparative dans GenScan.

Or, à partir d'ici, une certaine limite semble être atteinte dans la capacité d'intégration des logiciels. En effet, concernant cette information de similarités entre génomes, aucun logiciel à notre connaissance (à part EuGène, dont nous reparlerons en détails) ne l'intègre avec toutes les autres (*ab initio* + similarités avec des données d'expression transcrits/protéines). Par exemple, alors que les versions "intrinsèques + extrinsèques" GenomeScan et GeneID+ existent, les similarités inter-génomiques ne furent intégrées

[10] en exceptant toutefois le programme Rosetta (Batzoglou *et al.*, 2000) qui utilise un WAM pour les sites d'épissage et adopte en plus de la conservation l'usage du code pour le caractère codant.

que dans les versions intrinsèques correspondantes (respectivement GENSCAN et GENEID, pour créer TWINSCAN et SGP2).

Enfin, concernant l'approche qui considère des prédictions de logiciels comme source d'information, les méthodes d'intégration semblent là aussi relativement limitées. En effet, les logiciels développés sur cette base ne peuvent prendre en compte que cette information uniquement (Fig. 2.5 F), en faisant même parfois abstraction totale de la séquence génomique[11].

iii) L'amorce de la problématique

Ces intégrations successives de nouvelles sources d'information ne sont pas le fruit du hasard : il apparaît que chaque nouvelle combinaison décrite permet une augmentation des performances des logiciels par rapport à chacune des sources distinctes prise individuellement. à l'issue de cette étude, on s'attend donc naturellement à ce que les récents logiciels proposent une intégration globale de l'ensemble des informations aujourd'hui disponibles (Fig. 2.5 G). De plus, nous sommes convaincus que l'efficacité d'intégration des informations disponibles est un élément clef de la prédiction de gènes. Or, seuls trois outils semblent permettre une intégration de tous les types d'information : GAZE (Howe *et al.*, 2002) (bien qu'aucun test ne soit publié pour les informations de type "prédiction"), COMBINER (Allen *et al.*, 2004) (bien qu'aucun test ne soit publié pour les informations de génomique comparative), et EUGÈNE (Schiex *et al.*, 2001), dont nous reparlerons. Pourquoi les outils n'évoluent-ils pas au même rythme que les informations disponibles ? Pourquoi les logiciels existants sont-ils en général limités à un ensemble fixe de sources d'information ? C'est au travers de l'analyse des modèles, des algorithmes et des méthodes d'intégration que nous allons apporter des éléments de réponse à ces interrogations.

[11] on peut noter comme exception le logiciel COMBINER (Allen *et al.*, 2004), que l'on peut considérer comme un outil d'intégration générique au même titre que EUGÈNE (Schiex *et al.*, 2001) ou GAZE (Howe *et al.*, 2002).

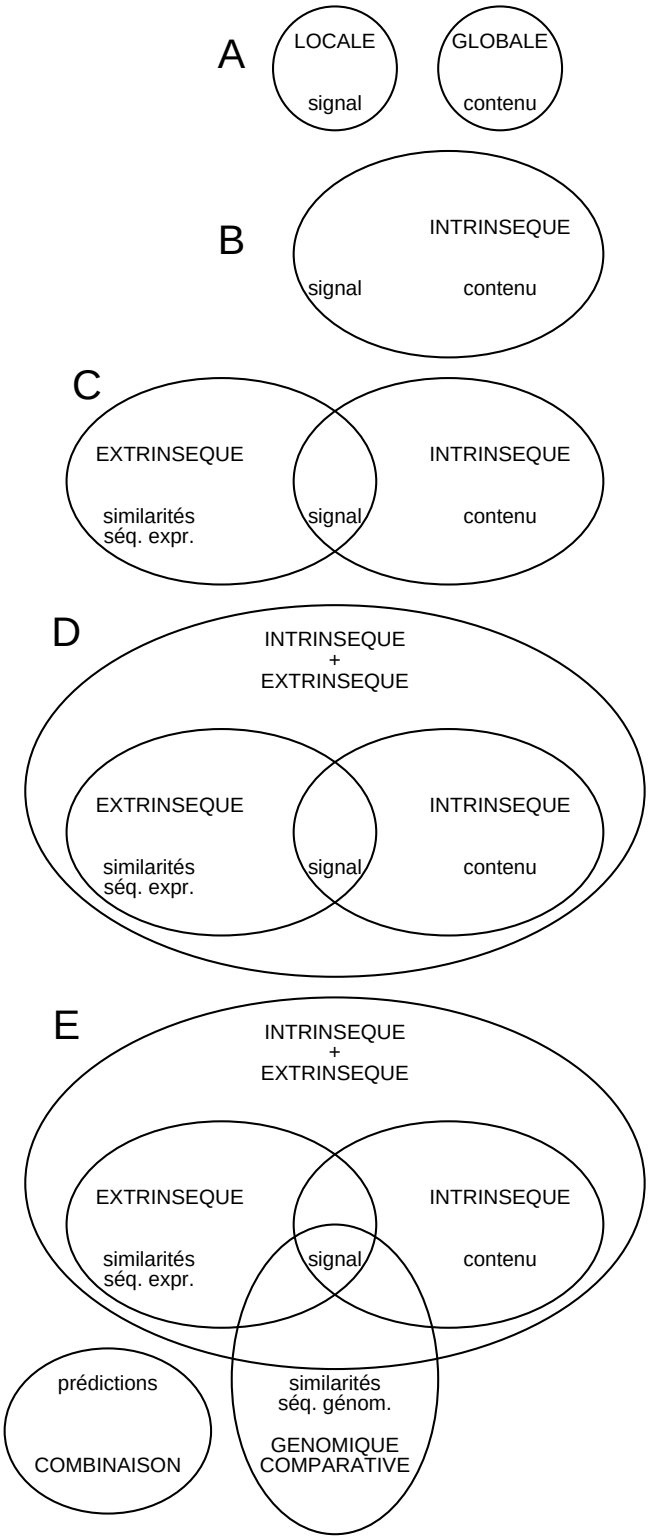

FIG. 2.4 – Dynamique de l'intégration des types d'informations au cours de l'évolution du domaine de la prédiction de gènes (1re partie). Légende ci-contre.

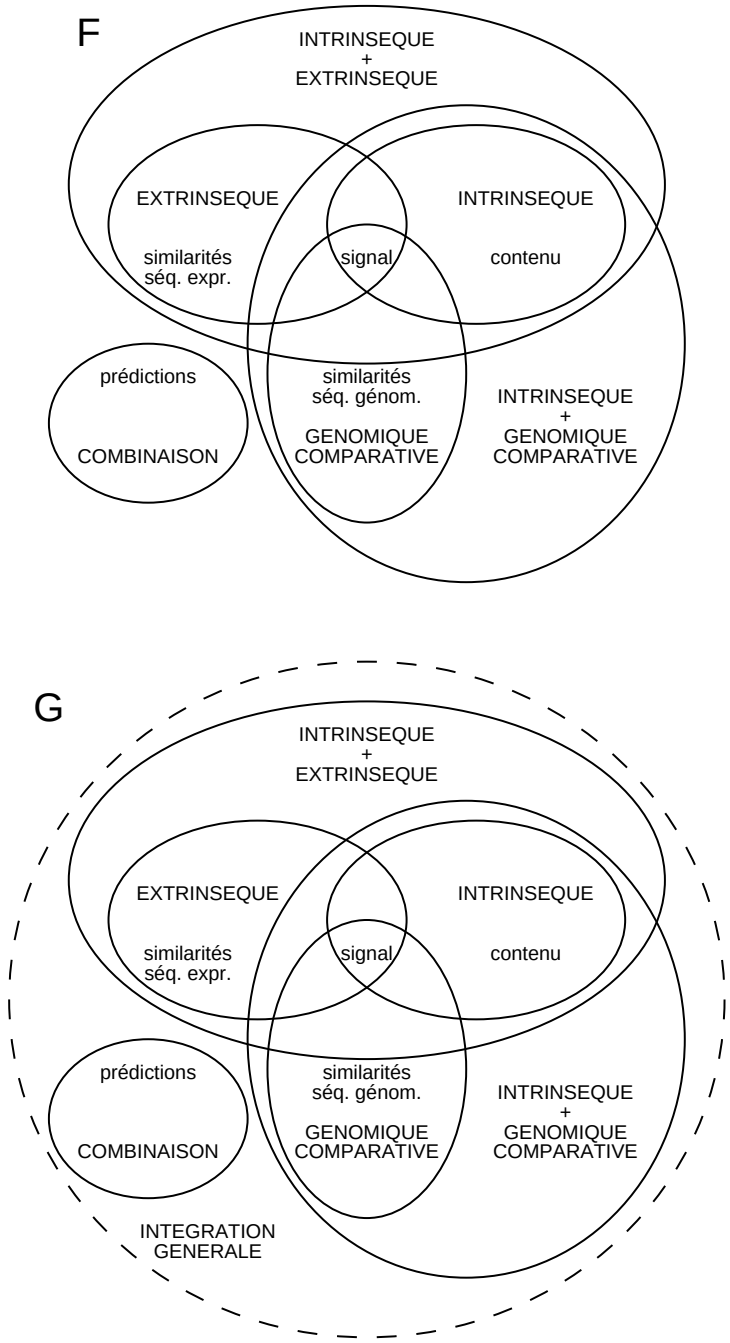

FIG. 2.5 – Dynamique de l'intégration des types d'informations au cours de l'évolution du domaine de la prédiction de gènes (suite). En lettres minuscules, les sources d'informations de type *signal*, *contenu*, *similarité* avec des séquences exprimées, *similarité* avec des séquences génomiques et prédictions d'autres logiciels. En lettres majuscules, le nom du type d'approche associé aux logiciels qui intègrent l'ensemble d'information entouré. Détails et exemples dans le texte.

III Les modèles et les algorithmes associés

Après ce large inventaire des informations disponibles pour aider la localisation des régions codantes dans les génomes, nous allons nous intéresser d'une part aux modèles qui sont destinés à les incorporer et qui servent à représenter l'espace de recherche, et d'autre part aux algorithmes qui permettent d'identifier dans cet espace une prédiction de gènes tenant compte des informations.

Parmi les approches développées se dégagent deux familles principales de méthodes (Guigó, 1998) qui dépendent du niveau de décomposition du problème : l'approche basée sur les segments (niveau exons) et l'approche basée sur les positions (niveau nucléotides).

1/ Niveau exon : assemblage de segments

a. Présentation générale

L'idée de cette approche est la suivante : la structure de gènes d'une séquence génomique étant définie par un ensemble d'exons, chacun localisé par une position de début et de fin, identifier correctement les gènes de la séquence revient donc à trouver l'ensemble des positions qui caractérisent les exons composant ces gènes.

Formellement, si l'on caractérise un exon $e = \{d, f\}$ par un couple de coordonnées indiquant sa position d de début et f de fin dans la séquence (avec $d < f$), une structure de gènes $g = \{e_1, e_2, \cdots, e_n\}$ est définie par un ensemble ordonné ou assemblage de n exons $e_1 = \{d_1, f_1\}, \cdots, e_n = \{d_n, f_n\}$ tels que $\forall i_{(1 \leq i \leq n-1)}$ on a $f_i < d_{i+1}$. De plus, si l'on attribue un score $S(e)$ à tout exon e, on peut également attribuer un score $S(g)$ à tout assemblage $g = \{e_1, \cdots, e_n\}$ tel que $S(g) = f(S(e_1), \cdots, S(e_n))$, où f est une fonction des scores (généralement la somme).

Pour trouver les gènes contenus dans une séquence génomique donnée, la méthode basée sur les exons procède en plusieurs étapes distinctes :

- localisation et attribution d'un score aux signaux dans la séquence qui représentent des frontières d et f possibles d'exons ;
- construction et attribution d'un score aux exons potentiels e de la séquence, chacun étant défini par un couple de signaux d et f ;
- identification d'un assemblage optimal g^*, c'est-à-dire d'un ensemble d'exons potentiels non chevauchants dont le score $S(g^*)$ est maximal, ce score étant défini par une fonction des scores des exons assemblés.

L'espace de recherche représenté par le modèle est donc l'ensemble des assemblages possibles. Cette approche est utilisée par des logiciels comme GeneID (Guigó *et al.*, 1992; Parra *et al.*, 2000), GapIII (Xu *et al.*, 1994), Fgeneh (Solovyev *et al.*, 1994; Solovyev et Salamov, 1997), Grail (Xu et Uberbacher, 1997), GeneGenerator (Kleffe *et al.*, 1998), GlimmerM (Salzberg *et al.*, 1999; Majoros *et al.*, 2003), Dagger (Chuang et Roth, 2001). Nous présentons ici les grandes lignes de la méthode, dont on peut trouver des

variantes.

b. Caractérisation des exons potentiels

En premier lieu, la séquence est parcourue pour détecter et attribuer un score aux signaux déterminant des frontières possibles d'exons (codons start et stop, sites d'épissage). Puis, les exons potentiels de la séquence (définis par des intervalles entre ces signaux) sont construits. Chacun de ces exons reçoit un score qui est fonction de celui de ses signaux et de diverses statistiques reflétant son caractère codant. C'est à ce niveau que s'effectue l'intégration des informations.

i) Les signaux

La première étape est la lecture de la séquence pour caractériser les signaux potentiels, ce qui s'effectue en cherchant les mots `ATG, GT, AG, TAA, TAG, TGA,` qui correspondent respectivement au codon start, au site donneur d'épissage, au site accepteur d'épissage, et aux trois codons stop. Chaque signal potentiel détecté reçoit un score déterminé par une méthode de type *signal*, comme nous avons vu précédemment (par exemple un PWM ou un WAM). À noter qu'un premier filtrage peut être réalisé à ce niveau afin d'alléger la suite de la procédure. Pour cela, on peut choisir d'écarter du processus les signaux dont le score est inférieur à une valeur seuil, que l'on considère *a priori* comme éliminatoire (Solovyev et Salamov, 1997; Guigó *et al.*, 1992).

Puisque le nombre maximum de signaux contenus dans une séquence génomique est une fonction linéaire de sa taille, cette étape est d'une complexité linéaire en temps et en espace par rapport à la longueur de la séquence.

ii) Les exons

Dans un second temps, on construit tous les exons possibles à partir de ces signaux, en respectant des contraintes d'origine biologique (pour un rappel sur la structure des gènes, voir la Fig. 1.3 page 5). Par exemple, un exon doit être flanqué par un des couples possibles de signaux (codon start+site donneur, ou site accepteur+site donneur, ou site accepteur+codon stop, ou codon start+codon stop) qui définissent des types d'exon précis (respectivement exon initial, exon interne, exon terminal et exon unique). De plus, on sait que pour faire partie d'un gène, un exon ne peut contenir de codon stop dans la phase de lecture du gène (déterminée par la position du codon start dans la séquence, voir page 8). Comme celle-ci ne peut être connue *a priori* pour un exon pris individuellement (excepté pour un exon initial pour lequel la position du start est connue), on considère toutes les phases pour chaque exon (ce qui revient à construire autant de copies de chaque exon qu'il y a de phases possibles).

Remarque 6 *Le nombre de signaux présents dans la séquence augmentant linéairement en fonction de la taille de celle-ci, on pourrait penser de prime abord que le nombre d'exons possibles, qui dépend du nombre de paires de*

signaux, augmente de façon quadratique. Or, ce n'est pas le cas en pratique. En effet, on considère généralement que les codons stop apparaissent le long de la séquence suivant un processus de Poisson, interrompant ainsi régulièrement tout exon de la phase correspondante. On peut par conséquent négliger la probabilité que la taille d'un exon soit supérieure à un certain seuil, ce qui permet de considérer que la taille des exons potentiels d'une séquence est bornée. C'est pourquoi leur nombre augmente linéairement en pratique avec la taille de la séquence génomique.

Enfin, il faut attribuer un score à chacun de ces exons. Il est en général fonction des scores de ses signaux, calculés précédemment, et d'un score reflétant son caractère codant. Ce dernier repose typiquement sur les mesures statistiques de type *contenu* décrites précédemment. En outre, puisque la longueur de chaque exon potentiel est connue, il est aisé de prendre en compte une distribution de longueur estimée au préalable à partir d'exons déjà identifiés pour modifier son score[12]. On peut également inclure à cette étape un score donné par des informations de type *similarité*, si des régions codantes d'autres gènes présentent des séquences semblables (Parra *et al.*, 2003). La façon dont ces informations sont intégrées sera abordée plus loin.

c. Assemblage optimal

Une fois que les exons potentiels sont caractérisés, l'objectif est de produire un assemblage de score global maximum, pour proposer la meilleure structure de gènes possible. Le score d'un assemblage est généralement défini comme la simple somme des scores des exons le composant. Des contraintes doivent être respectées pour garantir la pertinence biologique de la solution globale (respect de la phase de lecture, non chevauchement des exons, ...).

Le nombre d'assemblages possibles augmente exponentiellement avec le nombre d'exons (de type interne), et donc avec la taille de la séquence. C'est pourquoi les logiciels actuels utilisent la programmation dynamique. Il s'agit d'une méthode de résolution itérative appliquée aux problèmes respectant la propriété de Bellman (Bellman, 1957), c'est-à-dire pouvant être décomposés en sous-problèmes, la solution optimale étant elle-même composée de sous-solutions optimales. Sans entrer dans les détails (nous décrirons d'autres méthodes de programmation dynamique), l'algorithme prend en entrée un ensemble d'exons, et produit en sortie l'assemblage qui offre le cumul de scores le plus important. Cet assemblage est construit de façon itérative par une formule de récurrence, calculant pour chaque exon (pris dans l'ordre croissant des positions de site accepteur) le score du meilleur assemblage se terminant par cet exon. Pour davantage de précisions, prière de se référer au travail de Guigó (1998).

Les premiers algorithmes proposés pour répondre à ce problème font état d'une complexité soit exponentielle (Parra *et al.*, 2000), soit quadratique en fonction de la taille de la séquence génomique (Solovyev *et al.*, 1994; Xu *et al.*,

[12] l'intégration de l'information des longueurs est plus problématique dans le cas des HMM, comme nous allons le voir.

1994). Avec la publication de l'algorithme GenAmic (Guigó, 1998), intégré depuis dans les logiciels GeneID (Parra *et al.*, 2000) et Fgeneh (Solovyev et Salamov, 1997), le procédé acquiert une complexité linéaire en temps et en espace avec la longueur de la séquence génomique.

d. Intégration des informations : par segments

i) Problématique

Nous avons évoqué en conclusion de la section précédente (consacrée aux sources d'informations) la difficulté d'intégrer des informations de natures différentes (Fig. 2.5 page 37). Dans l'approche basée sur les segments, selon les sources d'informations exploitées, chaque exon potentiel peut être caractérisé par plusieurs valeurs numériques, en fonction par exemple de son contenu statistique, de ses frontières, de sa longueur, ou encore de l'existence de séquences exprimées ou génomiques similaires.

La question qui se pose alors est de combiner ces valeurs d'une façon appropriée, afin d'attribuer à l'exon potentiel correspondant un score unique représentant au mieux sa probabilité de faire partie de la structure génique. Or, il ne s'agit pas d'une tâche évidente, et ce pour deux raisons essentielles :

- Les différentes sources d'information ne proposent pas forcément des valeurs sur la même échelle de notation.
- L'importance et la confiance relative que l'on peut accorder à chaque source d'information varient généralement de l'une à l'autre.

Les tentatives de résolution de ce problème décrites dans la littérature fournissent quelques exemples intéressants de stratégies entreprises, que les informations soient de type *contenu*, *signal*, *similarité*, ou *prédiction*.

ii) Méthode empirique

La façon la plus simple et la plus intuitive d'intégrer différentes valeurs numériques en une seule est de les additionner. Pour les mettre à l'échelle et prendre en compte différents niveaux de confiance, il suffit au préalable d'appliquer à chacune une fonction mathématique spécifique, comme une simple multiplication par un paramètre de pondération (ou poids).

Par exemple, à un ensemble d'informations que l'on souhaite mettre à l'échelle $\{I_1, \cdots, I_n\}$ peut être attribué un ensemble de poids $\{p_1, \cdots, p_n\}$ pour donner à chaque exon possible un score $S = \sum_{i=1}^{n} p_i \cdot I_i$. La détermination de ces paramètres représente un point délicat de la méthode. On peut les fixer de manière empirique, ce qui revient à procéder à des tests et à des modifications en fonction des erreurs observées[13]. Cette procédure se retrouve dans le cas de Fgenehb avec des similarités de séquences protéiques (Solovyev et Salamov, 1997) ou Grail avec des similarités de séquences transcrites (Xu et Uberbacher, 1997).

GeneParser Une alternative intéressante fut développée par Snyder et Stormo (1993) pour estimer ces paramètres de pondération dans le logiciel

[13] par tâtonnement pourrait-on dire de façon imagée et légèrement péjorative.

précurseur GeneParser, qui fut le premier à proposer la programmation dynamique dans le domaine (Snyder et Stormo, 1993) et à combiner les approches intrinsèque extrinsèque (Snyder et Stormo, 1995).

Le principe est de réaliser un apprentissage itératif avec un réseau de neurones à partir d'un jeu de données de séquences génomiques pour lesquelles la structure génique est connue. Des valeurs initiales des paramètres sont choisies aléatoirement, et les séquences du jeu d'apprentissage sont présentées au logiciel, qui prédit des structures de gènes (en général incorrectes à la première étape). Puis, les valeurs des poids sont modifiées par le réseau de neurones, l'objectif étant de définir des valeurs qui permettent d'affecter pour une structure correcte donnée un score global supérieur à celui de toutes les structures incorrectes possibles. Comme il n'est pas possible de tester toutes les structures potentielles, ce sont les prédictions incorrectes du logiciel qui sont utilisées pour les calculs. Le logiciel est relancé sur le jeu d'apprentissage avec les paramètres modifiés, et ainsi de suite (Snyder et Stormo, 1993).

La méthode a cela de remarquable que la qualité d'un jeu de paramètres est estimée en fonction des prédictions produites sur des séquence entières, et non localement sur les segments. Cette approche pragmatique permet de combiner efficacement l'approche intrinsèque et l'approche extrinsèque (Snyder et Stormo, 1995). De plus, une procédure automatique est définie, plus rigoureuse et moins exigeante en moyens humain qu'un ajustement manuel, ce qui permet d'envisager de renouveler l'apprentissage si les données disponibles évoluent.

Gaze Pour tester l'intégration de données de séquences transcrites dans le logiciel Gaze, Howe *et al.* (2002) ont également utilisé des paramètres de pondération pour réaliser une mise en échelle des sources d'information. La perspective d'une procédure automatique d'estimation de ces paramètres fut également évoquée, mais la méthode choisie et mise en application à cette fin, basée sur une optimisation par descente de gradient (Stormo et Haussler, 1994) ne semble pas aboutir à des résultats satisfaisants d'après les auteurs (Howe *et al.*, 2002). La piste ne semble malheureusement pas avoir été poursuivie avec succès. Par exemple, Gaze fut très récemment utilisé pour annoter le génome du poisson *Tetraodon nigroviridis* en intégrant des informations de tous les types, dont des prédicteurs de gènes *ab initio* et des similarités avec des séquences transcrites, traduites et génomiques (Jaillon *et al.*, 2004). La procédure de calibrage de ces informations comprend bien une normalisation des scores (sur une échelle de 1 à 100) suivie d'une pondération par des paramètres de confiance (voir les informations supplémentaires relatives à la publication), mais la méthode de détermination du paramètre associé à chaque source d'information n'est pas décrite. Il semble qu'il s'agisse encore d'une recette empirique et qu'une procédure d'estimation rigoureuse n'est pas encore d'actualité. Dans le même ordre, on peut évoquer l'une des premières versions du logiciel Grail (Uberbacher et Mural, 1991), où les valeurs de paramètres de pondération (ajoutés directement aux scores des exons) étaient déterminés de façon empirique. Bien

que les auteurs aient également évoqué l'intérêt de mettre en place un cadre théorique permettant une estimation automatique de ces paramètres, cela ne semble pas avoir donné de résultats positifs, la publication relative à la version suivante ne détaillant pas de processus semblable (Xu et Uberbacher, 1997).

iii) Méthode probabiliste

La mise à l'échelle des différentes informations peut s'effectuer en les incorporant au sein d'un cadre probabiliste. Les informations de type *contenu* et *signal* se prêtent bien à ce traitement, car comme nous l'avons vu elles se quantifient souvent sous la forme de logarithmes de rapport de vraisemblance. Si l'on fait l'hypothèse que ces informations sont indépendantes, il est correct d'additionner simplement ces valeurs afin d'obtenir un score pour chaque exon.

Malheureusement, l'opération ne semble pas si simple, comme on peut le constater à travers les publications relatives au logiciel GeneID (Parra *et al.*, 2000), qui utilise l'approche par assemblage d'exons à partir de mesures de rapport de vraisemblance.

En effet, l'application d'un modèle probabiliste présente un point faible, précisément au niveau de l'hypothèse sous-jacente d'indépendance des données. Même en restant dans un cadre *ab initio*, il est délicat d'affirmer que les informations de type *signal* et *contenu* ne sont pas reliées. De fait, le contenu statistique des exons est mesuré sur la longueur totale du segment, qui comprend obligatoirement le contexte des signaux flanquants. Ce chevauchement limite l'exactitude du modèle. De multiples imperfections supplémentaires difficilement évitables (voire inévitables) s'ajoutent à cela, provenant par exemple des différentes mesures d'information (ne serait-ce que du fait de l'utilisation de jeux de données toujours imparfaits[14]). Par exemple, comme le soulignent justement Parra *et al.* (2000) pour GeneID, une approximation découle du fait que le caractère codant d'un exon est calculé à partir d'un rapport de vraisemblance du modèle codant sur un modèle non-codant imparfait (construit dans ce cas uniquement avec des introns).

Dans la version *ab initio* de GeneID, un paramètre de pondération constant est ajouté au score de chaque exon potentiel afin de corriger les imperfections du modèle. Malgré le cadre probabiliste, l'estimation de ce paramètre est effectuée de façon empirique par un processus d'optimisation simple sur le jeu d'apprentissage (Parra *et al.*, 2000).

Le logiciel Sgp2 (Parra *et al.*, 2003) est en fait une version de GeneID qui intègre l'information de similarité avec des séquences génomiques. Avant l'intégration, chaque exon potentiel reçoit deux scores : le score intrinsèque issu des méthodes de GeneID et le score extrinsèque reflétant le niveau de similarité. Les deux sont des logarithmes de rapport de vraisemblance. Sous l'hypothèse d'indépendance des informations, une simple addition est envisageable. Cependant, comme le soulignent les auteurs (Parra *et al.*, 2003),

[14] voir par exemple la partie consacrée aux modèles intergéniques page 20, ou la remarque 5 page 27 au sujet de modèles de signaux faux positifs.

cette hypothèse n'est pas réaliste, et un paramètre de pondération est associé au score de similarité pour le mettre à l'échelle. La valeur de ce paramètre est également déterminée de façon empirique.

iv) Conclusion

La modélisation des gènes par segments offre un cadre souple d'intégration des données issues de différentes sources, si l'on en juge par la variété des informations incorporées. On trouve en effet des logiciels basés sur l'assemblage d'exons aussi bien pour exploiter les traditionnelles informations intrinsèques que pour des similarités avec des transcrits, des protéines ou des séquences génomiques.

Cependant, malgré les tentatives d'utilisation d'un cadre probabiliste théorique, la fusion d'informations hétérogènes nécessite en général un calibrage délicat à réaliser. Cette analyse des méthodes par segments semble confirmer notre hypothèse d'un point faible dans la capacité d'intégration d'informations des logiciels existants (Fig. 2.5 page 37). L'approche la plus intéressante nous semble être la procédure d'estimation des paramètres de pondération par apprentissage développée dans le logiciel GeneParser (Snyder et Stormo, 1995). Malheureusement, cette méthode d'optimisation des performances semble être dans le domaine la seule de son espèce. De plus, les performances de ce logiciel furent dépassées par celles de programmes plus efficaces, dont certains utilisent un tout autre cadre d'intégration d'informations pour modéliser les structures de gènes possibles d'une séquence génomique : le cadre des HMM.

2/ Par positions : les HMM

a. Philosophie sous-jacente

La deuxième approche d'intégration d'informations pour la prédiction de gènes utilise des modèles de chaînes de Markov à états cachés (HMM pour *"Hidden Markov Models"*, ou modèles de Markov cachés). Le principe de cette approche basée sur les positions nucléotidiques est de considérer que le score d'un segment peut être défini comme la somme des scores locaux des éléments de base le composant (typiquement les nucléotides). Dans sa version la plus simple, l'hypothèse est donc faite que les caractéristiques globales d'un exon n'influencent pas son potentiel codant. La séquence est ainsi décomposée en nucléotides individuels, ce qui est le niveau le plus réduit. Enfin, le tout est défini à l'intérieur d'un cadre entièrement probabiliste.

L'approche par HMM fut initialement proposée dans le domaine de la détection des gènes par Krogh *et al.* (1994) pour l'annotation de la bactérie *Escherichia coli*. Depuis, de nombreux prédicteurs exploitent cette méthode, parmi lesquels figurent Genie (Reese *et al.*, 2000a), HMMgene (Krogh, 2000), Veil (Henderson *et al.*, 1997), GenScan (Burge et Karlin, 1997), Fgenesh (Salamov et Solovyev, 2000), Unveil et Exonomy (Majoros *et al.*, 2003), Augustus (Stanke et Waack, 2003), ou Snap (Korf, 2004). Comme référence sur les HMM, voir par exemple (Rabiner, 1989; Durbin *et al.*, 1999).

b. Modélisation par les HMM

i) Formalisme

Un modèle de chaînes de Markov à états cachés, ou HMM (pour *"Hidden Markov Models"*) est un modèle probabiliste associé à une suite ordonnée de variables aléatoires x_i qui vérifie la propriété de Markov (voir page 16). La différence avec les simples modèles de Markov est qu'à chaque variable x_i est attribué un état j parmi un ensemble J d'états possibles. La succession des états le long de la suite de variables est appelée chemin, que l'on note π, où π_i est l'état associé à la variable x_i.

À chaque état correspond un modèle de Markov définissant les probabilités conditionnelles pour les variables. L'enchaînement des états, le chemin, suit lui-même un processus markovien. Formellement, cela donne

Définition 3 (HMM)
Un HMM est défini par :

- *une variable aléatoire x ordonnée par un indice i et dont l'ensemble des valeurs est $\mathcal{A}$.*
- *un ensemble d'états J associés aux variables, π_i étant l'état de la variable x_i.*
- *des probabilités d'émission associées à chaque état*

$$e_j(b) = P(x_i = b|\pi_i = j) \quad \forall j \in J \text{ et } \forall b \in \mathcal{A}$$

- *des probabilités de transition entre états*

$$t_{j,j'} = P(\pi_i = j'|\pi_{i-1} = j) \quad \forall j, j' \in J$$

Remarque 7 *à cette définition vient s'ajouter une matrice de probabilité supplémentaire, déterminant les probabilités initiales de se trouver dans chacun des états au début d'une chaîne. En effet, les $t_{j,j'}$ définissent les probabilités de trouver chaque état j' après chaque état j, mais ne précisent pas celles qui correspondent à l'absence de contexte, qui sont données par*

$$t_{0,j} = P(\pi_1 = j) \qquad \forall j \in J$$

Remarque 8 *Comme nous avons vu, les probabilités d'émission caractérisant les états peuvent définir des modèles de chaîne de Markov, prenant ainsi en compte les variables précédentes. Dans ce cas, il faut remplacer la définition des probabilités d'émission par*

$$e_j(b|w') = P(x_i = b|\pi_i = j, w_{i-k}^k = w') \quad \forall j \in J \text{ , } \forall b \in \mathcal{A} \text{ , et } \forall w' \in W^k$$

où w_{i-k}^k est le mot de longueur $k > 0$ dont la première lettre est x_{i-k} et W^k l'ensemble des mots possibles de longueur k (tableau 2.1 page 17).

ii) Application aux séquences nucléiques

Appliquer un HMM à une séquence nucléique (avec l'alphabet $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$) revient à considérer que l'on peut y distinguer différentes régions (associées aux états), chacune possédant ses propres caractéristiques statistiques. Par exemple, on peut considérer un cas simpliste avec un HMM comportant deux états C^1 et C^0, le premier étant codant et le second non-codant. Chaque état est alors caractérisé par un modèle de Markov distinct, reflétant les propriétés markoviennes distinctes des régions codantes et des régions non-codantes (probabilités d'émission).

Pour une séquence génomique donnée, les données observables sont les bases successives, soient les valeurs de la variable x_i avec i comme position dans la séquence. Sans information d'annotation pour la séquence, comme c'est le cas dans le cadre de la prédiction de gènes, la répartition des états le long de la séquence est inconnue. C'est pourquoi l'on dit que les états sont cachés. Localiser les régions codantes revient donc à reconstruire les états cachés pour la séquence.

iii) Représentation graphique

Tout HMM peut être représenté graphiquement. Pour rester dans le cadre de la prédiction de gènes, reprenons notre exemple simple de HMM où l'on distingue uniquement les régions codantes du reste de la séquence génomique par les états C^1 et C^0. Un schéma de ce HMM est proposé en figure 2.6.

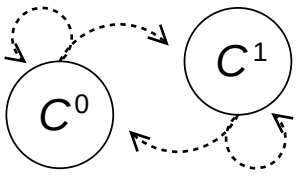

FIG. 2.6 – Représentation d'un HMM basique distinguant un état non-codant C^0 d'un état codant C^1. Les flèches pointillées représentent les transitions possibles entre les états, à chacune est associée une probabilité $t_{C^n,C^{n'}}$, n et n' prenant les valeurs 0 ou 1 selon les états de départ et d'arrivée. Au sein de chaque état sont définies des probabilités d'émission des 4 nucléotides.

Bien entendu, les prédicteurs de gènes actuels utilisent des modèles bien plus complexes, distinguant en général des états intergéniques, introniques, UTR3' et UTR5', plusieurs types d'exons (initial, interne, terminal) dans chaque phase. De plus, afin de permettre la représentation d'un gène quel que soit son orientation (sur le brin sens ou anti-sens, voir les remarques pages 3 et 6), tous les états excepté l'intergénique sont généralement doublés.

c. Applications intéressantes des HMM

i) Vraisemblance

Pour une séquence génomique donnée dont on connaît les positions des exons, il est possible de calculer sa vraisemblance en fonction d'un HMM. La probabilité conjointe d'une séquence observée x de longueur L et d'une annotation π est de

Transition

t_{C^0,C^0}	t_{C^0,C^1}	t_{C^1,C^1}	t_{C^1,C^0}
0.9	0.1	0.7	0.3

Emission

	A	C	G	T
C^1	0.2	0.4	0.3	0.1
C^0	0.35	0.25	0.15	0.25

TAB. 2.2 – Probabilités caractérisant un HMM à deux états C^0 et C^1. Dans cet exemple, l'ordre des modèles markoviens pour les probabilités d'émission est $k = 0$. Probabilités initiales : $t_{0,C^0} = 0.8$, $t_{0,C^1} = 0.2$.

$$P(x,\pi) = t_{0,\pi_1} e_{\pi_1}(x_1) \prod_{i=2}^{L} t_{\pi_{i-1},\pi_i} e_{\pi_i}(x_i)$$

EXEMPLE Prenons notre modèle de la Fig. 2.6 ci-contre, avec les probabilités indiquées dans la table 2.2. La vraisemblance selon ce modèle de la séquence aac`AG` (avec la partie codante en majuscules) est la valeur

$$\begin{aligned} & t_{0,C^0} \times e_0(A) \times t_{0,0} \times e_0(A) \times t_{0,0} \times e_0(C) \times t_{0,1} \times e_1(A) \times t_{1,1} \times e_1(G) \\ & = 0.8 \times 0.35 \times 0.9 \times 0.35 \times 0.9 \times 0.25 \times 0.1 \times 0.2 \times 0.7 \times 0.3. \end{aligned}$$

ii) Estimation des paramètres d'un HMM

Pour pouvoir faire de la prédiction de régions codantes avec un HMM, il faut déterminer les paramètres le définissant à l'aide d'un jeu de données constitué d'un ensemble de séquences génomiques. Deux cas de figures sont possibles, selon si la structure des gènes de ce jeu de données est connue ou non.

ANNOTATION CONNUE On cherche à caractériser un modèle de type HMM pour construire un prédicteur de gènes pour un génome donné. Imaginons que l'on dispose d'un jeu de données d'une ou plusieurs séquences que l'on considère représentatives de l'espèce considérée. On souhaite déterminer les valeurs des paramètres du HMM (définition 3) qui maximisent la vraisemblance des données observées en fonction du modèle considéré. Pour une séquence unique de longueur L, ces valeurs sont données par

$$e_j(b) = \frac{\sum_{i=1}^{L}(x_i = b, \pi_i = j)}{\sum_{i=1}^{L}(\pi_i = j)} \quad \text{et} \quad t_{j,j'} = \frac{\sum_{i=1}^{L-1}(\pi_i = j, \pi_{i+1} = j')}{\sum_{i=1}^{L-1}(\pi_i = j)} \quad (2.5)$$

Remarque 9 *Dans ce cas simplifié, les probabilités d'émission des différents nucléotides dépendent uniquement de la position courante. Les état sont donc ici caractérisés par des modèles de Markov d'ordre $k = 0$. Il est possible de prendre en compte les nucléotides précédents en estimant au lieu de simples $e_j(b)$ des $e_j(b|w^k)$, soient les probabilités de trouver dans l'état j une base b après un mot w^k de longueur k. Ceci s'effectue en appliquant pour chaque état la formule d'estimation d'un modèle markovien (formule* (2.4) *page 19).*

ANNOTATION INCONNUE Dans ce cas, on dispose d'un jeu de données constitué d'une ou plusieurs séquences génomiques dont on ne connait pas *a priori* la répartition des régions codantes. Il est toutefois possible d'estimer les paramètres d'un HMM en ayant recours à un algorithme de type EM (pour *"Expectation-Maximisation"*). On peut ainsi réaliser une estimation "en ligne" sur la séquence à annoter elle-même. Cette méthode demande cependant une grande quantité de données et ne garantit pas l'optimalité des paramètres obtenus, contrairement à l'estimation précédente qui maximise la vraisemblance des données observées. On ne la trouve dans le cadre de la prédiction de gènes que pour les espèces procaryotes[15] (Nicolas *et al.*, 2002).

iii) Annotation par l'algorithme de Viterbi

Pour en revenir à la problématique qui nous intéresse, à savoir l'identification de structures de gènes, le but est de localiser les régions codantes dans une séquence nucléique donnée avec un modèle de type HMM. Si l'on fait l'hypothèse que le modèle est correct, c'est-à-dire qu'il représente au mieux la structure génique de l'espèce en question, la structure de gène la plus probable pour la séquence est donnée par la succession d'états (ou chemin) la plus probable selon le modèle. On note π^* ce chemin optimal. Il est défini par

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\ P(x, \pi)$$

Ce chemin peut être trouvé de façon récursive, par programmation dynamique en parcourant la séquence de gauche à droite. La procédure à suivre est donnée par l'algorithme de Viterbi, la version adaptée aux HMM de l'agorithme de plus court chemin de Bellman (Bellman, 1957). Imaginons que la probabilité $v_j(i)$ du chemin le plus probable arrivant à l'état j du nucléotide x_i soit connue pour tous les états j. Ces probabilités sont alors calculables pour le nucléotide x_{i+1} par

$$v_{j'}(i+1) = e_{j'}(x_{i+1}) \max_j (v_j(i) t_{j,j'})$$

Cette formule de récurrence est appliquée tout le long de la séquence pour tous les états à chaque position. À l'extrémité droite, pour retrouver le chemin optimal, il est nécessaire d'avoir stocké au préalable des "pointeurs arrière" mémorisant à chaque application de la formule l'état j identifié pour le $v_{j'}(i+1)$ considéré, comme autant de fils d'Ariane que l'on déroule qu'il y a d'états possibles. Ainsi, au terme de la procédure itérative, on identifie le chemin le plus probable par une étape dite de retour (*"trace back"*). L'algorithme 1 ci-contre détaille l'algorithme complet de Viterbi.

Pendant l'étape de retour, afficher les π^*_{i-1} successifs permet de présenter la prédiction de structure de gènes la plus probable pour la séquence à annoter en fonction du modèle.

[15] où la tâche est plus aisée du fait de la quasi-absence d'introns.

Algorithme 1 : Algorithme de Viterbi

Initialisation $(i = 0)$:
$v_0(0) = 1, v_j(0) = 0$ pour $j > 0$;
Récurrence $(i = 1, 2, \cdots, L)$:
$v_{j'}(i) = e_{j'}(x_i) \max_j (v_j(i-1)t_{j,j'})$;
$ptr_i(j') = \text{argmax}_j \ (v_j(i-1)t_{j,j'})$;
Terminaison :
$P(x, \pi^*) = \max_j v_j(L)$;
$\pi_L = \text{argmax}_j(v_j(L))$;
Retour $(i = L, L-1, \cdots, 1)$:
$\pi^*_{i-1} = ptr_i(\pi^*_i)$;

Remarque 10 *Ces calculs demandent un grand nombre de multiplications de probabilités entre elles, ce qui entraîne en pratique l'apparition de très petits nombres. Or, les arrondis qui sont effectués dessus (en raison des limites de précision des ordinateurs actuels) les altèrent considérablement, ce qui provoque des erreurs. Pour éviter ce problème, on transforme généralement les probabilités en logarithmes de probabilités, ce qui permet d'additionner les valeurs au lieu de les multiplier, sans changer les propriétés des formules.*

d. Intégration des informations : par positions

Comme nous l'avons fait pour l'approche par segments, nous allons nous intéresser pour l'approche par HMM au point qui semble représenter une limite pour la prédiction de gènes, à savoir la capacité à intégrer des informations hétérogènes.

La problématique est la même que précédemment décrite (page 41), et concerne la nécessité de calibrer les mesures provenant des sources d'informations pour les mettre sur une même échelle, et de pondérer l'importance de chacune en fonction de la confiance relative qu'on lui accorde. Les stratégies adoptées dépendent fortement de la nature des informations exploitées.

i) Intégration des longueurs : les GHMM

L'inconvénient avec les HMM de la forme sus-décrite est qu'ils ne permettent pas de prendre en compte les longueurs des exons, comme le propose simplement l'approche basée sur les segments. Pour combler cette lacune, il faut recourir à des modèles semi-markoviens ou GHMM, pour *"Generalized Hidden Markov Models"* (Rabiner, 1989; Kulp *et al.*, 1996; Burge et Karlin, 1997).

L'idée générale est de considérer que la probabilité d'une série ininterrompue de nucléotides du même état (précisément les états qui correspondent aux régions dont on souhaite modéliser la taille) n'est pas uniquement le produit des probabilités associées à chaque nucléotide. Au lieu de cela, elle dépend aussi de la longueur de cette série, ce qui revient à dire que la probabilité d'un segment est fonction de sa longueur, comme pour la méthode d'assemblage d'exons.

L'algorithme de Viterbi doit donc subir une modification pour fonctionner dans ce cadre. Par manque de place, nous ne détaillerons pas cette variante, décrite par Rabiner (1989) et Burge (1997). Ce qu'il faut en retenir, c'est qu'il est possible ainsi de prendre en compte des distributions de longueurs explicites (estimées à partir de statistiques réalisées sur un jeu de données) pour certains états. Cette version devient quadratique en temps de calculs, ce qui n'est pas supportable en pratique. Plusieurs approches ont été proposées pour remédier à ce problème, comme par Stanke et Waack (2003) ou Burge et Karlin (1997)[16].

ii) Les données d'expression

Le premier prédicteur de gènes de type HMM qui prit en compte des similarités entre la séquence génomique et des séquences exprimées fut GENIE (Kulp *et al.*, 1997). Depuis, cette information est exploitée dans de nombreux logiciels, car cela permet d'améliorer leurs performances (Salamov et Solovyev, 2000; Krogh, 2000; Yeh *et al.*, 2001; Birney *et al.*, 2004).

Le principe est toujours le même : en premier lieu, une recherche de similarités est réalisée à l'aide d'un programme d'alignement (comme le célèbre BLAST (Altschul *et al.*, 1997)) entre la séquence génomique et les séquences d'une ou plusieurs bases de données. Dans le cas de bases protéiques, la séquence génomique est auparavant traduite dans les 6 phases possibles. Puis, si une région génomique est comprise dans un alignement (étant par conséquent similaire à une partie de séquence exprimée) cette information est généralement incorporée dans le modèle en augmentant la probabilité que les nucléotides de cette région se situent dans des états codants. Chaque logiciel a sa propre formule d'intégration, et s'il ne nous semble pas opportun de les détailler individuellement, certains points généralement partagés méritent d'être soulignés :

- La contribution apportée par la similarité est fonction du pourcentage d'identité entre les séquences des régions alignées.
- Cet apport intervient généralement en modifiant les probabilités d'émission associées aux états codants.
- Des hypothèses d'indépendance sont en général nécessaires pour garantir l'exactitude du cadre théorique du HMM.

C'est notamment sur ce dernier point que l'on peut s'interroger. En effet, l'influence du contenu statistique d'une région génomique s'exerce conjointement sur la vraisemblance calculée à partir d'un modèle codant et sur la probabilité que la région présente des similarités avec des séquences codantes. Par conséquent, les deux sources d'informations ne sont pas strictement indépendantes.

[16] en limitant cette prise en compte aux états correspondant aux exons, dont la taille est bornée en pratique (voir la remarque 6 page 39).

iii) Intégration des données génomiques : les PHMM

Hormis le logiciel TWINSCAN (Korf *et al.*, 2001), les prédicteurs de gènes basés sur des HMM intègrent généralement les similarités avec des séquences génomiques grâce à des modèles PHMM, pour *"Pairwise Hidden Markov Models"*. Décrits pour l'alignement de séquences par Durbin *et al.* (1999), ils furent initialement adaptés à la détection de régions codantes par Kent et Zahler (2000), Pachter *et al.* (2002) et par Meyer et Durbin (2002).

Alors qu'un HMM permet la modélisation d'une séquence génomique unique avec plusieurs états définissant une annotation, un PHMM autorise la prise en compte simultanée de deux séquences figurant dans un même alignement, chacune bénéficiant d'une annotation distincte. La différence fondamentale est que les probabilités d'émission ne sont plus associées à un seul, mais à une paire de nucléotides (un pour chaque séquence).

L'approche par génomique comparative fait l'objet du chapitre 5, et sera donc développée plus avant. Concernant l'intégration d'information, les PHMM représentent un cadre théorique rigoureux et efficace (du point de vue des performances des logiciels). Malheureusement, cette modélisation est par définition limitée à la prise en compte d'une seule séquence d'un seul génome externe, et laisse de côté d'autres types d'information, comme par exemple les données d'expression génique.

D'autres modèles sont encore explorés, comme les EHMM (pour *"Evolutionary Hidden Markov Models"*) (Pedersen et Hein, 2003) ou les HMM phylogénétiques (Siepel et Haussler, 2004), sans pour l'instant aboutir à de résultats satisfaisants, et toujours au dépend de l'intégration des autres informations.

iv) Conclusion

Les méthodes de détection de régions codantes basées sur une approche par positions semblent présenter des défauts comparables à ceux de l'approche par segments. Bien que permettant la prise en compte individuelle de pratiquement tout type d'information, la complexité et la rigidité des modèles limite une généralisation de l'intégration.

3/ Autres méthodes

D'autres méthodes encore sont exploitées, mais étant beaucoup moins utilisées nous ne ferons que les évoquer rapidement.

i) Les grammaires

Avant l'arrivée des HMM, une approche originale fut proposée par Dong et Searls (1994) avec un modèle de structure génique basé sur des grammaires (Durbin *et al.*, 1999). Dans un tel formalisme, un gène est considéré comme une structure syntaxique organisée. Ainsi, la détection de gène dans une séquence génomique revient à faire de la reconnaissance syntaxique dans un texte. GENEDECODER (Asai *et al.*, 1998) est un autre exemple d'emploi de grammaires, dans ce cas des grammaires stochastiques. On peut noter

que les HMM et les grammaires de type stochastiques régulières sont des modélisations équivalentes.

ii) La classification

Les méthodes de classification permettent de représenter des classes d'objets, et éventuellement de prédire à quelle classe appartient un objet donné. Elles passent par la caractérisation des objets par un ensemble de variables, que l'on peut voir comme des propriétés quantifiables. L'objet concerné dans le cadre de la prédiction de gènes est en général l'exon, les classes sont "exon faux positif" et "exon vrai positif", et les variables sont des caractéristiques de type *contenu* ou *similarité*.

ANALYSE DISCRIMINANTE L'analyse discriminante est une méthode de classification utilisée en version linéaire dans FGENEH (Solovyev *et al.*, 1994; Solovyev et Salamov, 1997) ou en version quadratique dans MZEF (Zhang, 1997). Le principe de la méthode est d'utiliser les variables caractéristiques des objets pour définir un espace à plusieurs dimensions (par exemple deux coordonnées pour un plan). Ainsi, on peut représenter un objet par un point dans cet espace, et un ensemble d'objets par une distribution, ou un "nuage" de points. Si les objets appartiennent à deux classes différentes, il est possible que les points associés à une même classe soit relativement "regroupés" dans l'espace (formant par exemple deux nuages de points plus ou moins distincts). Le but de l'approche est de définir une fonction des variables qui permette de séparer au mieux les deux classes de points (une droite qui sépare les deux nuages dans notre exemple). Ainsi, lorsque l'on observe un nouvel objet dont on ignore la classe d'appartenance, on peut prédire celle-ci en appliquant la fonction sur ses variables (de quel côté est le point le représentant par rapport à la droite). Cette fonction est la fonction discriminante, dans le cas de FGENEH il s'agit d'une combinaison linéaire des valeurs, et dans MZEF une fonction de forme quadratique.

LES ARBRES DE DÉCISION Autre méthode de classification, ils permettent de définir une procédure de planification d'un ensemble de tests, représentée sous la forme d'un arbre. À chaque nœud de l'arbre correspond le test d'une condition impliquant une ou plusieurs variables, le résultat du test (en général de type binaire oui/non) détermine quelle branche descendante doit être suivie, menant à un autre test, et ainsi de suite. À chaque feuille de l'arbre est attribuée une catégorie, et lorsque l'on cherche à laquelle appartient un objet donné, on applique sur les variables le caractérisant les tests jusqu'à l'arrivée à une feuille, qui indique la classe appropriée. Comme exemple d'utilisation d'arbres de décision, on peut citer MORGAN (Salzberg *et al.*, 1998a) et GLIMMERM (Majoros *et al.*, 2003)

LIMITES DE LA CLASSIFICATION Les méthodes de classification ne doivent pas être considérées sur un même niveau que l'assemblage d'exons et les HMM. En effet, bien qu'elles permettent de combiner plusieurs sources d'in-

formations (les variables), elles ne répondent pas au même problème, à savoir identifier la meilleure structure génique possible. Elle peuvent servir en revanche à déterminer si un exon potentiel est codant ou non, par exemple pour filtrer des candidats avant de procéder à une méthode d'assemblage d'exons. On peut bien sûr envisager une application des classifications pour faire de même avec des prédictions de gènes entiers, pour déterminer s'ils semblent corrects. Cependant, l'aspect combinatoire des gènes possibles pose problème. En effet, s'il est possible de considérer de façon exhaustive tous les exons potentiels d'une séquence (dont le nombre est une fonction linéaire de la taille, voir la remarque 6 page 39) afin de les classer dans les catégories de faux positifs et de vrais positifs, il est impossible de faire de même avec les gènes possibles, dont le nombre augmente trop rapidement.

iii) Les réseaux bayésiens

Les réseaux bayésien sont des modèles probabilistes permettant une représentation graphique de dépendances entre des variables aléatoires (Pearl, 1988). Cette approche récente dans le domaine (Pavlovic *et al.*, 2002) constitue un formalisme flexible et puissant pour faire de l'inférence, c'est-à-dire déterminer des valeurs de probabilité pour les variables contenues dans le réseau, et permet comme pour les HMM de recourir à des méthodes d'estimation des paramètres du modèle (dans ce cas des probabilités conditionnelles). Notons que les HMM représentent un cas particulier de réseaux bayésiens.

Pour l'instant, ils ne sont utilisés que dans le cadre de l'approche de détection de gènes par combinaison de prédictions d'autres logiciels (Pavlovic *et al.*, 2002; Yada *et al.*, 2003; Zhang *et al.*, 2003). Cet usage paraît tout naturel au vu du principe même des réseaux bayésiens, dédiés à des systèmes de combinaison d'experts. L'application de cette méthode pour la fusion de prédicteurs de gènes offre un formalisme et des résultats bien plus satisfaisants que les procédures basées sur de simples opérateurs logiques[17] comme avec (Murakami et Takagi, 1998; Rogic *et al.*, 2002; Tech et Merkl, 2003).

iv) Les graphes

À notre connaissance, deux logiciels seulement utilisent un modèle basé sur un graphe . Il s'agit de DAGGER (Chuang et Roth, 2001) et de EUGÈNE (Schiex *et al.*, 2001), tous deux utilisant des graphe appelés DAG (pour *"Directed Acyclic Graph"*) sur lesquels nous reviendrons abondamment dans la suite du document. L'approche de DAGGER a cela d'intéressant qu'elle propose une procédure d'estimation de ses paramètres de pondération par un processus d'optimisation. Malheureusement, certaines contraintes limitent l'application de cet outil (une longueur maximum est requise pour tous les états, le modèle ne peut considérer plusieurs gènes par séquence ou des gènes partiels).

[17] que l'on peut également considérer comme des cas particuliers de réseaux bayésiens.

IV Analyse

1/ Performance des logiciels existants

a. Comment évaluer ?

Pour évaluer un logiciel de prédiction de gènes, il faut disposer d'un jeu de données fiable de séquences annotées par des experts, où l'on connait précisément les positions des exons. Le logiciel testé est lancé sur le jeu d'évaluation, puis les prédictions obtenues sont comparées à l'annotation de référence. Les mesures de référence des performances furent clairement établies lors de la première évaluation comparative à grande échelle de prédicteurs de gènes par Burset et Guigó (1996). Depuis, ces critères sont largement utilisés, que ce soit dans chaque présentation individuelle d'un logiciel ou dans les projets d'évaluations indépendantes entrepris depuis (Reese *et al.*, 2000b; Pavy *et al.*, 1999b; Kraemer *et al.*, 2001; Guigó *et al.*, 2000; Rogic *et al.*, 2001).

i) Les critères

Considérons un objet prédit par un logiciel. S'il figure également dans l'annotation de référence, on le qualifie de "vrai positif". Sinon, il s'agit d'un "faux positif". D'un autre côté, un objet qui figure dans l'annotation de référence mais qui est absent des prédictions est appelé "faux négatif".

Définition 4
Soient respectivement VP, FP et FN le nombre total de vrais positifs, faux positifs et faux négatifs d'une prédiction pour une annotation de référence donnée. La sensibilité SN et la spécificité SP sont données par les formules

$$SN = \frac{VP}{VP + FN} \qquad SP = \frac{VP}{VP + FP}$$

En d'autres termes, la sensibilité reflète le pourcentage d'objets annotés qui sont correctement détectés, et la spécificité le pourcentage de prédictions qui sont correctes.

ii) Les niveaux

Dans le domaine de l'identification des gènes, ces critères se mesurent généralement selon trois niveaux de référence, selon l'objet considéré : le niveau nucléotide, le niveau exon, et le niveau gène. Un exon prédit n'est considéré comme vrai positif que s'il est strictement identique à l'exon annoté, ce qui implique que les frontières de début et de fin soient les mêmes. De la même façon, un gène prédit n'est un vrai positif que si tous les exons de la prédiction et de l'annotation sont identiques.

b. Les résultats

Lorsque l'on analyse des résultats d'évaluations de prédicteurs de gènes, il est important de garder à l'esprit certaines considérations. En effet, de nom-

breux facteurs influencent les performances des logiciels (Burset et Guigó, 1996; Rogic *et al.*, 2001; Guigó *et al.*, 2000; Korf, 2004) :

- Concernant le jeu de test. Les performances des logiciels sont sensibles à certains critères du jeu d'évaluation (essentiellement la proportion de régions codantes dans le génome, la moyenne ou la variabilité du *GC*% et le nombre moyen d'exons par gènes) qui peuvent varier considérablement d'une espèce à l'autre, voire au sein d'une même espèce d'un jeu de données à l'autre.
- Concernant le logiciel. Outre bien sûr la quantité/qualité des informations utilisées et l'efficacité des méthodes sous-jacentes, d'autres critères influencent les performances et sont généralement associés à un numéro de version du logiciel. Par exemple, le jeu de données qui a servi pour l'entrainement ou l'estimation des paramètres, les divers réglages internes (paramètres empiriques), ou la version des bases de données de séquences exprimées (pour les logiciels utilisant ce type d'information).

Les très nombreuses différences entre les logiciels ont pour conséquence de rendre difficile une interprétation simple des évaluations. À quoi peut-on attribuer un écart de performance ? Au choix des informations utilisées, à la qualité du modèle théorique, au paramétrage, ou simplement à la simple variabilité dûe au jeu de test ?

Néanmoins, certaines estimations générales se dégagent des évaluations. À ce jour, on peut considérer que les meilleurs outils atteignent des valeurs de sensibilité au niveau gène aux alentours de 50% ou plus en approche *ab initio* (Korf, 2004; Schiex *et al.*, 2001) et de 75% en combinant de nombreux types d'information (Allen *et al.*, 2004; Schiex *et al.*, 2001), ceci pour une espèce dont le génome est relativement compact en gènes comme *Arabidopsis thaliana*, et sur des jeux de données expertisés et donc très informés. Les chiffres sont en revanche beaucoup moins élevés pour une espèce à faible proportion d'exons, comme *Homo sapiens*, chez qui même avec des configurations massives d'information les logiciels ne détectent correctement que 10 à 20% des gènes (Parra *et al.*, 2003; Flicek *et al.*, 2003).

Par conséquent, même si les performances des logiciels sont en constante amélioration (Burset et Guigó, 1996; Brent et Guigó, 2004), des progrès considérables restent à réaliser avant de considérer que les prédictions de gènes produites par les logiciels sont fiables.

2/ Evolution du domaine

Evolution des informations Suite aux progrès de la génomique, et notamment des projets de séquençage massif de génomes et de transcriptomes, la variété, la quantité et la qualité des données disponibles n'ont cessé d'augmenter durant ces dernières années.

Comme nous avons vu, la conséquence de cette évolution se traduit par une tendance générale des outils vers l'intégration massive des informations.

RIGIDITÉ DES MÉTHODES En revanche, il semble que les méthodes utilisées limitent les capacités d'intégration des logiciels. En effet, pour intégrer une information d'un nouveau type, il n'est pas rare de recourir au développement d'un nouveau logiciel spécifique, voire au développement d'un modèle de gène complètement nouveau.

Pour illustrer cette idée, prenons un exemple dans les méthodes probabilistes. Le logiciel GENSCAN (Burge et Karlin, 1997) est un bon représentant des prédicteurs de type *ab initio* basés sur des HMM (plus précisément GHMM, voir page 49), n'exploitant que des informations intrinsèques. Pour intégrer des informations de séquences exprimées (en l'occurrence de protéines), un nouveau logiciel, GENOMESCAN (Yeh *et al.*, 2001), fut développé à partir du même modèle, en modifiant notamment les probabilités d'émission pour les états codants. Les informations de similarités avec des séquences génomiques furent également exploitées, mais à partir de GENSCAN (et non pas de GENOMESCAN, ce qui aurait poursuivi la fusion d'informations). Une des solutions fut de modifier également les probabilités des régions codantes en fonction des séquences similaires détectées, donnant le logiciel TWINSCAN (Korf *et al.*, 2001). Une autre alternative fut de changer complètement le modèle sous-jacent, pour passer à un PHMM (voir page 51), créant par exemple le logiciel DOUBLESCAN (Meyer et Durbin, 2002). Au bout du compte, s'il est clair que l'approche *ab initio* reste en deçà des autres au niveau des performances, on aboutit à 3 autres logiciels distincts, présentant chacuns ses forces et ses faiblesses propres, sans qu'aucun ne puisse exploiter l'ensemble des informations.

Les dernières méthodes en date n'échappent pas à la règle : les récents HMM phylogénétiques (Siepel et Haussler, 2004) proposent des modèles particulièrement sophistiqués pour capter des informations d'évolution moléculaire à partir de plusieurs séquences génomiques. Cependant, l'incorporation de toute autre information (distribution de longueur des régions, séquences exprimées) s'en trouve compromise.

V CONCLUSION

Cette large étude de l'état de l'art du domaine de la prédiction de gènes nous a permis de mettre en évidence les points suivants :

- Les performances des logiciels de prédiction de gènes ne sont pas encore satisfaisantes à ce jour.
- Les avancées du domaine de la génomique produisent régulièrement de nouveaux types d'information.
- L'intégration adéquate de nouvelles sources d'information permet généralement d'augmenter la qualité des prédictions.
- La grande majorité des modèles de gènes existants nous semblent limités en terme de capacité d'intégration.

Face à l'accumulation des données issues de la génomique qui représentent un potentiel d'information considérable, il semblerait que le domaine de re-

cherche de l'identification de structure de gènes dans les génomes ne progresse pas de concert. Nous pensons qu'une des raison majeure à cela réside dans la rigidité des modèles utilisés par la plupart des logiciels de prédictions.

Dans le but de permettre d'une part l'intégration de l'ensemble des informations actuellement disponibles, et d'anticiper d'autre part sur les évolutions futures de la génomique, il nous semble que la stratégie optimale est de se baser sur un logiciel comprenant :

- Un modèle de gènes souple et évolutif qui permette l'incorporation d'informations diverses.
- Une procédure de paramétrisation fiable pour exploiter au mieux les informations choisies, et réutilisable pour permettre l'adaptation du logiciel à l'évolution des données produites par les projets de séquençage de génomes et transcriptomes.

Durant la suite de ce travail, nous allons adopter cette stratégie et démontrer qu'elle est efficace pour l'amélioration des qualités de prédictions de gènes par intégration de nouveaux types d'information.

It has not escaped our notice that the specific pairing we have postulated immediately suggests a possible copying mechanism for the genetic material. [...] We have also been stimulated by a knowledge of the general nature of the unpublished experimental results and ideas of Dr. M. H. F. Wilkins, Dr. R. E. Franklin and their co-workers

— Watson et Crick (1953)

This work was supported in part by a Grant-in-Aid [...] from the Ministry of Education, Science, Sports and Culture of Japan.

— Gotoh (2000)

I chose to use quadratic discriminant analysis [...] as the basis for a new program called MZEF (Michael Zhang's Exon Finder).

— Zhang (1997)

We wish to thank Chris Burge for answering email queries about GENSCAN *and offering to let us see the source code (although we declined this generous offer).*

— Korf *et al.* (2001)

There is no fundamental limit to the length of sequences which GeneParser can analyze. The current version has been tested on sequences as long as 73,000 nucleotides. [...] the run time increases with the square of the sequence length.

— Snyder et Stormo (1995)

The work in this paper is completely my own.

— Korf (2004)

Chapitre 3

Le logiciel EuGène

Ce chapitre est consacré à la présentation du logiciel EuGène (Schiex *et al.*, 1999, 2001), qui a servi de base aux travaux de cette thèse. EuGène est un programme de prédiction de gènes initialement destiné à l'annotation du génome de la plante *Arabidopsis thaliana* (l'arabette).

Comme tout prédicteur de gènes, son objectif est d'identifier pour une séquence génomique donnée (dont les bases sont numérotées dans le sens de lecture) les positions nucléotidiques de début et de fin de chacun des exons qu'elle contient. Et à l'instar des méthodes existantes, EuGène utilise un modèle de gène pour représenter l'ensemble des prédictions possibles, diverses sources d'information paramétrant ce modèle, et un algorithme de programmation dynamique qui lui permet d'identifier la meilleure prédiction en fonction des informations disponibles.

Dans un premier temps, nous verrons comment l'espace des prédictions est modélisé par un graphe , quelles sont les sources d'information exploitées et comment elles sont intégrées dans le graphe, puis comment identifier la meilleure prédiction possible en utilisant un algorithme de programmation dynamique. Enfin, nous terminerons ce chapitre par une analyse du logiciel concernant ses propriétés et performances, et par les développements nécessaires à la réalisation des objectifs de la thèse.

I Le graphe d'EuGène

L'originalité majeure d'EuGène réside dans son modèle de gène. Alors que la plupart des logiciels récents sont basés sur des HMM (voir page 44), le modèle s'articule ici sur un graphe orienté sans circuit ou DAG (pour *"Directed Acyclic Graph"*) chargé de représenter l'ensemble des prédictions de structure génique pour une séquence génomique donnée.

1/ Définitions préliminaires

Définition 5 (Graphe orienté)
Un graphe orienté $G = \{\mathcal{V}, \mathcal{E}\}$ est défini par :

- *un ensemble $\mathcal{V}$ fini de sommets (*"vertices"*)*
- *un ensemble $\mathcal{E}$ d'arcs orientés (*"edges"*), chaque arc ayant une origine et une cible dans $\mathcal{V}$. Un arc $e = (v_0, v_1)$ a pour origine v_0 et pour cible v_1. On dit que v_1 est un successeur de v_0 et que v_0 est un prédécesseur de v_1.*

Définition 6 (Chemin, circuit)
Soit un graphe orienté $G = (\mathcal{V}, \mathcal{E})$. Un chemin $u \rightarrow u'$ de longueur l d'un sommet u appelé extrémité initiale vers un sommet u' appelé extrémité terminale est une série $\langle v_0, \ldots, v_l \rangle$ telle que $v_0 = u, v_l = u'$ et $\forall i, 0 \leq i \leq l-1, (v_i, v_{i+1}) \in \mathcal{E}$.

Un circuit ou cycle est un chemin dont l'extrémité initiale et l'extrémité terminale sont confondues.

Définition 7 (DAG)
Un DAG (pour "Directed Acyclic Graph"*), ou graphe orienté acyclique, est un graphe orienté sans circuit.*

À tout arc peut être associée une valeur numérique que nous appellerons poids. Un DAG peut être schématisé par une figure contenant des points représentant les sommets, dont certains sont reliés par des flèches représentant les arcs.

2/ Structure du graphe

a. Description générale

Globalement, le graphe d'EuGène est composé de pistes horizontales représentant les diverses annotations possibles pour chaque nucléotide (d'une façon comparable aux états des HMM) et d'arcs obliques permettant des transitions entre pistes (comme les transitions entre les états des HMM). Chaque nucléotide est représenté par des sommets dans le graphe, l'orientation de la séquence se fait dans le sens conventionnel 5'→3', celle des arcs de gauche à droite, et le nombre de sommets constituant les pistes est proportionnel à la taille de la séquence à annoter. Le principe fondamental du graphe d'EuGène est que toute structure génique cohérente de la séquence d'ADN peut être représentée par un chemin traversant le graphe d'une extrémité à l'autre. Le passage d'un tel chemin sur une piste donnée au niveau d'un nucléotide indique que la structure de gène correspondant à ce chemin attribue à ce nucléotide l'annotation associée à la piste en question.

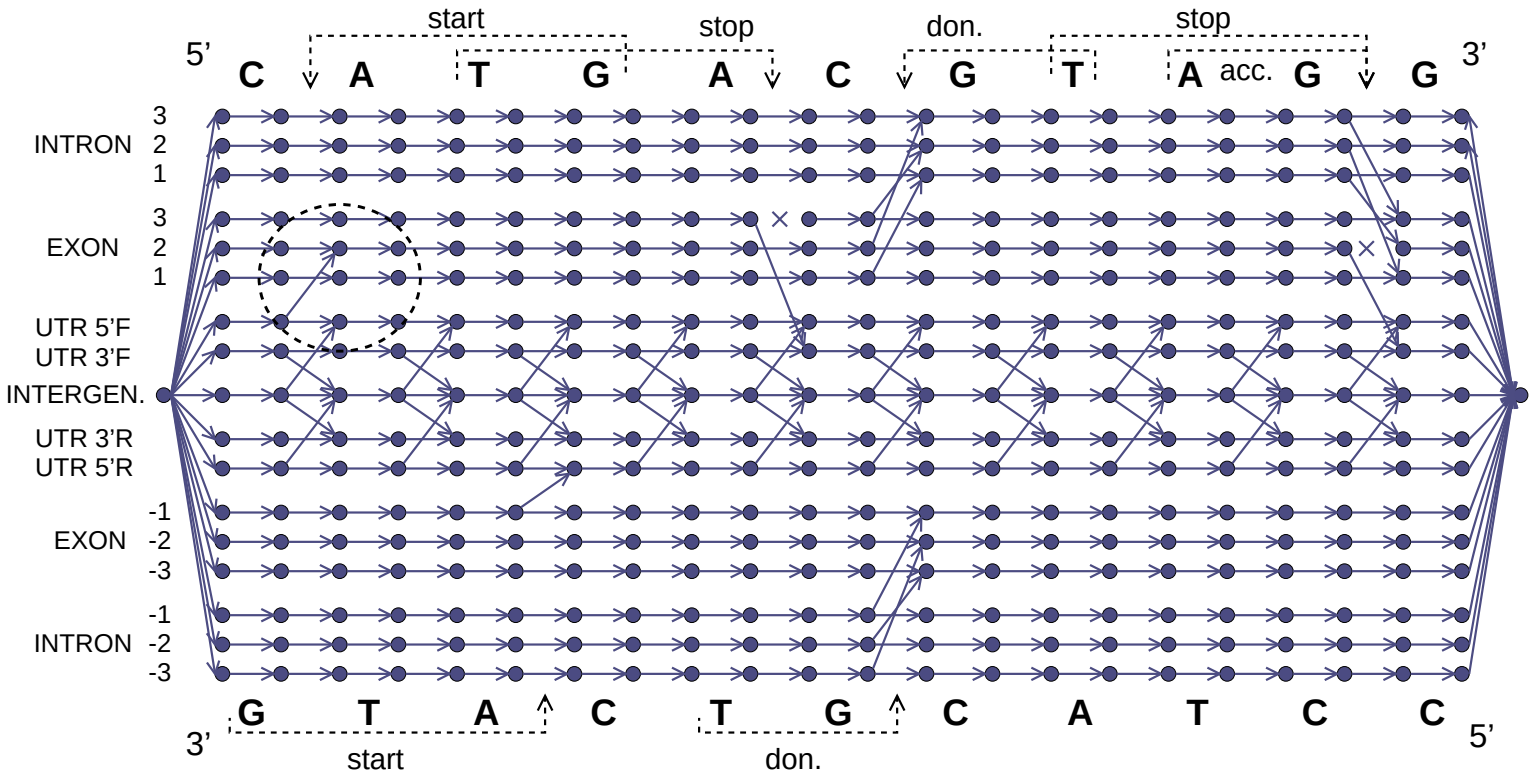

FIG. 3.1 – Le graphe orienté sans circuit d'EUGÈNE. La séquence génomique est indiquée en haut (brin sens), la séquence du bas étant l'inverse-complémentaire (brin anti-sens). Les pistes horizontales correspondent aux annotations possibles pour chaque nucléotide (légendées à gauche), qui sont de haut en bas pour le brin sens : intron dans les 3 phases (la phase d'un intron est fonction de la position du site donneur d'épissage par rapport au dernier codon de l'exon précédent), exon dans les trois phases, UTR5' et UTR3' (UTR5F et 3F pour le brin sens *"Forward"*), et la piste intergénique au milieu et unique (pas de distinction des brins). La moitié inférieure reprend les même annotations pour le brin anti-sens (avec un signe "-" devant les phases et la lettre R après les UTR pour *"Reverse"*). Pour chaque piste, deux sommets (les points) sont créés pour chaque nucléotide. Les flèches horizontales représentent les arcs de type *contenu* (au niveau de chaque nucléotide) et les arcs de type *signal* (entre les nucléotides). Les signaux biologiques permettant le passage d'une piste à l'autre par la création d'arcs *signal* supplémentaires obliques sont indiqués par des flèches pointillées sur la séquence. Aux extrémités du graphe figurent les sommets G et D. La partie entourée en pointillés est détaillée en Fig. 3.2 page 63.

b. Description formelle

Définition 8 (Le DAG d'EuGène)
Soit $\mathcal{S}$ une séquence génomique donnée de longueur L (chaque nucléotide étant numéroté dans l'ordre de lecture de la séquence) et J un nombre d'annotations nucléotidiques possibles. Le graphe $\mathcal{G}$ construit dans EuGène *relatif à $\mathcal{S}$ et J est défini par :*

- *un ensemble de sommets g_i^j et d_i^j pour tout $1 \leq i \leq L$ et tout $1 \leq j \leq J$ (gauche et droite).*
- *un ensemble d'arcs de type* contenu *c_i^j d'origine g_i^j et de cible d_i^j pour tout $1 \leq i \leq L$ et tout $1 \leq j \leq J$.*
- *un ensemble d'arcs de type* signal *s_i^j d'origine d_i^j et de cible g_{i+1}^j pour tout $1 \leq i \leq L-1$ et tout $1 \leq j \leq J$.*
- *un ensemble d'arcs de type* signal *$s_i^{j,j'}$ pour tout signal biologique détecté dans la séquence et permettant un changement d'annotation de j à j' entre les nucléotides de position i et $i+1$. Par exemple, au niveau d'un codon start `ATG` détecté par une des sources d'information (voir plus bas), un arc partant du sommet droit de la piste UTR5' rejoint le sommet gauche suivant de la piste exon de la phase appropriée, comme illustré en Fig. 3.2 ci-contre).*
- *deux sommets spéciaux G et D servant respectivement d'origine et de cible à des arcs spéciaux s_1^j et s_L^j de cible et d'origine g_1^j et d_L^j pour tout $1 \leq j \leq J$.*
- *un ensemble de poids C_i^j, S_i^j et $S_i^{j,j'}$ attribués respectivement à tout arc c_i^j, s_i^j et $s_i^{j,j'}$.*

La structure du graphe d'EuGène est illustrée pour une courte séquence d'exemple en Fig. 3.1 page précédente.

c. Chemins et prédictions

Définition 9
Soit $\mathcal{G}$ un graphe construit par EuGène *pour une séquence génomique donnée $\mathcal{S}$ de longueur L et un ensemble de J annotations. Un chemin π traversant le graphe avec $\langle \pi_1 = j_1, \pi_2 = j_2, \cdots, \pi_L = j_L \rangle$ est un chemin tel que :*

- *les extrémités inititiales et terminales de π sont respectivement G et D.*
- *π contient les arcs de type* contenu *$\langle c_1^{j_1}, c_2^{j_2}, \cdots, c_L^{j_L} \rangle$.*

Son poids est défini comme étant la somme des poids de tous les arcs qui le composent.

Il définit une prédiction de de structure génique telle qu'à chaque nucléotide x_i de $\mathcal{S}$ est attribuée l'annotation π_i. Le poids de la prédiction est égal au poids de π.

Le chemin de poids minimum traversant $\mathcal{G}$ est le chemin optimal[1]*. Il est noté $\pi^*(\mathcal{G})$, et définit la prédiction de gènes optimale.*

[1] bien qu'il soit théoriquement possible que plusieurs chemins puissent avoir le poids minimum, nous considérons négligeable la probabilité d'un tel évènement, comme il est fait dans tous les autres logiciels.

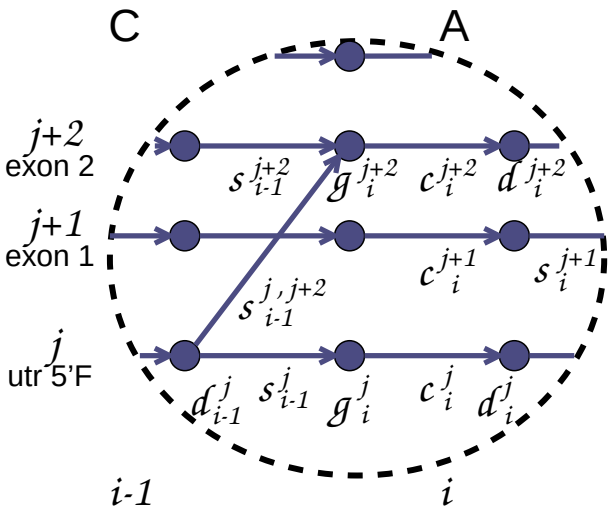

FIG. 3.2 – Détail du graphe d'EUGÈNE et de son algorithme. La région agrandie correspond aux deux premiers nucléotides de la Fig. 3.1 page 61, indiqués en haut. Pour cet exemple, les indices i et j correspondent respectivement à la position du `A` du premier `ATG` et à la piste UTR5F. Les arcs c de type *contenu* relient les sommets g aux sommets d de la même piste et les arcs s de type *signal* relient les sommets d à un ou plusieurs sommet g de la position suivante. Ici, la présence du codon start permet le passage du sommet d_{i-1}^{j} en UTR5' au sommet g_i^{j+2} en exon de la phase +2. L'algorithme de programmation dynamique d'EUGÈNE détermine pour chaque sommet d quel est le précédent sommet d dans le chemin optimal venant de l'extrémité initiale G. Dans cet exemple, à la position i pour la piste j le meilleur chemin arrivant à d_i^j a un poids de $W_i^j = (W_{i-1}^j + S_{i-1}^j + C_i^j)$ (unique possibilité). Pour la piste $j+2$, le chemin optimal menant à d_i^{j+2} reçoit un poids égal soit à $(W_{i-1}^{j+2} + S_{i-1}^{j+2} + C_i^{j+2})$, soit à $(W_{i-1}^{j} + S_{i-1}^{j,j+2} + C_i^{j+2})$, selon lequel est le moins élevé (voir la formule de récurrence (3.3) page 69).

Remarque 11 *L'hypothèse est faite qu'à une position nucléotidique donnée ne peut correspondre qu'une seule annotation. Toutefois, il arrive* in vivo *qu'un même nucléotide fasse exceptionnellement partie de plusieurs structures fonctionnelles (cas de gènes partiellement chevauchants par exemple). Bien que ces cas puissent être pris en compte en ajoutant quelques pistes dans le graphe , ils sont ignorés par les prédicteurs du fait de leur faible fréquence (le cas précis de l'épissage alternatif est abordé dans le chapitre 6).*

L'objectif est donc d'identifier π^* parmi tous les chemins possibles, ce qui produit la prédiction de gènes optimale.

3/ Quelques propriétés du graphe

i) Espace de recherche

Puisque toute structure génique cohérente peut être représentée par un chemin traversant le graphe , celui-ci permet de modéliser l'ensemble des prédictions de gènes pour la séquence à annoter. Notre espace de recherche est donc défini par l'ensemble des chemins possibles.

ii) Taille du graphe

Le nombre de pistes étant constant, le nombre de sommets augmente linéairement avec la taille de la séquence. Il en va de même pour le nombre maximum d'arcs, qui ne peuvent relier que des sommets correspondant à des positions nucléotidiques identiques ou adjacentes.

iii) Conservation du chemin optimal

Cette propriété est utile pour intégrer des informations dans le modèle par pondération d'arcs, comme nous allons le voir plus loin. Soit $\mathcal{G}$ le graphe d'EuGène pour une séquence donnée. Considérons une opération $\Delta^+(\mathcal{G}, i, j, \delta)$ qui réalise $C_i^j = C_i^j + \delta$, et une opération $\Delta^-(\mathcal{G}, i, j, \delta)$ qui réalise $C_i^{j'} = C_i^{j'} - \delta$ pour tout $1 \leq j' \leq J$ avec $j' \neq j$.

$$\begin{array}{ll} \text{Si} & \mathcal{G}^+ = \Delta^+(\mathcal{G}, i, j, \delta) \text{ et } \mathcal{G}^- = \Delta^-(\mathcal{G}, i, j, \delta) \\ \text{Alors} & \pi^*(\mathcal{G}^+) = \pi^*(\mathcal{G}^-) \qquad \forall i_{1 \leq i \leq L} \text{ et } \forall j_{1 \leq j \leq J} \end{array}$$

Ceci revient à dire que pour un graphe donné, quelle que soit la position nucléique considérée, augmenter le poids d'une piste ou diminuer d'autant toutes les autres ne change pas le chemin optimal du graphe produit[2].

II Les informations utilisées

Initialement, tous les arcs se voient attribuer un poids nul. Pour permettre une distinction entre les annotations plus ou moins satisfaisantes,

[2] la propriété est même plus forte, car cela ne modifie pas l'ordre des chemins s'ils sont triés par poids.

EuGène exploite des sources d'information, celles-là même que nous avons vues dans le chapitre précédent. En jouant sur le poids de certains arcs, le principe consiste à pénaliser ou favoriser certaines prédictions en fonction des informations disponibles.

1/ Informations de type *contenu*

Afin d'estimer pour chaque nucléotide la vraisemblance qu'il appartienne à chacun des types d'élément génomique en fonction de son contexte, EuGène utilise des IMM, ou modèles de Markov interpolés, dont l'ordre s'adapte à la quantité de données disponibles pour l'estimation des paramètres (Salzberg *et al.*, 1998b, voir aussi en page 23 de ce document). Les exons bénéficient d'un modèle 3-périodique (fonction de la phase), et chaque autre élément (intron, UTR5', UTR3' et intergénique) d'un simple IMM. Pour l'arabette, les modèles sont estimés à partir d'un jeu de données de séquences annotées du nom d'AraClean (Korning *et al.*, 1996). Concernant la difficulté d'estimer un modèle pour l'intergénique (voir page 20), l'hypothèse est faite ici qu'il est proche de l'intronique, avec la différence que les introns sont transcrits et donc orientés. L'intergénique n'étant pas soumis à une telle contrainte d'orientation, son modèle est construit par mélange du modèle intron et de son inverse-complémentaire.

La pondération des arcs de type *contenu* s'effectue de la façon suivante. Pour toute position i et pour toute piste j disposant d'un IMM M^j, le poids de l'arc c_i^j est

$$C_i^j = -\log P(x_i^j | M^j, x_{i-1}^j, \cdots, x_{i-k}^j) \tag{3.1}$$

où $P(x_i^j | M^j, x_{i-1}^j, \cdots, x_{i-k}^j)$ est la probabilité du nucléotide x_i selon le modèle M^j et les k nucléotides précédents (ordre de la chaîne de Markov). On a donc un poids $C_i^j \in [0, +\infty]$.

2/ Informations de type *signal*

a. Codon start et sites d'épissage

Pour créer et pondérer les arcs de type *signal*, EuGène peut utiliser des logiciels spécifiques de détection de signaux. Ces logiciels sont lancés sur la séquence à annoter, et les prédictions produites en sortie sont ensuite intégrées dans le graphe . Elles se présentent généralement sous la forme d'un ensemble de scores σ, chacun étant associé à la position d'un signal potentiel. Bien que censés refléter la probabilité que le site désigné soit effectivement fonctionnel, ces scores ne sont pas intégrés directement dans le graphe . Ils passent par une étape de mise à l'échelle, à l'issue de laquelle les arcs $s_i^{j,j'}$ (permettant une transition entre pistes) et s_i^j (poursuite sur la même piste) sont pondérés ainsi

$$\begin{cases} S_i^{j,j'} = & -\log(\alpha\sigma^\beta) \\ S_i^j = & -\log(1-\alpha\sigma^\beta) \end{cases} \tag{3.2}$$

où σ est le score proposé par le logiciel (avec $0 \leq \sigma \leq 1$) et α et β sont des paramètres de mise à l'échelle que l'on attribue au logiciel considéré.

Ainsi, chaque source d'information est pondérée de façon spécifique par des paramètres propres, qui reflètent d'une certaine façon son importance et donc la confiance qui lui est accordée. À l'origine, EuGène utilisait les logiciels NetStart (Pedersen et Nielsen, 1997) pour la prédiction des codons `ATG` de démarrage de traduction, NetGene2 (Tolstrup *et al.*, 1997) et de SplicePredictor (Brendel et Kleffe, 1998) pour les sites donneurs et accepteurs d'épissage.

Remarque 12 *Pour le cas des logiciels de détection de sites d'épissage, deux paires distinctes de paramètres α et β sont attribuées afin de considérer séparément les prédictions de sites donneurs et de sites accepteurs (les deux classes n'étant pas forcément identifiées avec la même précision).*

La détermination des paramètres α et β pour chaque source d'information s'effectue par optimisation des performances d'EuGène sur un jeu de données d'apprentissage, et sera présentée en détail dans le chapitre 4.

b. Codon stop et signaux de transcription

La prédiction des codons stop est basique, elle se fait par simple lecture et localisation sur la séquence des triplets `TAA`, `TAG` et `TGA`. Puisque l'on considère qu'un codon stop à une position i provoque toujours l'arrêt de la traduction quelque soit son contexte nucléotidique, l'arc permettant de continuer dans la piste exonique j de la phase correspondante est tout simplement supprimé (par l'attribution d'une pénalité infinie, ce qui donne $S_i^j = -\infty$). Le passage obligatoire vers l'UTR3' de la piste j' est pondéré par une pénalité constante z, on a donc $S_i^{j,j'} = z$.

Concernant les signaux de démarrage et d'arrêt de transcription, il n'existe à ce jour aucun logiciel qui les prédit de façon fiable[3]. Par conséquent, les passages permettant de signaler le début ou la fin d'un transcrit sont considérés possibles à toutes les positions nucléotidiques, et pondérés par une constante y, ce qui donne formellement $S_i^{j,j'} = y$, et $S_i^j = 0$ pour tout i, et pour chacun des couples j et j' suivants : intergénique→UTR5F et UTR5R→intergénique pour le début d'un transcrit (brin sens et anti-sens repectivement), et UTR3F→intergénique et intergénique→UTR3R pour la fin d'un transcrit (également brin sens et anti-sens, voir la Fig 3.1 page 61).

[3]la localisation des promoteurs est un exercice difficile et peu précis jusqu'à présent, et les signaux de poly-adénylation sont très peu conservés chez *A. thaliana*, comparativement à ceux d'*H. sapiens* par exemple.

3/ Informations de type *similarité*

Comme nous avons vu dans le chapitre précédent, il est intéressant de pouvoir exploiter des informations de type *similarité* (page 29). C'est pourquoi EuGène dispose d'un moyen d'intégration de similarités entre la séquence génomique et des séquences de molécules exprimées (ARNm et protéines).

L'idée est dans un premier temps de rechercher à l'aide de logiciels d'alignement spécifiques dans les bases de données d'éventuelles séquences présentant des similarités avec des régions de la séquence génomique. Dans un second temps, le graphe d'EuGène est modifié en diminuant ou augmentant le poids de certains arcs. L'objectif est de favoriser les chemins produisant une prédiction en accord avec les informations détectées.

a. ARNm

Les séquences transcrites présentant des similarités avec la séquence génomique sont tout d'abord recherchées dans des bases de données de transcrits comme dbEST (Boguski *et al.*, 1993) par le biais de logiciels d'alignements comme Blast (Altschul *et al.*, 1990) et Sim4 (Florea *et al.*, 1998). Ces logiciels produisent une sortie comportant un ensemble de transcrits alignés sur la séquence génomique. Il faut savoir que dans un alignement entre deux séquences, on distingue trois types de qualificatifs possibles pour une position donnée : si les deux bases de cette position sont identiques, il s'agit d'un appariement (*"match"*), si les deux bases sont différentes, il s'agit d'un mésappariement (*"mismatch"*), et si une base n'a pas de vis-à-vis, il s'agit d'un brèche séparant les appariements (*"gap"*). Un alignement entre une séquence génomique et une séquence transcrite est donc typiquement composé de régions d'appariements correspondant aux exons, et de régions de brèches correspondant aux introns, qui n'ont pas d'équivalent sur le transcrit mature (voir Fig. 2.3 page 30).

Comme beaucoup de séquences transcrites présentes dans les bases de données ne sont pas spécifiques et s'alignent sur plusieurs régions génomiques, des filtres de qualité sont nécessaires. Par exemple, un transcrit doit s'aligner au moins sur 80% de sa longueur totale, avec plus de 90% d'identités (moins de 10% de mésappariement). Les informations contradictoires entre transcrits sont gérées ainsi : les alignements sont classés par ordre décroissant de leur nombre d'introns détectés, et par nombre d'appariements. Puis les alignements sont considérés un par un, et si l'un d'entre eux est incompatible avec un précédent (un appariement en face d'une brèche) il est éliminé. Par conséquent, un transcrit aberrant est filtré si un autre apporte une correction. Cependant, en cas d'épissage alternatif (voir la remarque 3 page 6), seul le variant le plus épissé et le plus long est gardé.

Pour EuGène, l'intégration consiste à favoriser les prédictions en accord avec les alignements et de pénaliser celles qui sont incompatibles. Pour cela, les arcs de *contenu* sont pondérés en ajoutant à leur poids une valeur ε selon l'état de l'alignement à la position considérée : pour un appariement

(où l'on s'attend à une région transcrite), les arcs de toutes les pistes sont pénalisés, sauf pour les pistes exon et UTR. Pour une brèche, seules les pistes introniques ne sont pas alourdies. Comme le montre la propriété de conservation du chemin optimal (page 64), il est équivalent de pénaliser les états incohérents avec l'information et de favoriser les états cohérents.

À l'instar des paramètres α et β de la formule (3.2) page 66, le poids ε est estimé par un processus d'optimisation.

b. Protéines

Comme pour les transcrits, les séquences protéiques présentant des similarités avec une des 6 traductions possibles de la séquence génomique sont recherchées dans des bases de données qui sont SwissProt (Boeckmann *et al.*, 2003), PIR (Barker *et al.*, 2000), et TrEMBL (Boeckmann *et al.*, 2003). La recherche d'alignement est réalisée par le programme BlastX (Altschul *et al.*, 1990) qui effectue au préalable la traduction de la séquence génomique et recherche des alignements locaux (peu de brèche). Les alignements sont filtrés sur la base de leur taux de similarité (une *E-value*[4] inférieure à 10^{-6} est exigée).

Chaque base source de données dispose d'un paramètre ρ reflétant la confiance qu'on lui accorde et qui sert à pondérer les arcs du graphe . Pour chaque alignement, un score moyen $\overline{\sigma}$ est calculé en divisant le score donné par BlastX par la longueur de l'alignement. La gestion de la redondance entre bases de données se fait en ne considérant pour un nucléotide donné que le meilleur alignement (qui sont classés par ordre décroissant de ρ puis de $\overline{\sigma}$). Comme l'information protéique plaide en faveur du caractère codant, l'intégration dans le graphe s'effectue en soustrayant la valeur $\rho\overline{\sigma}$ du poids des arcs *contenu* de la piste exon correspondante (la phase est indiquée par l'alignement).

Les paramètres ρ des bases de données protéiques font également l'objet d'une estimation par optimisation.

III L'algorithme d'EuGène

Une fois que le graphe contient toutes les informations nécessaires sous la forme de pondérations d'arcs, l'objectif est d'identifier π^*, le chemin le moins pénalisé traversant le graphe , qui correspond à la meilleure prédiction possible. Bien entendu, une approche exhaustive n'est pas envisageable, du fait de la combinatoire des possibilités. EuGène utilise donc un algorithme de programmation dynamique de recherche de plus court chemin, appliqué ici à la recherche du plus petit poids.

Plus précisément, l'algorithme d'EuGène est une variante de l'algorithme de Bellman (Bellman, 1957; Cormen *et al.*, 1990) que l'on trouve dans

[4] valeur donnée par les programmes Blast que l'on peut considérer grossièrement comme reflétant la probabilité que la similarité entre les séquences de l'alignement soit due au hasard.

les HMM dans sa version de Viterbi (Rabiner, 1989, voir aussi en page 49 de ce document). Une modification de cet algorithme lui permet de prendre en compte des contraintes sur les longueurs minimum des éléments (Schiex *et al.*, 2001), elle n'est pas présentée ici pour des raisons de place.

1/ Description générale

Comment fonctionne cet algorithme de programmation dynamique ? Le principe s'appuie sur un raisonnement récursif :

Tout d'abord, supposons que l'on cherche à déterminer dans un DAG $G = \{\mathcal{V}, \mathcal{E}\}$ comme défini en page 60 la nature et le poids du chemin optimal π^* pour une extrémité initiale v_1 et une extrémité terminale v_n ($v_1 \rightarrow v_n$). Soit $V_{n-1} = \{v_{n-1}^1, v_{n-1}^2, \cdots, v_{n-1}^J\}$ l'ensemble des sommets précédant v_n, c'est-à-dire que $\forall j_{(1 \leq j \leq J)}$ il existe un arc $(v_{n-1}^j, v_n) \in \mathcal{E}$. Pour tout j, le poids d'un chemin optimal $v_1 \rightarrow v_{n-1}^j \rightarrow v_n$ reliant v_1 à v_n et passant par v_{n-1}^j est égal au poids du chemin optimal de v_1 à v_{n-1}^j additionné au poids de l'arc $v_{n-1}^j \rightarrow v_n$. Puisque le chemin optimal $v_1 \rightarrow v_n$ passe forcément par l'un des sommets v_{n-1}^j, son poids peut être calculé en choisissant le sommet v_{n-1}^j qui minimise cette somme, ce qui permet d'identifier le sommet précédant v_n dans le chemin optimal. Pour calculer le poids du chemin optimal de v_1 à v_{n-1}^j, il suffit d'appliquer le même raisonnement en considérant les sommets V_{n-2}, et ainsi de suite. Ainsi, on définit un algorithme récursif de calcul de chemin optimal.

Un exemple d'application est illustré en Fig. 3.2 page 63.

2/ Formule de récurrence

À chaque sommet d_i^j (Fig. 3.2 page 63) sont associées deux variables : une variable W_i^j destinée à contenir *in fine* le poids du chemin le plus léger d'extrémité initiale G (extrémité gauche du graphe) et d'extrémité terminale d_i^j, et une variable θ_i^j mémorisant le sommet qui précède g_i^j dans ce chemin optimal. La valeur de W_i^j peut être calculée par une formule de récurrence appliquée de gauche à droite dans le graphe pour tout i et j :

$$\boxed{W_i^j = C_i^j + \min_{j'}(W_{i-1}^{j'} + S_{i-1}^{j',j})} \tag{3.3}$$

Cela revient à chercher parmi tous les sommets droits $d_{i-1}^{j'}$ de provenance possible quel est celui qui minimise le poids du plus court chemin passant par la piste j' correspondante. Ce sommet est mémorisé dans θ_i^j. Arrivé a D, une fois la formule appliquée à tout le graphe , on obtient la valeur W_D, soit le poids associé à la meilleure prédiction possible. Celle-ci peut être identifiée par la procédure de retour, en remontant le long de la prédiction en suivant successivement tous les θ_i^j (algorithme 2 page suivante).

3/ L'algorithme détaillé

Algorithme 2 : Algorithme simplifié d'EuGène

Initialisation $(i = 0)$:
$W_0 = 0$;
Récurrence $(i = 1, 2, \cdots, L)$:
Pour $(j = 1, 2, \cdots, J)$:
$W_i^j = C_i^j + \min_{j'}(W_{i-1}^{j'} + S_{i-1}^{j',j})$;
$\theta_i^j = \text{argmin}_{j'} \ (C_i^j + W_{i-1}^{j'} + S_{i-1}^{j',j})$;
Terminaison :
$W(\pi^*) = \min_j W_L^j$;
$\pi_L = \text{argmin}_j(W_L^j)$;
Retour $(i = L, L-1, \cdots, 1)$:
$\pi_{i-1}^* = \theta_i^{\pi_i^*}$;

IV Quelques propriétés intéressantes d'EuGène

1/ Complexité algorithmique

La formule de récurrence est appliquée à tous les sommets d_i^j dont le nombre est linéairement fonction de la taille de la séquence. De plus, pour une application donnée de la formule, le nombre de sommets de destination possibles est borné par le nombre de pistes qui est une constante. Par conséquent, l'algorithme d'EuGène bénéficie d'une complexité linéaire en temps et en espace par rapport à la taille de la séquence.

2/ Performances

EuGène a fait l'objet d'une évaluation comparée de ses performances sur le jeu de données du nom d'AraSet (Pavy *et al.*, 1999b). Sa précision le classe parmi les meilleurs prédicteurs de gènes chez *Arabidopsis thaliana* (Schiex *et al.*, 2001), ce qui en fait une base de travail intéressante.

3/ Comparaison avec les HMM

Le modèle d'EuGène et son algorithme peuvent être comparés à ceux des prédicteurs de gènes basés sur des modèles de Markov cachés (HMM) (voir le chapitre précédent). Pour faire le parallèle entre les deux approches, on peut considérer que les pistes d'EuGène correspondent aux états des HMM, les arcs de type *contenu* aux probabilités d'émission, et les arcs de

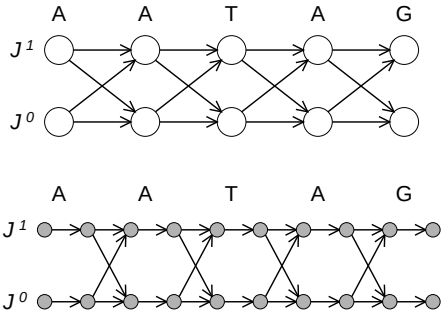

FIG. 3.3 – Comparaison des deux modélisations possibles par un HMM (haut) et par un DAG (bas) sur une courte séquence d'exemple `AATAG`, avec deux annotations J^0 et J^1. Pour le HMM, toutes les flèches représentent les probabilités de transition entre les états (cercles).

type *signal* aux probabilités de transition entre états (Fig. 3.3). Certaines différences sont toutefois notables :

En premier lieu, un HMM est un modèle probabiliste, et de cette caractéristique découlent certaines contraintes de cohérence d'ordre mathématiques. Par exemple, la somme de toutes les probabilités conditionnelles doit être égale à 1. EUGÈNE, lui, fonctionne avec des scores (les poids des arcs) libres de toute contrainte, ce qui le dégage du cadre strict probabiliste sans en perdre le potentiel d'exploitation, puisque des algorithmes semblables s'appliquent dans les deux cas. Il bénéficie ainsi d'une structure souple et appropriée pour envisager des développements. Notons également que dans les HMM, les probabilités de transition sont homogènes et donc les mêmes tout le long de la séquence, alors que EUGÈNE peut gérer des variations selon les besoins.

De plus, l'estimation classique des probabilités des HMM par maximisation de la vraisemblance des données fait l'hypothèse que le modèle est correct, ce qui n'est pas forcément le cas comme nous avons vu. EUGÈNE adopte une approche beaucoup plus pragmatique, et estime ses paramètres par un processus d'optimisation de ses performances globales (que nous verrons dans le chapitre suivant). Ainsi, les paramètres peuvent s'adapter aux imperfections du modèle et en atténuer l'impact sur les performances du logiciel.

Enfin, beaucoup de prédicteurs de gènes basés sur des HMM utilisent un modèle de type GHMM, ce qui implique une prise en compte de distributions de probabilités de longueur associées aux états. EUGÈNE ne prend des longueurs en considération que pour tenir compte de certaines contraintes biologiques et éviter des annotations incohérentes, en exigeant une taille minimum pour certains éléments (les introns par exemple). Ceci correspond à des distributions sur les longueurs qui ne sont pas des distributions de probabilités.

V COMMENT INTÉGRER DE NOUVELLES INFORMATIONS ?

Pour intégrer un nouveau type de données, il faut tout d'abord mettre au point une façon de prendre en compte les informations dans le modèle par

une procédure appropriée de pénalisation de pistes. Puis, l'intensité de cette pondération peut être déterminée en optimisant les paramètres associés à la source d'information. Ce processus d'optimisation est d'une importance majeure, nous allons maintenant le voir en détail.

Un joueur qui court en travers ne peut lire le jeu qu'en diagonale.

— Philippe Guillard

(Petits bruits de couloir, 1999)

Je voudrais simplement que nous ayons une courte pensée pour ceux de mes camarades du spectacle qui n'ont actuellement aucun travail, sous prétexte qu'ils n'ont aucun talent.

— Pierre Desproges

Chapitre 4

Estimation des paramètres par optimisation

Comme nous l'avons vu dans le chapitre précédent, l'intégration d'une source d'information dans le logiciel EuGène nécessite généralement l'estimation de paramètres spécifiques qui servent à pondérer certains arcs : un couple α et β pour chaque logiciel de prédiction de signal (codon start, sites d'épissage), un poids z pour le codon stop et un poids y pour le début et la fin de transcription ; si l'on ajoute les informations de type *similarité*, on a une valeur ε pour les transcrits et un coefficient ρ pour chaque base de données protéique. Ce réglage s'effectue grâce à un processus d'optimisation des performances du logiciel (sensibilité et spécificité) sur un jeu de données d'apprentissage contenant un ensemble de séquences génomiques. Ce chapitre présente le processus d'optimisation, ainsi que le travail réalisé dans le cadre de cette thèse qui a permis son amélioration.

I Motivations : pourquoi optimiser ?

La première question que l'on peut se poser est : pourquoi un travail sur ce processus d'optimisation ? Nous avons vu à l'issue de l'étude de l'état de l'art (chapitre 2) que l'intégration d'informations représentait un point essentiel du domaine de la prédiction de gènes. Or, le processus d'optimisation est capital car toute prise en compte d'une nouvelle information passe par une nouvelle procédure de pondération des arcs du graphe d'EuGène qui dépend généralement de paramètres spécifiques.

Dans le cas général, la modification des poids induite par la prise en compte de la nouvelle source d'information considérée n'est pas simple à déterminer, car elle reflète en quelque sorte la confiance qui lui est attribuée comparativement aux autres sources d'information, qui est difficilement quantifiable *a priori*. Dans certains cas on peut choisir de simplement suppri-

mer des arcs (pondération par une valeur infinie), ou simplement se contenter d'ajouter des arcs sans les pénaliser (ce qui revient à passer d'un poids infini à nul). Mais le simple fait de modifier le graphe d'EuGène altère l'espace des prédictions possibles, et comme la prédiction optimale est définie par rapport à cet espace, il est possible qu'elle soit radicalement modifiée par l'intégration de connaissances nouvelles. Les paramètres de confiance relative associés à une source d'information donnée dépendent de l'ensemble des autres sources d'information. Par conséquent, si cet ensemble est modifié (par exemple en cas d'une intégration nouvelle), l'équilibre global doit être restauré par un nouveau calibrage de tous les paramètres.

Une autre raison soulignant l'importance de l'optimisation concerne l'aspect de l'évolution du logiciel. Si l'on se penche sur la littérature du domaine de la localisation de gènes (voir le chapitre 2) on constate que de nouveaux types d'information sont régulièrement exploités par les logiciels de prédiction (Fig. 2.4 et 2.5 page 36). Or, nous avons vu que les nouvelles intégrations demandent souvent le développement de méthodologies spécifiques, et il est difficile dans un système complexe de garantir une cohésion de l'ensemble des éléments et une continuité dans les performances. Pour un logiciel en bioinformatique, il en va de même que pour une espèce en biologie, la capacité d'adaptation est essentielle à la survie dans un environnement compétitif qui évolue rapidement.

De façon similaire, une capacité d'évolution est également nécessaire pour que le logiciel puisse s'adaptater à la croissance du nombre de génomes séquencés. En effet, comme les génomes d'espèces différentes ne présentent pas les mêmes caractéristiques, un ensemble de valeurs spécifiques de paramètres est nécessaire pour chaque espèce. Vu que l'éventail de génomes séquencés s'élargit avec le temps, un logiciel évolutif est indispensable.

Enfin, l'évolution du logiciel que permet l'optimisation des paramètres représente un intérêt capital concernant les possibilités de recherche. Cette thèse ayant pour but d'expérimenter de nouvelles intégrations d'informations, il est capital de disposer d'une procédure qui ne limite pas les tentatives d'explorations méthodologiques. Chaque nouveau développement méthodologique devrait pouvoir être testé efficacement et dans un délai raisonnable.

Or, il se trouve qu'au début de l'étude, le processus d'optimisation des paramètres d'EuGène comportait de sérieuses limitations au niveau de sa programmation informatique : il était rigide, dans le sens où, en étant fortement dépendant de l'ensemble des sources d'informations prises en compte, il n'offrait pas la possibilité d'être modifié aisément. De plus, il était coûteux en temps de calcul, ce qui limitait aussi son utilisation. Étant peu automatisé, il demandait une part importante d'intervention humaine, pour son déroulement d'une part, et pour l'interprétation de ses résultats d'autre part. Enfin, il n'avait fait l'objet d'aucune étude ou estimation de son efficacité ou de sa fiabilité.

Par conséquent, il était absolument nécessaire de reprendre l'estimation des paramètres avant d'envisager tout développement méthodologique de prise en compte d'information dans le logiciel EuGène. L'objectif de ce tra-

vail sur le processus d'optimisation consiste donc à en améliorer la souplesse, la rapidité, l'automatisation ainsi que la fiabilité.

II Le critère : que veut-on optimiser ?

L'objectif à terme est bien entendu d'amener EuGène aux meilleures performances possibles en terme de précision de ses prédictions. Par conséquent, et contrairement aux méthodes classiques d'estimation de paramètres dans les HMM, c'est sur ce critère de performance que repose le processus d'optimisation. Comme pour toute procédure d'apprentissage supervisé en intelligence artificielle, il est nécessaire de disposer d'un jeu de données d'entraînement et de définir un critère précis à optimiser.

1/ Le jeu de données

Le rôle du jeu de données d'apprentissage est de servir de base à l'entraînement du logiciel. Par rapport à l'ensemble des données existantes, un jeu idéal est un sous-ensemble représentatif, de grande taille, et bien caractérisé. Pour l'identification de structures géniques, le jeu d'apprentissage contient des séquences génomiques contenant des gènes dont les positions des exons sont connues. Les connaissances sur ces gènes sont en général issues de travaux d'experts biologistes, qui ont annoté un gène à partir de la séquence de son ARNm.

Le jeu d'apprentissage choisi est un ensemble de séquences génomiques d'*Arabidopsis thaliana* du nom d'AraClean (Korning *et al.*, 1996). Ce jeu est issu d'un travail de correction d'erreurs d'annotation présentes dans les bases de données publiques, et a fait l'objet d'une attention particulière d'experts en annotation. Dans la version utilisée, il comporte 144 séquences, chacune contenant un gène entier, flanqué en général d'un certain contexte nucléotidique en amont et en aval de la séquence codante. Les coordonnées des exons figurant dans l'annotation expertisée sont prises comme référence.

Sans remettre en cause la qualité de l'annotation réalisée, nous avons toutefois choisi de procéder à une modification notable du jeu d'apprentissage afin qu'il soit plus conforme à une configuration génomique réaliste. En effet, les logiciels de localisation de gènes ont généralement une fâcheuse tendance à occasionner des fusions ou des fissions de gènes dans leurs prédictions, et il est souhaitable qu'EuGène ne prenne pas une telle orientation. Or, dans un jeu d'apprentissage qui ne comporte qu'un gène par séquence (comme c'est le cas ici), le logiciel est entraîné dans une situation atypiquement confortable, où la fusion de gènes est impossible et dans laquelle on ne s'attend donc pas à ce que les paramètres s'adaptent pour éviter ce danger. Afin de placer EuGène dans une configuration génique plus représentative et de le confronter aux risques conjugués de fusion et de fission pendant le processus d'apprentissage, nous avons réalisé une version d'AraClean qui contient 2 gènes par séquence. Chaque séquence de ce nouveau jeu de données, que nous

appellerons AraClean2, est le fruit de la concaténation de 5 éléments, à savoir une séquence d'AraClean, puis une séquence intergénique provenant d'un jeu aimablement fourni par M. Lescot (Université de Ghent, communication personnelle) puis une deuxième séquence d'AraClean, le tout étant flanqué par deux séquences intergéniques supplémentaires de part et d'autre. Ainsi, même si chaque séquence du jeu de données obtenu ne correspond pas vraiment à un fragment génomique "authentique", on s'attend à ce que le jeu AraClean2 reflète une situation convenablement réaliste.

2/ Evaluation des performances

Afin de permettre au processus d'optimisation d'aboutir à des modifications de paramètres produisant une augmentation des performances d'EuGène, il faut définir un critère à optimiser qui soit fonction de la qualité des prédictions du logiciel sur le jeu de données construit.

Classiquement, les performances d'un prédicteur de gènes (et plus généralement de tout outil de détection) s'évaluent par deux valeurs essentielles : la sensibilité et la spécificité (Burset et Guigó, 1996). La sensibilité correspond au pourcentage d'éléments à détecter qui le sont correctement, et la spécificité correspond au pourcentage d'éléments prédits qui sont corrects (voir la page 54 du chapitre 2). On caractérise en général un niveau nucléotide, un niveau exon, et un niveau gène, sachant qu'un exon ou un gène n'est considéré comme correct que si la prédiction est exactement identique à la référence (au nucléotide près).

Afin de permettre une mesure de la qualité des prédictions sur le jeu d'apprentissage, nous avons défini un critère Φ. Il s'agit d'une combinaison linéaire des valeurs de sensibilité et de spécificité au niveau gène (respectivement notées $\mathcal{S}_n^g$ et $\mathcal{S}_p^g$) et au niveau exon ($\mathcal{S}_n^e$ et $\mathcal{S}_p^e$). Nous appellerons *"fitness"* la valeur de ce critère, qui est égale à

$$\Phi = \phi_n^g \cdot \mathcal{S}_n^g + \phi_p^g \cdot \mathcal{S}_p^g + \phi_n^e \cdot \mathcal{S}_n^e + \phi_p^e \cdot \mathcal{S}_p^e \tag{4.1}$$

avec $\phi_n^g + \phi_p^g + \phi_n^e + \phi_p^e = 1$. Le niveau nucléotide, jugé peu informatif, n'est pas considéré en raison de son faible potentiel discriminant. Différentes combinaisons des valeurs de ϕ sont possibles, la plus simple étant $\phi_n^g = \phi_p^g = \phi_n^e = \phi_p^e = \frac{1}{4}$, mais en pratique, puisque les séquences contiennent peu d'éléments intergéniques (mettant à l'épreuve la spécificité) l'accent est mis sur la sensibilité avec $\phi_n^g = 0.45$, $\phi_p^g = 0.1$, $\phi_n^e = 0.35$, et $\phi_p^e = 0.1$.

Pour résumer, la qualité attribuée à un ensemble de paramètres donné est représentée par sa *"fitness"*, déterminée par l'analyse du résultat d'une exécution d'EuGène sur tout le jeu de données AraClean2. Il faut garder à l'esprit que cette *"fitness"* ne représente pas uniquement la qualité du logiciel en soi, car elle est conditionnée par un certain ensemble de sources d'information, des paramètres qui leur sont associés, et du jeu de données sur lequel elle est mesurée.

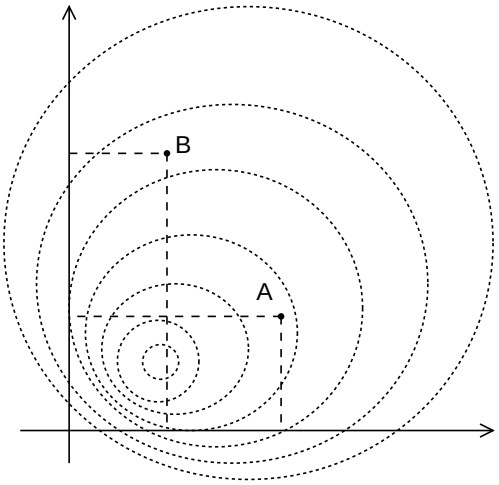

FIG. 4.1 – Exemple de problème d'optimisation avec un espace de recherche à deux dimensions. Les axes représentent les variables à estimer et les courbes de niveau en pointillés illustrent les valeurs du critère d'intérêt (*"fitness"*), l'optimum étant le cercle le plus petit. À chaque couple de variables correspond une *"fitness"*, l'objectif étant de trouver le jeu de variables qui optimise le critère. Ici, le point A bénéficie d'une meilleure *"fitness"* que le point B. Cet exemple sera repris pour illustrer les différentes méthodes.

III LES MÉTHODES : COMMENT OPTIMISER ?

Pour procéder aux modifications des paramètres, deux méthodes distinctes d'optimisation sont utilisées. Cette section présente les motivations et les principes théoriques sous-jacents.

1/ L'algorithme génétique

a. Pourquoi ce type de méthode ?

La principale raison du choix d'un algorithme génétique est que cette méthode était déjà testée dans une version prototype, et semblait donner de bons résultats (Schiex *et al.*, 2001). De plus, il faut prendre en considération le fait que la relation entre un ensemble de paramètres et la valeur de Φ (la *"fitness"*) n'est pas une fonction simple, dans le sens où elle n'est ni dérivable, ni continue, ni même exprimable par une formulation mathématique. En terme mathématiques, on se trouve là devant un problème mal posé, pour lequel il peut y avoir plusieurs solutions (configuration de paramètres donnant des performances optimales), voire une infinité (c'est le cas ici, les paramètres étant des variables continues). De plus, comme le nombre de paramètres dépasse la dizaine (2 pour NETSTART, 4 pour NETGENE2—2 donneurs et 2 accepteurs—, 4 pour SPLICEPREDICTOR, 1 pour le début et fin de transcription, 1 pour la fin de traduction, sans tenir compte des similarités), et qu'il s'agit de nombres réels, l'espace de recherche est gigantesque. Il se trouve en outre que le profil de cet espace de recherche est particulièrement "accidenté" (nous en aurons la confirmation plus loin), c'est-à-dire qu'il comporte une multitude d'optima locaux. Enfin, comme nous verrons également plus loin, le temps de calcul exigé par l'évaluation du critère est considérable et exclut les méthodes demandant un nombre important de simulations. Bref, puisque les algorithmes génétiques ne font aucune hypothèse sur la fonction à optimiser et qu'ils sont réputés efficaces pour les espaces de

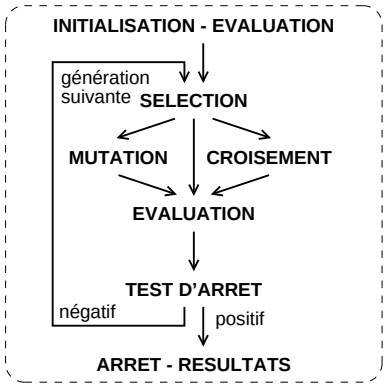

FIG. 4.2 – Schéma du fonctionnement général d'un algorithme génétique. Les différentes étapes du processus sont présentées dans le texte.

recherches complexes, cette approche est tout à fait appropriée.

b. Quel en est le fonctionnement ?

Les algorithmes génétiques doivent leur nom (ainsi que leur terminologie) à la théorie de Charles Darwin sur l'évolution des espèces (mentionnée à l'origine sous le terme de *"descent with modification"*) qui introduit des notions de variations, de compétition pour la vie et de sélection naturelle (Darwin, 1859). L'objectif est de maximiser un critère qui est le résultat d'une fonction d'évaluation d'un ensemble de variables qui définissent un espace de recherche. La terminologie emploie le terme de *"fitness"* pour le critère, de chromosome ou d'individu pour un ensemble de valeurs donné des variables, de population pour un ensemble d'individus, et de génération pour une étape du processus[1]. Initialement destinée à des chaînes de valeurs binaires (0/1), l'application des algorithmes génétiques s'est en pratique rapidement étendue à des chromosomes plus complexes, comme c'est ici le cas. Comme ouvrage de référence sur les algorithmes génétiques, on peut citer Holland (1975) et Goldberg (1989).

On travaille donc sur une population de N individus (chacun étant caractérisé par un chromosome qui contient une valeur par variable d'intérêt) qui évolue au cours des générations g. À chaque individu n_i^g avec $i \in N$ correspond une *"fitness"* $\Phi(n_i^g)$. L'objectif est d'obtenir au terme de l'optimisation un individu avec une valeur de *"fitness"* la plus élevée possible. Le déroulement du processus est schématisé dans la Fig. 4.2, les notations utilisées sont présentées dans le Tab. 4.1 ci-contre, et quelques-unes des principales étapes sont détaillées ci-dessous.

i) initialisation

Au départ, le processus est initialisé par la création d'une population de N individus dont les valeurs des variables sont attribuées soit aléatoirement soit arbitrairement, selon les informations dont on dispose sur certaines régions de l'espace de recherche). Les individus, après une évaluation initiale

[1] Si la comparaison entre le phénomène de l'évolution des espèces et ce genre de modélisation mathématique peut paraître considérablement réductrice, voire d'un goût douteux, elle n'en reste pas moins amusante et indéniablement efficace.

"fitness" ..	critère à optimiser
$\boldsymbol{N}$	taille de la population
$\boldsymbol{g}$	génération, itération du processus
$\boldsymbol{G}$	nombre maximum de générations
$\boldsymbol{n_i^g}$	individu n° i (avec $i \in N$) de la génération g
$\boldsymbol{\Phi(n_i^g)}$	*"fitness"* de l'individu n_i^g
$\boldsymbol{\mathcal{V}_j(n_i^g)}$	variable n° j du chromosome de l'individu n_i^g
$\boldsymbol{P(m)}$	proportion d'individus mutés
$\boldsymbol{P(c)}$	proportion d'individus croisés
$\boldsymbol{\eta}$	nombre d'individus conservés par l'élitisme
$\boldsymbol{\gamma}$	bruit gaussien paramétrant la mutation
$\boldsymbol{\xi}$	valeur aléatoire paramétrant le croisement, avec $-0.5 \leq \xi \leq +1.5$
$\boldsymbol{\Phi^g_{max}}$	*"fitness"* maximum de la génération g
$\boldsymbol{\Delta}$	taux d'augmentation minimum de la meilleure *"fitness"* exigé entre deux générations

TAB. 4.1 – Notations utilisées pour les algorithmes génétiques

de leur *"fitness"*, entrent dans le cycle des générations.

ii) évaluation

La fonction d'évaluation est primordiale dans un algorithme génétique, car c'est elle que l'on cherche à optimiser. La seule contrainte qui s'exerce à son égard est d'ordre pratique ; au vu de la fréquence attendue de son exécution, il est préférable que cette évaluation ne soit pas exigeante en temps de calcul. Elle est exécutée sur chaque individu de la population, afin de lui décerner sa *"fitness"*.

iii) sélection

Au début de chaque génération, la population est soumise à une sélection qui va permettre aux individus les plus adaptés de se reproduire. Un individu a d'autant plus de chances de transmettre son chromosome (ses valeurs) à la génération suivante que sa *"fitness"* est élevée. Par exemple, on peut définir la probabilité qu'un individu n_i^g se reproduise par l'expression $\frac{\Phi(n_i^g)}{\sum_{j=1}^{N} \Phi(n_j^g)}$.

Pour garantir que les meilleurs individus soient préservés d'une génération à la suivante, on peut recourir à une variante optionnelle appelée élitisme : cela consiste avant d'appliquer la sélection à prendre le ou les η (ce nombre étant le paramètre de l'élitisme) meilleurs individus sur la base de leur *"fitness"*, et de les projeter directement sans modification dans la population de génération supérieure.

Remarque 13 *L'application de l'élitisme garantit que la qualité de la solu-*

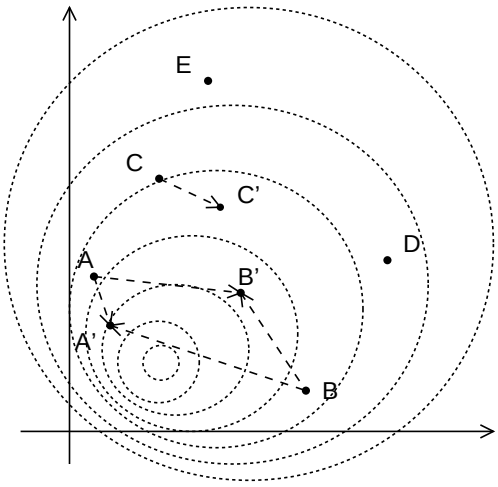

FIG. 4.3 – Illustration de l'algorithme génétique sur l'exemple précédent. La population initiale est composée de 5 individus A,B,C, D et E dont les chromosomes (les coordonnées des points) ont été tirés aléatoirement. Les flèches pointillées représentent le résultat de 2 opérateurs : croisement entre A et B (qui donnent respectivement A' et B') et mutation pour C (qui donne C'). Les points bénéficiant de la meilleure *"fitness"* ont plus de chance de faire partie de la génération suivante, favorisant un regroupement progressif de la population autour de l'optimum.

tion proposée ne diminue pas d'une itération à l'autre.

Le processus suivant la sélection est la reproduction, qui fait intervenir les opérateurs de mutation et de croisement.

iv) mutation

Cet opérateur agit en "bruitant" les variables d'un individu, ce qui revient à les modifier aléatoirement dans leur domaine de valeur. On peut choisir pour muter une variable $\mathcal{V}_j(n_i^g)$ en $\mathcal{V}_j(n_i^{g+1})$ une formule comme

$$\mathcal{V}_j(n_i^{g+1}) = \mathcal{V}_j(n_i^g) + \gamma$$

où γ est un bruit gaussien (ajusté sur le domaine de valeur de $\mathcal{V}_j$). Le pourcentage d'individus subissant la mutation est fixé par un paramètre de l'algorithme génétique noté $P(m)$.

v) croisement

L'opération de croisement consiste à remplacer deux individus de la génération courante par deux individus fils, dont les chromosomes sont obtenus par l'exécution d'une opération sur chaque paire des variables parentes. Par exemple, une variable fille $\mathcal{V}_j(n_i^{g+1})$ peut être calculée à partir de deux variables parentes $\mathcal{V}_j(n_j^g)$ et $\mathcal{V}_j(n_k^g)$ par la formule

$$\mathcal{V}_j(n_i^{g+1}) = \xi \cdot \mathcal{V}_j(n_j^g) + (1 - \xi) \cdot \mathcal{V}_j(n_k^g)$$

avec ξ choisi uniformément dans $[-0.5\ ;\ +1.5]$, ce qui permet d'explorer l'espace entre les variables mais aussi au-delà. Le croisement concerne une fraction de la population déterminée initialement par le paramètre $P(c)$ de l'algorithme génétique.

vi) arrêt du processus

Généralement, deux conditions d'arrêt sont exploitables. La première est déterminée par la valeur G, définie avant le lancement de l'optimisation, qui

borne simplement le nombre maximum de générations possibles. Le processus s'arrête donc dès que $(g \geq G)$. On peut également autoriser un arrêt en fonction de l'évolution du critère, en évitant d'attendre systématiquement le nombre maximum de générations. Si on note Φ^g_{max} la valeur maximum de *"fitness"* portée par un individu de la génération g, on peut fixer la fin du processus dès que $(\Phi^g_{max} - \Phi^{g-1}_{max} < \Delta \cdot \Phi^{g-1}_{max})$, où Δ représente le taux d'augmentation minimum que l'on souhaite. Au terme de l'algorithme, les variables du chromosome que porte l'individu présentant la meilleure *"fitness"* correspondent au résultat de l'optimisation.

c. Quelques propriétés

Les algorithmes génétiques présentent une forte composante stochastique[2]. Bien que rien ne garantisse qu'au terme du processus le résultat proposé corresponde à l'optimum de la fonction considérée, des travaux théoriques ont cependant permis de démontrer une convergence asymptotique des algorithmes génétiques (Cerf, 1994). De plus, certains réglages, comme la taille de la population en nombre d'individus ou le fait de lancer plusieurs exécutions (*"run"*) du processus avec des populations initiales différentes[3] permet en pratique une exploration satisfaisante de l'espace global de recherche. L'inconvénient subsistant est que cette exploration reste plutôt "grossière", dans le sens où si l'espace global de recherche semble bien parcouru, les solutions proposées gagneraient à être affinées localement (localement signifiant que l'on considère l'entourage proche de l'espace de recherche, défini par des valeurs voisines de variables). C'est pour cela qu'il n'est pas surprenant de compléter l'utilisation d'un algorithme génétique par un algorithme d'optimisation locale, comme présenté ci-dessous.

2/ Optimisation par exploration locale

a. Motivations

L'objectif de cette étape est de parfaire la solution proposée par l'algorithme génétique (ou les solutions proposées, dans le cas où plusieurs exécutions sont lancées) en réalisant une exploration locale de l'espace de recherche autours des valeurs proposées. Cet affinage permet en pratique d'augmenter rapidement (comparé à l'utilisation de l'algorithme génétique seul) la qualité d'un jeu de paramètres donné. Nous avons expérimenté deux méthodes d'optimisation locale : un algorithme du simplexe, et un algorithme de recherche linéaire.

b. Le simplexe

La méthode du simplexe de Nelder-Mead (Nelder et Mead, 1965) est un algorithme itératif utilisable pour l'optimisation de fonctions non linéaires.

[2] stochastique se dit d'un phénomène qui relève partiellement du hasard.

[3] en modifiant la graine du générateur de nombres pseudo-aléatoires avant chaque *"run"*.

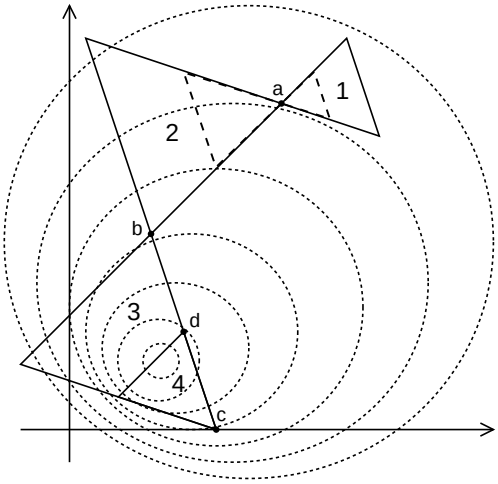

FIG. 4.4 – Illustration de l'algorithme du simplexe. Pour chaque simplexe (triangle) un chiffre indique l'itération correspondante et une lettre le coin à la meilleure *"fitness"*. Le simplexe 1 est l'initial. Par rapport à son meilleur coin (a), il est étendu en 2, qui est par rapport à son meilleur coin (b) réfléchi en 3, qui est contracté en 4 par rapport à c. Les deux simplexes non retenus à chaque itération ne sont pas affichés, excepté pour la première (en pointillés, le contracté et le réfléchi du simplexe 1).

Le principe est de construire un simplexe, c'est-à-dire pour un espace de recherche de dimension n un volume convexe de même dimension, défini par $(n+1)$ "coins" indépendants (de simples points de l'espace de recherche) et délimité par des hyperplans de dimension $(n-1)$. Par exemple, il s'agit d'un triangle pour un espace de recherche à deux dimensions ou d'un tétraèdre pour $n = 3$. L'objectif est de contenir l'optimum de la fonction à l'intérieur de ce simplexe, dont la taille est modifiée à chaque itération (principalement réduite) jusqu'à ce qu'il soit suffisamment réduit pour enfermer la solution avec une précision satisfaisante. À chaque itération, l'application de trois opérateurs sur l'ancien simplexe (contraction, réflexion et expansion) en fonction du coin proposant la meilleure *"fitness"* permet la construction de trois nouveaux simplexes, dont les coins sont produits par combinaison linéaire des anciens. Le simplexe contenant le coin qui donne la meilleure valeur du critère d'intérêt est conservé pour l'itération suivante.

Plus précisément, un simplexe V de $\mathcal{R}^n$ est un ensemble $(v_0, v_1, \cdots, v_n)$ de points de $\mathcal{R}$, tel que v_0 propose la meilleure *"fitness"*. À chaque itération, on peut créer et tester trois simplexes $U = (u_0, u_1, \cdots, u_n)$ tels que $\forall i \in [1..n]$, $u_i = v_0 + \lambda(v_i - v_0)$, avec des valeurs de λ en général de $1/2$, -1 et -2 pour donner un simplexe respectivement contracté, réfléchi et étendu.

La figure 4.4 illustre un exemple de déroulement de l'algorithme du simplexe.

En pratique, il s'est avéré que l'application de cette méthode n'a pas donné de résultats satisfaisants pour ce problème. Une raison probable est que les opérations réalisées sur un coin donné du simplexe (et constitué ici par un ensemble de valeurs de paramètres) appliquent la même combinaison linéaire d'un bloc à tous les paramètres, alors que le besoin en affinage au sortir de l'algorithme génétique est très hétérogène selon le paramètre considéré. Nous n'avons donc pas poursuivi cette voie, et opté pour une autre méthode, de recherche linéaire.

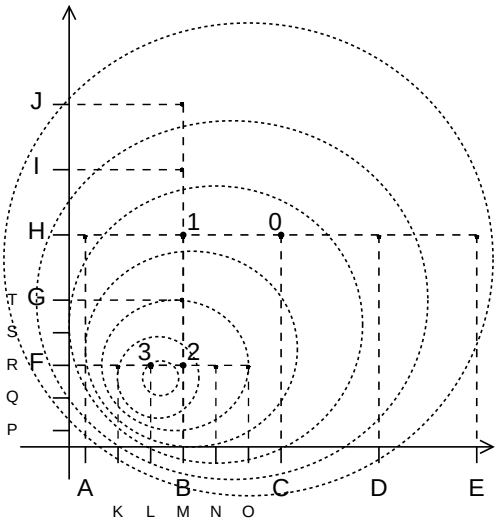

FIG. 4.5 – Exemple de recherche linéaire. Les traits pointillés indiquent les valeurs testées, et les chiffres les points optimaux de chaque échantillonnage. Le point 0 (C,H) est l'initial. Sa variable d'abscisse est échantillonnée dans l'intervalle de A à E autour de C, son ordonnée étant fixée. La *"fitness"* du point 1 étant la meilleure, B devient la nouvelle abscisse optimale. Après échantillonnage de l'ordonnée de F à J, le couple (B,F) est retenu. Pour l'itération suivante, les intervalles initiaux sont réduits et recentrés autour des valeurs optimales (de K à O en abscisse et de P à T en ordonnée).

c. La recherche linéaire (*"line search"*)

Cette méthode, inspirée d'algorithmes d'optimisation classiques de type Newton, repose sur un fonctionnement itératif simple. Comme précédemment, on cherche à optimiser une fonction dépendant d'un ensemble de variables (en l'occurrence les performance d'EUGÈNE en fonction d'un ensemble de paramètres), variables qui définissent un espace de recherche. Ici, on part d'une solution issue de l'algorithme génétique, et dont les variables initiales sont données.

La méthode consiste à procéder de façon itérative à l'exploration systématique par échantillonnage de valeurs dans des parties à dimensions réduites de l'espace de recherche (typiquement 1 ou 2 dimensions). À chaque itération du processus, pour chaque variable (ou combinaison d'un nombre restreint de variables en cas de dimensions multiples) on teste par mesures successives de *"fitness"* un certain nombre de valeurs (ou de combinaisons de valeurs) choisies dans un intervalle réduit autour de la valeur initiale. Ces valeurs échantillonnées sont déterminées par la division de l'intervalle en un certain nombre (ce nombre de division étant un paramètre du processus). La variable est alors fixée à la valeur qui permet d'atteindre la meilleure *"fitness"*, et l'on passe à la variable suivante. Après avoir échantillonné toutes les variables, on réduit les intervalles autour des nouvelles valeurs en diminuant la taille des intervalles précédents d'un certain facteur (également un paramètre important du processus) et on passe à l'itération suivante. La figure 4.5 illustre le déroulement de l'algorithme sur notre exemple simple.

Comme pour l'algorithme génétique, deux conditions d'arrêt sont utilisables : la stabilisation de la croissance de la *"fitness"* maximum, et l'atteinte du nombre maximum d'itérations.

IV Mise en application

1/ Programmation

L'évaluation Le programme chargé de l'évaluation des performances (voir en page 76) a été développé en *Perl*, langage puissant et rapide pour l'analyse de fichiers de textes (format des prédictions et des annotations). La sortie contient les valeurs de sensibilité et spécificité au niveau gène et exon, en considérant la possibilité d'avoir plusieurs gènes (prédits et/ou réels) par séquence. Si de nombreux utilitaires pour accomplir ce genre de tâche sont aujourd'hui disponibles (Keibler et Brent, 2003), il n'en était pas de même il y a trois ans, c'est pourquoi il était nécessaire d'en développer un.

L'algorithme génétique Concernant l'algorithme génétique, un programme existait déjà, mais étant très rudimentaire il ne permettait pas un grand pannel de possibilités. Nous avons travaillé à partir d'un *"package"* conçu pour permettre l'adaptation générique d'un algorithme génétique à tout problème d'optimisation (Durand et Alliot, 1999). Le développement logiciel a consisté à l'installer et à l'adapter, en programmant (en langage *C*) les parties spécifiques du processus d'estimation des paramètres d'EuGène. Le programme *Perl* d'évaluation des performances est régulièrement appelé à travers une commande système.

La recherche linéaire L'ancienne version de la procédure de recherche linéaire nécessitait l'utilisation d'un script spécifique à une utilisation donnée (valeur des intervalles, nombre de divisions) et demandait en outre un investissement important en assistance de l'utilisateur. L'interprétation des résultats n'était pas non plus évidente. Un nouveau programme a donc été développé en langage *C*, qui appelle aussi le programme d'évaluation.

2/ Réglages et exécution

Nous avons vu dans la description des méthodes que le processus d'optimisation dépend lui aussi de nombreux paramètres. Bien que disposant généralement de valeurs par défaut, ils sont fortement dépendants de la nature du problème à optimiser. L'estimation des paramètres de l'optimisation a été réalisée de manière empirique au cours des tests de développement et d'exécution[4].

Pour donner un ordre d'idée des valeurs utilisées et du temps d'exécution, on peut considérer pour l'algorithme génétique une taille de population de 30 à 50 individus et un nombre maximum de générations du même ordre de grandeur. Pour la recherche linéaire, le choix se porte en général sur un nombre de divisions d'intervalles de 5 à 10, à un coefficient de réduction des

[4] bien sûr, il était possible de confier l'estimation des paramètres même de l'optimisation à un autre processus d'optimisation, mais cela n'aurait probablement pas contribué à une simplification de la procédure.

Exécution	1	2	3	4	5	6	7	8	9
"fitness"	89.0	89.0	89.1	89.4	89.7	89.7	90.1	90.1	90.5
S_n^e	96	96.2	96.1	96.5	96.6	96.6	96.6	96.7	96.9
S_p^e	96.8	96.7	96.1	96.6	96.6	96.6	97	96.6	96.7
S_n^g	83.3	84	84	84	84.7	84.7	85.4	85.4	86.1
S_p^g	81.6	79.1	80.7	81.8	81.3	81.3	82	82	81.6

TAB. 4.2 – Résultats de 9 exécutions indépendantes de l'algorithme génétique combiné à la recherche linéaire. Sont présentées les valeurs de sensibilité et spécificité au niveau gène et exon.

intervalles de 0.6, et à un nombre maximum d'itérations de 5.

Ainsi réglé, le processus d'optimisation demande entre 3 jours et une semaine de calculs ininterrompus pour aboutir à une configuration de paramètres satisfaisante (sur un processeur AMD Athlon 1,7 GHz). Cette durée est dûe au fait qu'une seule évaluation d'un ensemble de paramètres donné demande l'exécution d'EUGÈNE sur la totalité du jeu d'apprentissage (plus d'une centaine de gènes).

Une nouvelle version d'EUGÈNE dédiée à *Arabidopsis thaliana* a pu ainsi être créée, en utilisant comme jeu de données d'apprentissage l'ensemble des séquences expertisées des jeux d'ARACLEAN2 et ARASET (Pavy *et al.*, 1999b)[5]. C'est cette version qui est proposée à ce jour à la distribution sur le site Internet[6] d'EUGÈNE, et qui a été utilisée pour la réannotation du génome d'*A. thaliana* dans le cadre du projet CATMA (Crowe *et al.*, 2003).

3/ Robustesse

Le processus d'optimisation étant un élément clef de l'intégration d'informations, il est absolument nécessaire de pouvoir s'y fier pour envisager la poursuite des développements. Afin de savoir si la valeur de *"fitness"* obtenue est assez reproductible malgré le caractère stochastique, nous avons réalisé une série de plusieurs exécutions indépendantes (9) dont les résultats sont présentés dans le tableau 4.2. Étant donné que la *"fitness"* correspondant à un jeu de paramètres aléatoires (non optimisés) se situe aux alentours de 60 à 80 (valeurs relevées sur les individus de la première génération des 9 exécutions indépendantes) et que l'on obtient après optimisation des valeurs proches de 90, on peut conclure que le processus d'optimisation est efficace et robuste.

Remarque 14 *En revanche, il faut noter que le terme "reproductible" ne peut être employé ici, car si les différentes valeurs de* "fitness" *obtenues sont similaires, il n'en est pas de même pour les valeurs des paramètres qui varient selon les exécutions. Bien qu'il se dégage pour certains un intervalle de*

[5] l'ancienne version n'avait bénéficié que du jeu ARACLEAN.

[6] http ://www.inra.fr/bia/T/EuGene/

valeurs fréquemment observées, ce n'est pas le cas général, ce qui témoigne de la complexité de l'espace de recherche. De même, des corrélations entre paramètres sont très difficiles à distinguer.

V Conclusion

Le travail accompli a permis d'améliorer le processus d'estimation des paramètres du logiciel EuGène, afin de le rendre évolutif par rapport aux productions de nouvelles données issues de la génomique. Cette étape était indispensable pour permettre des développements ultérieurs.

1/ Une optimisation pragmatique

Si l'on compare la procédure d'optimisation obtenue à ce qui se fait dans le domaine de la localisation de gènes, c'est tout d'abord son originalité qui apparaît. Généralement, les prédicteurs basés sur des HMM (utilisant des modèles de Markov cachés, présentés en page 44) estiment habituellement leurs paramètres par maximisation de la vraisemblance des données du jeu d'apprentissage. L'inconvénient majeur de la méthode probabiliste est qu'elle fait l'hypothèse que le modèle théorique sous-jacent est correct, ce qui n'est pas forcément le cas. Un exemple intéressant du décalage entre la probabilité d'une prédiction et son exactitude est signalé par (Howe *et al.*, 2002), qui ont exploré une procédure d'estimation des paramètres du logiciel Gaze en appliquant une méthode de descente de gradient (Stormo et Haussler, 1994). Le résultat obtenu est que le jeu de paramètres qui maximise la probabilité de la structure génique correcte ne produit pas nécessairement les meilleures prédictions. Les auteurs concluent que la probabilité de la prédiction correcte selon le modèle semble ne pas être la fonction objective la plus adéquate (Howe *et al.*, 2002), ce qui conforte l'idée d'une approche guidée par la précision des résultats.

La méthode adoptée par EuGène a cela d'intéressant qu'elle est totalement pragmatique, et n'est guidée que par les performances concrètes du logiciel *a posteriori*, ce qui permet aux paramètres de s'adapter aux éventuelles imperfections inhérentes au caractère arbitraire du modèle, du choix des sources d'informations et de la façon de les prendre en compte dans le graphe .

Les méthodes comparables ne sont pas nombreuses dans le domaine. Du côté probabiliste, HMMgene (Krogh, 1997) estime ses paramètres en maximisant la vraisemblance de son modèle HMM, au lieu de maximiser la vraisemblance des données observées. Comme autre approche, le logiciel Dagger (Chuang et Roth, 2001) estime ses coefficients de pondération en tentant de maximiser un critère de performance (moyenne de sensibilité et spécificité au niveau exon). La méthode d'optimisation utilisée est celle du simplexe que nous venons de décrire. Cependant, le potentiel d'intégration du logiciel ne paraît pas exploité (voire démontré) car il ne permet qu'une

approche *ab initio* (intrinsèque).

2/ Perspectives

Le processus d'optimisation pourrait faire l'objet d'études supplémentaires, testant l'influence par exemple des paramètres de l'algorithme génétique, ou des jeux de données d'apprentissage. Le principal défaut de la procédure reste son besoin en temps de calcul, plus important qu'une estimation classique de paramètres dans les HMM. Toutefois, au vu de l'efficacité satisfaisante du processus obtenu, il n'était pas primordial de consacrer d'avantage de temps pour son perfectionnement. Depuis, grâce aux efforts de l'équipe EuGène, le processus a été complètement intégré au logiciel, permettant ainsi une automatisation totale. Les méthodes sous-jacentes n'ont pratiquement pas été modifiées et fonctionnent dans la version actuelle d'EuGène.

Les perspectives apportées par ces développements concernent essentiellement deux aspects d'évolution du logiciel :

– l'extension du spectre d'action d'EuGèneà de nouvelles espèces.

– l'extension d'EuGèneà de nouvelles intégrations d'information.

Puisque pour le génome d'une espèce donnée (avec son jeu de données d'apprentissage associé) un ensemble spécifique de paramètres est généralement nécessaire et que celui-ci est produit par l'optimisation, l'adaptation d'EuGèneà de nouvelles espèces est désormais rendue plus facile. Les versions d'EuGène développées depuis dans le cadre de projets d'annotation des génomes d'*Oryza sativa* et de *Medicago troncatula* représentent une concrétisation de ce nouveau potentiel d'adaptation.

Le deuxième aspect concerne l'accessibilité au développement de nouvelles intégrations d'information pour aider la localisation des gènes. C'est ce potentiel qui nous intéresse particulièrement dans le cadre de cette thèse, et dont nous présentons une mise en application dans le chapitre suivant.

Le monde couraille à gauche, à dret, comme un troupeau qui court dans' bouette.
Y'en a qui calent, y'en a qui arrêtent, pis tout' les autres leu' pilent s'a tête!

— Michel "Plume" Latraverse
(Mouton noir)

Fucius a dit : "Une civilisation sans la Science,
c'est aussi absurde qu'un poisson sans bicyclette."

— Pierre Desproges

Chapitre 5

Intégration d'homologies inter et intra-génomiques

Maintenant que nous avons vu comment EuGène a acquis la capacité d'intégrer de nouvelles informations, nous allons nous intéresser à une mise en application de ce potentiel sur l'exploitation de données d'homologies entre séquences nucléiques.

I Contexte biologique

1/ L'homologie : *"qu'es aquo ?*[1] *"*

Certaines notions spécifiques de biologie sont requises avant d'aborder cette partie ; voici donc quelques rappels simplifiés de nomenclature.

a. Similarité

La similarité entre séquences biologiques est une mesure objective de leur ressemblance. Elle peut être quantifiée par un taux de similarité, ou taux d'identité, qui correspond à la proportion d'éléments identiques en vis-à-vis dans un alignement optimal entre ces séquences. Par exemple, le taux d'identité entre `GATTACA` et `GAGTAA` est de 5/7 (alignement : $\begin{smallmatrix}\texttt{GATTACA}\\ \texttt{GAGTA-A}\end{smallmatrix}$).

b. Homologie

Définition 10 (Homologie)
Deux ou plusieurs séquences sont homologues si et seulement si elles dérivent d'une séquence ancestrale commune. Par extension, on dit de gènes qu'ils sont homologues s'ils dérivent d'un même gène ancestral.

[1] "qu'est-ce que c'est ?" (languedocien)

Remarque 15 *L'homologie se manifeste généralement par une similarité significative entre les séquences concernées si la durée de divergence évolutive est suffisamment courte. En général, on admet qu'à partir d'un certain taux de similarité, dont la valeur dépend notamment de la longueur et de la nature (nucléique ou protéique) des séquences, ces dernières sont homologues.*

On distingue plusieurs catégories de gènes homologues. Voici la description des deux principales :

i) Orthologie

L'orthologie est une homologie issue d'un phénomène de spéciation[2].

L'explication biologique de ce phénomène est simple : après une spéciation, si un gène provenant de l'espèce ancestrale code pour une protéine dont la fonction biologique est nécessaire aux deux espèces "filles", chacune ayant hérité originalement d'une copie identique, cette dernière est conservée. En effet, la pression de sélection élimine les individus porteurs de mutations délétères pour la fonction biologique d'intérêt, les gènes orthologues sont donc limités dans leur évolution par des contraintes fonctionnelles, ce qui explique la ressemblance des séquences codantes correspondantes. Ce phénomène est largement répandu dans l'arbre du vivant, de nombreuses fonctions biologiques étant requises par plusieurs espèces. La détection d'orthologies s'effectue par comparaison de séquences génomiques d'espèces différentes, méthode qui porte le nom générique de génomique comparative.

On dit donc de deux gènes qu'ils sont orthologues s'ils sont homologues et s'ils codent pour des protéines assurant la même fonction dans deux espèces différentes (homologie inter-génomique).

ii) Paralogie

La paralogie est une homologie issue d'un phénomène de duplication.

Une duplication est un certaine forme de mutation impliquant une région du génome qui peut être de taille considérable (on parle alors de réarrangement chromosomique), voire même l'intégralité du génome. La région concernée est alors copiée puis insérée ailleurs dans le génome (à la façon d'un "copier-coller"), parfois à la suite de l'originale (cas de duplication en tandem).

Considérons un gène qui subit une telle duplication dans le génome d'une espèce donnée. Puisque la fonction biologique concernée est assurée normalement par l'une des deux copies qui produit la protéine d'intérêt, l'autre se retrouve libérée de toute pression de sélection et peut alors accumuler des mutations dans sa séquence nucléique, même si celles-ci affectent les régions codantes. Dans ce cas, il est possible que les modifications de la séquence protéique qui en découlent provoquent l'apparition d'une nouvelle fonction bénéfique pour l'organisme, qui peut se retrouver conservée dans le génome de l'espèce considérée. Ce phénomène est également très répandu dans les

[2]spéciation : apparition de différences génétiques, morphologiques, physiologiques ou éthologiques entre deux populations d'une même espèce, entraînant leur séparation en deux espèces distinctes.

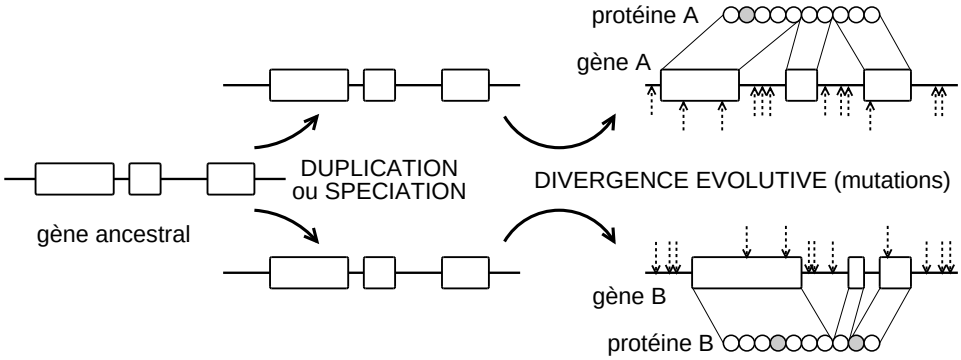

FIG. 5.1 – Exemple schématique illustrant l'apparition de gène homologues. À partir d'un gène ancestral (à gauche) subissant un phénomène de spéciation (orthologie) ou de duplication (paralogie), deux copies sont générées (gène A en haut et gène B en bas). Le trait horizontal représente les éléments non-codants de la séquence génomique et les rectangles les exons. Après un certain temps de divergence évolutive des mutations se fixent, préférentiellement dans les régions non-codantes (ce qui tend à faire diverger les séquences génomiques), et plus sporadiquement dans les régions codantes. Quelques mutations sont représentées sur ce schéma (flèches pointillées verticales). Celles qui surviennent dans les exons ne provoquent pas forcément un changement de la séquence protéique en raison de la redondance du code génétique (surtout si la base concernée est la troisième d'un codon). Les séquences protéiques sont représentées par les enchaînements de cercles (pour les acides aminés), grisés lorsque l'acide aminé correspondant a muté. Selon s'ils sont issus d'un phénomène de spéciation ou de duplication, les gènes A et B sont respectivement orthologues ou paralogues.

génomes, certaines duplications répétées donnant le jour à des "familles" de gènes issus de paralogie.

On dit donc de deux gènes qu'ils sont paralogues s'ils sont homologues et qu'ils appartiennent à la même espèce (homologie intra-génomique).

2/ L'homologie : quel intérêt ?

Le potentiel de l'analyse comparative de séquences génomiques pour aider la localisation de gènes est maintenant largement reconnu (Mathé *et al.*, 2002; Brent et Guigó, 2004). La raison essentielle qui fait de l'homologie une source d'information précieuse est la différence globale de conservation de séquence entre les régions codantes et les non-codantes.

En effet, le point commun des deux définitions précédentes (orthologie ou paralogie) réside dans la notion d'utilité fonctionnelle protéique. Comme dit plus haut, la pression de sélection ne s'exerce pas de la même façon sur les régions codantes (considérablement contraintes) que sur les non-codantes (plus tolérantes en terme d'évolution). Ceci se traduit par une différence de la quantité de mutations accumulées au cours de l'évolution dans les exons (que l'on cherche à localiser dans le cadre de notre problématique) par rapport au reste du génome. Les exons, qui évoluent plus lentement que les autres éléments, voient leur séquence nucléique davantage conservée. La

comparaison de séquences contenant des homologies peut donc apporter une information significative sur le caractère codant des régions génomiques. La Fig. 5.1 page précédente illustre la notion d'homologie.

Remarque 16 *À noter que les régions codantes ne sont pas les seules à être fonctionnelles, et donc conservées évolutivement. Des sites de fixation avec des protéines comme les promoteurs ou les régions régulatrices (Jareborg* et al.*, 1999; Hardison* et al.*, 1997) ou des gènes produisant des ARN fonctionnels ne codant pas pour des protéines subissent également une pression de sélection conséquente. Une des difficultés est précisément de distinguer ce type de conservation de celles qui sont révélatrices de régions codantes.*

Par rapport aux autres sources d'information de type *similarité* que sont les données d'expression ARN et protéines (voir le chapitre 2), l'approche par comparaison de séquences génomiques présente deux avantages.

En premier lieu, les données d'expression ont le défaut d'être incomplètes, dans le sens où l'on ne peut jamais être sûr d'avoir identifié la totalité des séquences transcrites ou traduites pour une espèce donnée. Puisque les gènes ne s'expriment pas tous simultanément, en permanence, dans toutes les cellules et en produisant un grand nombre de molécules, rien ne garantit qu'une série de prélèvements d'ARN ou de protéines permette de recueillir un échantillon exhaustif recouvrant la totalité des gènes de l'espèce, même en diversifiant les conditions d'expérience et les types de cellule (Ashburner, 2000)[3]. De plus, pour des raisons expérimentales, le séquençage d'un transcrit produit difficilement une séquence recouvrant la totalité de la molécule. En revanche, la séquence génomique contient par définition l'intégralité de l'information génétique.

Le deuxième défaut des données d'expression est leur manque de fiabilité. Pour des raisons de coûts expérimentaux, les transcrits ne sont pas toujours séquencés "proprement", c'est-à-dire que les séquences obtenues contiennent fréquemment des erreurs (dont la quantité peut rendre la séquence inexploitable). En revanche, la séquence génomique qui représente le support fixe et originel de l'information bénéficie généralement d'une attention plus soutenue.

Pour conclure sur la présentation du contexte, on peut évoquer l'avenir prometteur de l'approche par génomique comparative pour la prédiction de gènes (Brent et Guigó, 2004), en particulier si l'on considère l'augmentation régulière du nombre de génomes complets disponibles.

[3] où l'on peut lire par exemple : *"Now consider that under regular laboratory conditions only one-fourth or so of the genes of* Escherichia coli *are transcribed in log-phase growth, and the severity of the problem is obvious to all—who can refute the hypothesis that a gene is only transcribed in four brain cells (two on each side) during mating ?"*

ROSETTA	Batzoglou *et al.* (2000)	Ass. exons
CEM	Bafna et Huson (2000)	Ass. exons
SGP-1	Wiehe *et al.* (2001)	Ass. exons
PRO-GEN	Novichkov *et al.* (2001)	Align. global
TWINSCAN	Korf *et al.* (2001)	GHMM
DIALIGN	Morgenstern *et al.* (2002)	Align. global
DOUBLESCAN	Meyer et Durbin (2002)	PHMM
AGENDA	Rinner et Morgenstern (2002)	
	Taher *et al.* (2003, 2004a,b)	Ass. exons
SLAM	Alexandersson *et al.* (2003)	
	Cawley *et al.* (2003); Dewey *et al.* (2004)	GPHMM
SGP2	Parra *et al.* (2003)	Ass. exons
UTOPIA	Blayo *et al.* (2003)	Align. global
EUGÈNE'HOM	Foissac *et al.* (2003)	DAG
EVOGENE	Pedersen et Hein (2003)	EHMM
PROJECTOR	Meyer et Durbin (2004)	PHMM

TAB. 5.1 – Prédicteurs de gènes exploitant des informations d'homologies entre séquences génomiques. Sont indiquées les références et les méthodes correspondantes. Align. global = alignement global. Ass. exons = assemblage d'exons. (G)(P)HMM = *"(Generalized) (Pair) Hidden Markov Models"*. DAG = *"Directed Acyclic Graph"*. EHMM = *"Evolutionary HMM"*.

II ETAT DE L'ART

Le principe de la localisation de gènes par exploitation d'homologies est donc de détecter des régions de conservation évolutive par comparaison (alignement) de séquences, et de prendre en compte cette information pour produire une annotation.

Les premiers logiciels à développer cette approche furent probablement UTOPIA (Blayo *et al.*, 2003)[4], ROSETTA (Batzoglou *et al.*, 2000) et CEM (Bafna et Huson, 2000). Depuis, la liste ne cesse d'augmenter (voir le tableau 5.1, où figurent notamment les références bibliographiques des logiciels évoqués dans cette section). On peut distinguer deux approches principales adoptées par ces logiciels :

- une approche résolvant de façon simultanée les problèmes d'alignement et de prédiction de structure de gènes.
- une seconde approche traitant les deux problèmes en séquence.

[4]dont la méthode était diffusée depuis 1999 sans être publiée.

1/ Alignement global et prédiction simultanés

a. Description

À partir de deux séquences génomiques données, la philosophie de cette approche consiste à réaliser de façon simultanée un alignement entre ces deux séquences et une prédiction de leur structure génique. La résolution couplée des deux problèmes, qui sont interdépendants, est probablement l'approche la plus satisfaisante du point de vue de sa formulation. L'alignement recherché est global, c'est-à-dire que les séquences sont alignées sur toute leur longueur, produisant des régions d'appariements qui sont généralement interprétées comme étant codantes et des régions de mésappariement ou de brèches interprétées comme étant non-codantes.

Les logiciels, qui prennent donc en entrée deux séquences génomiques et produisent une double prédiction de gènes, peuvent être classés en deux catégories distinctes selon la méthode qu'ils utilisent : soit une extension d'un algorithme d'alignement global de séquence s'apparentant à la méthode de Needleman et Wunsch (1970) ; c'est le cas de Pro-Gen (Novichkov *et al.*, 2001), Dialign (Morgenstern *et al.*, 2002) et Utopia (Blayo *et al.*, 2003). Soit une extension d'un modèle probabiliste de type HMM, qui se base sur un modèle appelé PHMM (pour *"Pair(wise) Hidden Markov Models"*) ; c'est le cas de DoubleScan (Meyer et Durbin, 2002), SLAM (Alexandersson *et al.*, 2003) et Projector (Meyer et Durbin, 2004)[5]. Ce dernier type d'approche fut développé spécifiquement pour la génomique comparative entre séquences orthologues. Il s'agit d'un modèle utilisé par ailleurs pour l'alignement entre séquences (Durbin *et al.*, 1999) se différenciant des HMM par le fait qu'à chaque état ne correspond pas qu'une seule variable aléatoire, mais une paire. Les PHMM permettent d'intégrer de façon appropriée la problématique de l'alignement entre deux séquences et celle de leur annotation au sein d'un même modèle probabiliste.

b. Limitations

Ces approches présentent des défauts communs :

L'inconvénient majeur de ces méthodes concerne la complexité en temps de calcul des algorithmes utilisés par ces logiciels. En effet, les algorithmes d'alignement globaux prenant en compte les possibilités de longues régions de brèches présentent en général une complexité quadratique en temps et en espace avec la taille des séquences (Needleman et Wunsch, 1970). Si des méthodes permettent de réduire la taille de mémoire nécessaire (Meyer et Durbin, 2002), le temps de calcul exigé reste cependant une sérieuse limitation lorsque l'on s'intéresse à des projets d'annotation de grande envergure. Notons en outre que toutes ces méthodes sont limitées à exploiter l'information d'homologie contenue dans seulement deux séquences génomiques,

[5] ce récent logiciel est une variation de DoubleScan, qui nécessite que la structure de gènes de l'une des deux séquences génomiques soit connue pour annoter la seconde. À noter que cette approche se basant sur des annotations entraîne un risque considérable de contribuer au problème dramatique de propagation d'erreurs dans les bases de données.

car envisager leur extension à une prise en compte d'alignements multiples verrait la complexité en temps de calcul devenir exponentielle.

Ensuite, le champ d'application de ces méthodes est réduit en raison d'une réduction de leur espace de recherche. En effet, ne peut être considéré comme exon dans l'une ou l'autre des séquences qu'une région homologue, c'est-à-dire conservée dans les deux séquences. Cette contrainte ne paraît pas dramatique de prime abord lorsque l'on se limite à l'analyse comparative de deux génomes d'espèces proches évolutivement, comme l'humain et la souris, qui semblent présenter un taux de conservation très élevé entre gènes (Mouse Genome Sequencing Consortium, 2002)[6]. Cependant, un gène dont un seul des exons n'est pas conservé ne peut être correctement identifié. Notons que dans le cadre des PHMM, il est théoriquement possible de remédier à ce défaut en augmentant le nombre d'états possible (et par voie de conséquence le temps de calcul). Enfin, étendre l'idée de l'alignement global à des séquences multigéniques implique l'hypothèse que les gènes des deux espèces sont situés dans le même ordre et la même orientation, ce qui fait abstraction des évènements de réarrangement chromosomique.

Une autre limitation découle de la nécessité de disposer pour annoter une séquence génomique donnée d'une séquence homologue unique "recouvrant" la même région de toute sa longueur, ce qui en pratique n'est pas systématique. En effet, pour des raisons expérimentales, la longueur d'une molécule d'ADN entière que l'on peut "stocker" et séquencer est limitée ; c'est pourquoi le séquençage d'un génome passe par une étape de fragmentation, de lecture des fragments génomiques, puis de leur assemblage. Cette dernière étape peut s'avérer relativement longue, et les données génomiques disponibles pour l'annotation sont dans un premier temps sous la forme de fragments non assemblés. Ceci restreint l'application de méthodes utilisant l'approche d'alignement global, car pour une séquence génomique donnée il n'est pas garanti de disposer d'une séquence homologue ininterrompue sur toute sa longueur.

2/ Alignements locaux puis prédiction

La philosophie de cette seconde approche, considérant toujours deux séquences, est de séparer distinctement l'étape de recherche d'homologie par alignement de l'étape de prédiction de structure de gènes. En général, des algorithmes d'alignements locaux sont utilisés au préalable (externes ou développés *ad hoc*). Puis, dans un second temps, les régions d'homologie ainsi détectées sont utilisées pour aider la prédiction de gènes. Le traitement séparé de ces deux problèmes interdépendants est moins élégant mais se prête mieux, on le verra, à la prise en compte simultanée d'informations supplé-

[6] il nous semble toutefois important de garder à l'esprit que les gènes homologues bien conservés sont naturellement plus faciles à détecter que les autres ; par conséquent, il est facile d'en surévaluer la proportion, surtout si l'on envisage que l'ensemble des gènes humains — et à plus forte raison murins — connus n'est ni complet ni correct. Les estimations de proportion de gènes conservés sont donc à prendre avec prudence dans l'état actuel des connaissance.

mentaires.

Là aussi, deux familles de logiciels sont discernables, selon l'utilisation qui est faite des ces régions conservées : soit ces paires de régions homologues sont considérées comme des exons conservés et assemblées pour produire une double prédiction de gènes, à la façon de la méthode d'assemblage de segments vue dans le chapitre 2 (page 38) ; c'est le cas de ROSETTA (Batzoglou *et al.*, 2000), CEM (Bafna et Huson, 2000), SGP-1 (Wiehe *et al.*, 2001) et AGENDA (Taher *et al.*, 2004a). Soit ces homologies locales sont intégrées au sein d'une version étendue d'un prédicteur de gènes existant de type *ab initio* ; c'est le cas de TWINSCAN (Korf *et al.*, 2001) et de SGP2 (Parra *et al.*, 2003), extensions respectives de GENSCAN (Burge et Karlin, 1997) et GENEID (Parra *et al.*, 2000) intégrant des informations d'homologie locale dans un cadre probabiliste. Dans ce cas, une seule séquence est annotée, l'autre ne servant que de support de recherche d'homologies.

À la première famille de logiciels (assemblant les paires de régions) peut être adressé le même reproche sur la restriction de l'espace de recherche que pour l'approche d'alignement global, car une conservation de la structure exon-intron est nécessaire entre les gènes des deux séquences (voir ci-dessus).

L'affaire n'est pas la même pour la deuxième famille de logiciels, qui bénéficient des propriétés des prédicteurs *ab initio* servant de base à l'extension, notamment en ce qui concerne la modélisation d'un espace de recherche complet. Pour un exon, nulle obligation d'être conservé pour faire partie de la prédiction.

La limitation principale de l'approche est qu'elle n'est basée que sur un seul génome informatif. Il en découle une spécificité vis-à-vis des espèces considérées (en majorité homme-souris) et notamment en ce qui concerne la distance évolutive qui les sépare, facteur d'importance considérable (Zhang *et al.*, 2003). Cette limitation de la quantité d'information exploitable est regrettable, puisque de plus en plus de génomes sont disponibles, augmentant le nombre d'orthologues potentiels. Notons cependant que rien n'empêche en théorie l'extension de ces outils en permettant la recherche d'homologies dans différents génomes. Concernant les gènes paralogues, spécialement nombreux chez les plantes, ils sont rarement évoqués. Enfin, ces approches ne prennent en compte en plus de l'information *ab initio* que des données de type homologie, et pas de similarités avec des séquences exprimées.

Remarque 17 *Nous ne traitons pas ici les récentes approches appliquant des modèles évolutionnistes pour exploiter plusieurs séquences génomiques (Pedersen et Hein, 2003; Siepel et Haussler, 2004). Bien que prometteuses, elles ne permettent pas encore d'aboutir à des améliorations pratiques en termes de performances.*

III Mise en application dans EuGène'Hom

1/ Objectif

Pour élargir le potentiel d'EuGène, nous avons développé une version appelée EuGène'Hom (Foissac *et al.*, 2003), qui pour annoter une séquence génomique prend en compte des homologies détectées parmi un ensemble d'autres séquences. Cette version diffère d'EuGène simplement par les nouvelles sources d'informations qu'elle peut intégrer, et par les valeurs des paramètres associés.

L'objectif de cette version se décompose en plusieurs points :

- garder tous les avantages du prédicteur EuGène (espace de recherche, cohérence des prédictions, ensemble des informations intégrées...).
- exploiter l'information de type homologie.
- pouvoir intégrer des homologies avec plusieurs séquences génomiques
- ne pas être spécifique d'une seule espèce à annoter.
- ne pas demander d'hypothèse de conservation de structure exon-intron.
- pouvoir traiter des homologies intra- (paralogues) et inter-génomiques (orthologues).
- et tout cela de façon suffisamment efficace pour envisager une application à grande échelle.

2/ Détection des homologies

Concernant la partie de recherche d'homologies, nous avons choisi d'utiliser le logiciel TblastX (Altschul *et al.*, 1990; Gish et States, 1993; Altschul *et al.*, 1997), à qui l'on fournit la séquence génomique à annoter comme requête et un ensemble d'autres séquences génomiques où chercher les similarités. Deux raisons motivent le choix de ce logiciel.

D'une part, il s'agit d'un programme d'alignement local. Les structures de gènes homologues peuvent varier d'une espèce à une autre (orthologues) ou au sein d'une même espèce (paralogues). Certains domaines fonctionnels[7] peuvent être partagés par différentes paires de protéines. Contrairement à un alignement global qui exige une conservation tout le long d'une seule paire, un alignement local permet l'identification de plusieurs segments homologues avec différentes séquences.

D'autre part, l'alignement s'effectue par comparaison des séquences protéiques obtenues par traduction virtuelle des séquences nucléiques (chacune dans les 6 phases, comme expliqué dans la remarque 4 page 8). Le choix de comparer des séquences d'acides aminés plutôt que des séquences d'acides nucléiques permet d'augmenter la portée de la détection en terme de distance évolutive. En effet, en raison de la dégénérescence du code génétique, pour une protéine assurant une fonction biologique donnée, la variabilité possible

[7] partie de la protéine caractérisée par une structure et une fonction biochimique donnée

de la séquence d'ADN du gène correspondant est supérieure à celle de la séquence protéique (puisque pratiquement tous les acides aminés peuvent être codés par plusieurs triplets). Au-delà d'une certaine divergence évolutive, les similarités entre les séquences de deux gènes homologues peuvent ne plus être détectables qu'au niveau des acides aminés. Ainsi, préférer le niveau protéique au nucléique permet une détection plus sensible du signal évolutif entre organismes distants.

Concernant des distances plus proches entre séquences, le niveau protéique offre un autre avantage par rapport au nucléique, car il est supposé permettre une meilleure distinction entre les régions conservées codantes (exons) et les régions conservées non-codantes (signaux fonctionnels). En effet, entre organismes proches notamment, les séquences nucléiques de certains sites régulateurs d'expression génique peuvent être bien conservés et considérés à tort comme appartenant à des exons. Le passage en séquence d'acides aminés peut mettre en valeur certaines mutations dans les régions non-codantes par rapport à celles que l'on observe dans les régions codantes, qui tendent à ne pas perturber la séquence d'acides aminés ou les propriétés physico-chimiques.

Un dernier avantage de la comparaison au niveau protéique est qu'elle couvre un éventail plus large de distances évolutives entre les séquences qu'au niveau nucléique. Les alignements entre séquences d'acides aminés impliquent des matrices de distance plus complexes qu'entre les acides nucléiques. Ces matrices servent à attribuer un score pour tout appariement entre deux éléments en vis-à-vis dans un alignement. La façon la plus simple est de récompenser l'appariement et de pénaliser le mésappariement par des scores fixes, ce qui est fréquent pour les séquences d'ADN. Or, le cas des protéines est différent. L'effet sur la fonction biochimique d'une protéine que peut avoir la substitution d'un acide aminé pour un autre dépend fortement de la nature des acides aminés (encombrement stérique[8], charges électriques, ...). Certains acides aminés, similaires en terme de propriétés, sont fréquemment trouvés à la même position dans des alignements de protéines orthologues, alors que d'autres sont pratiquement incompatibles et rarement associés. Pour effectuer les alignements protéiques, il existe des matrices de substitution créées pour refléter ces distances, et qui associent un score à chaque paire d'acides aminés possible. Ici, puisque l'on s'intéresse uniquement aux conservations codantes, la matrice de substitution utilisée pour la recherche d'homologie avec le programme TBLASTX est modifiée pour pénaliser fortement (score de -500) toute paire contenant un codon stop afin de limiter le nombre d'alignements faux positifs.

3/ Intégration dans le graphe

La recherche de similarités par le logiciel TBLASTX fournit un ensemble d'alignements, chacun impliquant une partie de la séquence génomique à annoter dans une phase donnée avec une partie d'une séquence génomique

[8] occupation de l'espace de la molécule.

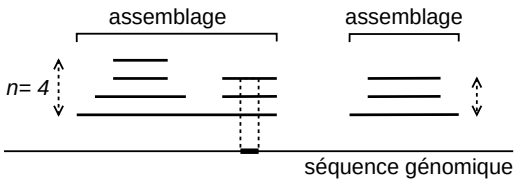

FIG. 5.2 – Alignements entre la séquence génomique (en bas) et les régions de similarité détectées par TBLASTX de même phase, regroupées en deux assemblages distincts. n est le nombre maximum d'alignements superposés dans le même assemblage.

de l'ensemble de recherche. Chaque alignement est présenté sous forme de séquences protéiques (après traduction virtuelle). Toutes les paires d'alignements chevauchants de phase identique sont regroupées en assemblages comme illustré en Fig. 5.2.

Pour intégrer la notion d'homologie dans le graphe d'EUGÈNE, on calcule pour chaque nucléotide x_i^j de position i et de phase j un score d'homologie. Pour cela, on considère l'assemblage comprenant la position i et de phase j. Soit n le nombre maximum d'alignements dans l'assemblage qui chevauchent la même position, et $c(x_i^j)$ le codon contenant le nucléotide x_i de phase j. Pour chaque alignement h de l'assemblage a, on note $S[c(x_i^j), h]$ le score donné par la matrice de substitution entre l'acide aminé de la séquence génomique (codé par $c(x_i^j)$) et l'acide aminé en vis-à-vis dans l'alignement h. Ce score est considéré comme nul si l'alignement h ne contient pas la position i. On définit un score d'homologie par la formule

$$S^H(x_i^j) = \omega \cdot \frac{1}{n} \sum_{h \in a} S[c(x_i^j), h]$$

où ω est le paramètre associé à la source d'information d'homologie, qui rejoint les autres dans le processus d'optimisation. L'idée de cette formulation est de prendre en compte le nombre d'alignements identifiés pour une région donnée tout en corrigeant un défaut du programme TBLASTX. En effet, ce programme a tendance à "déborder" autour des exons, poursuivant les alignements de part et d'autre, ne serait-ce que du fait de la conservation résiduelle des introns entre espèces proches. À l'intérieur d'un même groupe d'assemblage (Fig.5.2), une position génomique contenant peu de séquences homologues en vis-à-vis (généralement les extrémités de la région d'homologie) sera moins récompensée qu'une autre.

4/ Vers une généricité concernant l'espèce ciblée

Une propriété très intéressante de l'homologie est qu'il s'agit d'un phénomène universel, car toutes les espèces vivantes ont des gènes codant pour des protéines assurant les mêmes fonctions. L'exploitation de ce type d'information offre donc une opportunité intéressante de se libérer au moins partiellement de la contrainte de spécificité du logiciel pour une espèce donnée. En effet, cette spécificité peut s'avérer gênante si l'on souhaite annoter le génome d'une espèce pour laquelle les prédicteurs de gènes ne proposent pas de version spécifique. Il est alors recommandé d'en créer une en entraînant

un logiciel sur un jeu de données relatif à l'espèce d'intérêt, car utiliser une version entraînée pour une autre espèce peut causer des pertes drastiques de performance (Korf, 2004). Malheureusement, un apprentissage n'est pas toujours possible car les jeux de données font défaut pour la grande majorité des espèces. Cette spécificité étant due essentiellement aux modèles caractérisant les contenus statistiques des différentes régions, nous avons donc développé un modèle du codant indépendant de l'espèce.

a. Type *contenu* : modèle générique du codant

Pour caractériser le potentiel codant/non-codant d'une région donnée de façon indépendante de l'espèce considérée, nous avons utilisé un modèle probabiliste en deux étapes. Nous considérons qu'une séquence codante est générée en premier lieu comme une séquence d'acides aminés selon un modèle de Markov, puis traduite dans le sens inverse (de l'acide aminé vers le codon) suivant un usage du code le moins informatif possible. Cette approche, détaillée ci-dessous, est comparable dans l'esprit aux travaux de Staden (1984b) et de McCaldon et Argos (1988) évoqués dans le chapitre 2.

Il faut donc en premier lieu construire un modèle de Markov protéique (caractérisant des probabilité d'acides aminés), ce qui est fait à partir de l'ensemble des protéines de la base de données SwissProt (Boeckmann *et al.*, 2003).

Puis, considérant la séquence génomique à annoter, la probabilité d'un codon donné est calculée en multipliant la probabilité de l'acide aminé correspondant selon le modèle markovien protéique par la probabilité que cet acide aminé soit codé par ce codon en particulier (ce qui correspond à la préférence en codons). Pour caractériser cette dernière, il est possible de se baser sur le $GC\%$ de la zone génomique considérée, facteur le plus influant sur la préférence en codons. Pour cette version d'EuGène'Hom, nous avons simplement considéré une distribution uniforme des codons synonymes. Ainsi, la probabilité d'un codon donné est la probabilité indiquée par le modèle protéique pour l'acide aminé correspondant que divise le nombre de codons synonymes. À l'image des informations de type *contenu* dans EuGène (voir la formule (3.1) de la page 65), le logarithme de cette probabilité définit la pénalité associée au codon. Cette pénalité est répartie uniformément pour les trois nucléotides composant le codon, venant s'ajouter aux arcs de type *contenu* de chaque position.

Pour les modèles non-codants, un simple modèle de Markov nucléique d'ordre 0 est estimé sur la séquence même à annoter.

Comparé à des modèles de Markov nucléiques, ce modèle protéique n'est pas aussi riche en terme de quantité d'information, mais présente l'avantage d'être plus robuste par rapport aux variations locales du $GC\%$ génomique (voir page 23), et de ne pas être spécifique d'une espèce en particulier.

b. Type *signal* : détection de sites fonctionnels

i) Motivation

Les prédicteurs de signaux utilisés jusqu'alors par EuGène (NetStart, SplicePredictor, NetGene2, voir page 65) présentent l'inconvénient de n'offrir qu'un choix restreint d'organismes comme cible, ce qui rend difficile l'application d'EuGène sur des espèces pour lesquelles ces logiciels n'ont pas été entraînés. Puisque les utilisateurs n'ont aucune possiblité de réaliser un entraînement sur leur propre jeu de données, ils sont dépendants de la disponibilité des versions du logiciel. Bien que cette spécificité permette d'atteindre des résultats satisfaisants pour la détection des signaux fonctionnels, elle a tendance à limiter la capacité d'adaptation d'EuGèneà de nouvelles espèces. Nous avons donc choisi de doter EuGène d'un prédicteur de signaux indépendant[9], facilement entraînable sur tout jeu de données d'une espèce nouvelle.

ii) Méthode

Afin d'apporter à EuGène son propre prédicteur de signaux biologiques (codon start, sites d'épissage), nous avons opté pour une méthode basée sur un modèle probabiliste de type WAM (*"Weight Array Model"*, voir page 27).

Plus précisément, deux modèles WAM d'ordre 1 sont nécessaires, l'un pour représenter les signaux vrais positifs, l'autre pour les faux positifs. Puisque les sites d'épissage sont apparemment plutôt bien conservés au sein d'un même groupe taxonomique (Wiehe *et al.*, 2001), les modèles sont construits à partir de jeux de données d'*A. thaliana* AraClean et AraSet (Korning *et al.*, 1996; Pavy *et al.*, 1999b). Les probabilités sont estimées à partir des fréquences de dinucléotides (mots de deux lettres) observés pour chaque position dans le contexte proche (de 2 à 4 nucléotides) autour des mots `ATG`, `GT` et `AG` (respectivement pour le codon start, le site donneur et le site accepteur), en distinguant ceux qui correspondent aux vrais positifs des autres.

Concernant l'utilisation des modèles, un score dérivant du logarithme d'un rapport de vraisemblance est calculé, comme présenté à la page 18. Si $P^M(x_i|x_{i-1})$ représente la probabilité donnée par le modèle M (VP pour les vrais positifs ou FP pour les faux positifs) de trouver le nucléotide x à la position i en fonction du nucléotide à la position $i-1$, le score S^{WAM} d'un site potentiel dont le contexte inclut la région allant de la position i_1 à i_2 est donné par

$$S^{WAM} = \sum_{i=i_1}^{i_2} -\log \frac{P^{VP}(x_i|x_{i-1})}{P^{FP}(x_i|x_{i-1})}$$

Dans le graphe d'EuGène, pour pénaliser l'arc permettant d'utiliser un tel signal (Fig. 3.2 page 63), ce score est mis à l'échelle suivant l'expression $(\alpha \cdot S^{WAM} + \beta)$, où α et β, comme pour les autres sources d'information de type signal sont des paramètres à inclure dans le processus d'optimisation.

[9] non seulement vis-à-vis des versions et donc des espèces proposées, mais aussi des licences d'exploitation des autres logiciels.

IV Evaluation des performances

1/ Estimation des paramètres de la version EuGène'Hom

Les sources d'information utilisées par la version EuGène'Hom sont donc

- type *contenu* : modèle de Markov protéique (pas de paramètre à optimiser) ;
- type *signal* : prédicteur de signaux de type WAM (deux paramètres α et β par type de signal ;
- type *similarité* : score d'homologie, un paramètre ω.

Ces nouvelles sources d'information sont intégrées en estimant les paramètres associés par le moyen de la procédure d'optimisation décrite dans le chapitre précédent. Ainsi, on obtient un ensemble de paramètres caractérisant la version EuGène'Hom.

2/ Le jeu de données

Le jeu de données utilisé est un ensemble de 63 gènes homologues de plantes, la plupart d'*A. thaliana*, aimablement fourni par S. Aubourg (Unité de Recherche en Génomique Végétale, INRA/CNRS, communication personnelle). Ces gènes proviennent de 5 familles de gènes (alcool déshydrogénase, terpène synthétase, cinnamyl alcool déshydrogénase, et deux familles de fonction inconnue). Pour permettre une comparaison avec d'autres prédicteurs qui fonctionnent généralement par paires de séquences, à partir de chaque famille de gènes les deux paires présentant la similarité la plus forte et la plus faible au niveau protéique sont extraites pour constituer le jeu de test, le reste définissant le jeu d'apprentissage. Après examen attentif des annotations de références, nous avons écarté 12 séquences en raison d'incohérences dans les annotations ou de sites d'épissage atypiques (autres que `GT-AG`), ce qui mène à un jeu d'apprentissage de 35 gènes et un jeu de test de 16 gènes (122 exons, 75 kb) organisé en 9 paires (le même gène pouvant appartenir à deux paires différentes).

3/ Etude comparative

a. Les logiciels comparés

Nous avons réalisé une comparaison d'EuGène'Hom avec GenScan (Burge et Karlin, 1997), un prédicteur de type purement *ab initio*, Sgp-1 (Wiehe *et al.*, 2001), un des prédicteurs les plus comparables basés sur l'homologie et disponibles par l'Internet, et deux autres versions d'EuGène, que nous appellerons EuGèneAt et EuGèneAt+Hom.

GenScan a été exécuté sur le jeu de test à partir de son site Internet[10] avec les options par défaut, sauf pour l'organisme choisi (*A. thaliana*).

[10] http ://genes.mit.edu/GENSCAN.html

Programme	SNe	SPe	SNn	SPn	AC
SGP-1	11.48	35.00	35.29	93.02	0.53
GENSCAN	52.46	49.61	62.86	79.80	0.61
EUGÈNEAT	63.11	63.11	82.91	74.57	0.69
EUGÈNE'HOM	76.23	54.07	96.37	71.24	0.75
EUGÈNEAT+HOM	88.52	74.48	97.80	76.34	0.80

TAB. 5.2 – Performances sur le jeu de test. SNe = sensibilité au niveau exon, SPe = spécificité au niveau exon, SNn = sensibilité au niveau nucléotide, SPn = spécificité au niveau nucléotide, AC = *approximate correlation* (Burset et Guigó, 1996). La précision au niveau gène n'a pas été calculée en raison de la petite taille du jeu de données, comprenant un total de 16 gènes, 122 exons et 75 kb.

SGP-1 prend en entrée deux séquences et produit les deux annotations correspondantes. Nous l'avons utilisé depuis son site Internet[11] sur les 9 paires du jeu de test, retenant la meilleure prédiction pour les gènes apparaissant dans deux paires. Pour une comparaison optimale avec EUGÈNE'HOM, les options choisies sont : *taxonomic group* = *angiosperm*, *alignment method* = TBLASTX, *no post-processing* (sauf pour trois séquences pour lesquelles l'option *monotonic increasing unique* devait être activée pour permettre l'exécution), *substitution matrix* = Blosum80.

EUGÈNEAT correspond à la version d'EUGÈNE optimisée pour *Arabidopsis thaliana* décrite dans le chapitre 3. Les sources d'information utilisées sont des modèles markoviens spécifiques et les logiciels de prédiction de signaux externes, sans exploiter d'information de conservation évolutive.

EUGÈNEAT+HOM est une version hybride obtenue par simple ajout du score d'homologie et du modèle protéique d'EUGÈNE'HOM dans la version EUGÈNEAT sans avoir réestimé les paramètres.

b. Les résultats obtenus

La qualité des prédictions est mesurée en termes de sensibilité et de spécificité au niveau exon et au niveau nucléotide comme expliqué à la page 54 d'après Burset et Guigó (1996). Les résultats sont présentés dans le tableau 5.2.

Les performances de SGP-1, GENSCAN et EUGÈNEAT sont inférieures à celles observées par ailleurs (Wiehe *et al.*, 2001; Schiex *et al.*, 2001). Ceci est probablement dû au grand nombre d'exons composant les gènes du jeu de test (8 par gène en moyenne). De plus, certaines séquences proviennent d'organismes autres que *A. thaliana*, perturbant ainsi les modèles probabilistes du codant de GENSCAN et d'EUGÈNEAT.

La comparaison des différentes versions d'EUGÈNE confirme le bénéfice apporté par les méthodes développées dans EUGÈNE'HOM, notamment l'intégration de l'information d'homologie entre plusieurs séquences génomiques.

Malgré la taille réduite du jeu de test, ces résultats montrent que même

[11] http ://soft.ice.mpg.de/sgp-1/sgp-1noemail.html

sans modèle probabiliste spécifique d'une espèce, EuGène'Hom parvient à prédire la structure exon-intron de gènes de plantes avec une précision tout à fait correcte.

V Conclusion

Grâce au précédent travail sur le processus d'optimisation, nous avons pu développer, intégrer et mettre en application des nouvelles sources d'information pour la prédiction de gènes à travers le logiciel EuGène et sa version EuGène'Hom disponible sur Internet[12].

Une méthode basée sur des recherches de similarités entre séquences génomiques par le biais d'alignement locaux permet de détecter les homologies de type orthologie aussi bien que paralogie. Il est intéressant de noter que le domaine d'application d'EuGène dépasse ainsi le cadre strict de la génomique comparative puisque des séquences du même organisme peuvent être utilisées. L'intégration dans le graphe s'effectue en adéquation avec les autres sources d'information grâce au processus d'optimisation.

Pour élargir le spectre d'action d'EuGène'Hom, nous l'avons pourvu d'un modèle générique du codant basé sur le contenu statistique des protéines en acides aminés, robuste aux variations de $GC\%$ génomique. Cette méthode offre également la possibilité d'estimer un potentiel codant de base pour toute séquence génomique, même si aucun jeu de données de référence n'est encore disponible pour l'espèce d'intérêt.

Enfin, en vue de libérer EuGène d'une dépendance vis-à-vis de logiciels de prédiction de signaux, nous l'avons doté d'une méthode de détection de tout signal biologique entraînable rapidement à partir d'un simple alignement de motifs. Bien que relativement rudimentaire et moins performante que celles des logiciels spécialisés plus complexes, cette méthode basée sur des WAM a été incorporée au sein du *"pipe-line"* d'annotation de nouveaux génomes utilisé actuellement par l'équipe EuGène pour servir automatiquement de prédicteur de signaux si un logiciel spécifique n'est pas disponible.

Concernant les résultats, malgré la taille réduite du jeu de données utilisé, l'évaluation comparée des performances d'EuGène'Hom montre que les méthodes développées au cours de ce travail permettent d'apporter une amélioration significative de la qualité des prédictions de gènes de plantes.

La puissance de l'approche de la génomique comparative est aujourd'hui indéniable pour l'annotation des génomes (Mouse Genome Sequencing Consortium, 2002; Zdobnov *et al.*, 2002). L'exploitation des homologies entre séquences génomiques représente un potentiel considérable d'amélioration des méthodes de prédiction de gènes. Malheureusement, les outils développés jusqu'ici ne proposent pas de cadre général d'intégration des sources d'information disponibles. Avec le développement des méthodes mises en application dans la version EuGène'Hom, il est possible d'envisager un sys-

[12] http ://genopole.toulouse.inra.fr/bioinfo/eugene/EuGeneHom/cgi-bin/EuGeneHom.pl

tème d'annotation à large potentiel intégratif, prêt à profiter de l'avancement des grands projets de séquençage systématique des génomes.

La science sans religion est boiteuse. La religion sans science est aveugle.

— Albert Einstein

La seule chose dont je sois vraiment sûr, c'est que nous sommes de la même étoffe que les autres bêtes ; et si nous avons une âme immortelle, il faut qu'il y en ait une aussi dans les infusoires qui habitent le rectum des grenouilles.

— Jean Rostand
(Pensées d'un biologiste, 1954)

La seule certitude que j'ai, c'est d'être dans le doute.

— Pierre Desproges

Chapitre 6

Intégration d'informations de transcriptomes

Alors que le chapitre précédent est consacré à l'exploitation de données génomiques, celui-ci présente les méthodes que nous avons développées pour améliorer l'intégration des connaissances apportées par les projets de séquençage massif de transcriptomes.

Le séquençage à grande échelle des ARNm d'une espèce permet d'accéder à une quantité considérable de connaissances sur son mode de fonctionnement génique (Okazaki *et al.*, 2002; Yamada *et al.*, 2003; Imanishi *et al.*, 2004). Ce type d'information de transcrits, plus rapide, plus simple et moins onéreux à séquencer que la séquence génomique est disponible en grande quantité dans les bases de données publiques. Comme nous l'avons vu dans le chapitre 2, de nombreux logiciels de prédiction de gènes (dont EuGène, comme décrit dans le chapitre 3) exploitent ce type d'information (Xu et Uberbacher, 1997; Krogh, 2000; Reese *et al.*, 2000a; Brendel *et al.*, 2004; Haas *et al.*, 2003; Eyras *et al.*, 2004).

Dans ce chapitre, nous traiterons de deux évolutions récentes de la façon de considérer les données de transcriptome. La première est dûe à l'apparition d'un nouveau type de données caractérisant des transcrits dans l'intégralité de leur longueur et à la volonté de leur prise en compte spécifique ; la deuxième évolution fait suite à la croissance de la quantité de données de transcriptome et à la reconsidération qui s'impose du phénomène de l'épissage alternatif.

I Intégration d'ADNc "pleine longueur"

Cette première partie du chapitre montre un exemple relativement simple d'intégration d'un nouveau type d'information dans EuGène, qui illustre remarquablement la flexibilité du modèle et son potentiel d'évolution.

1/ Contexte et motivation

L'identification d'un transcriptome s'effectue par séquençage des molécules transcrites dans un ensemble de cellules. En général, ce séquençage passe par la production à partir d'ARNm de molécules appelées ADNc (pour ADN complémentaire) car il s'agit de la copie sur support ADN de l'information portée par l'ARNm (séquence codante CDS + UTR5' et UTR3', rappel en Fig. 1.3 page 5). Malheureusement, il se trouve que pour des raisons expérimentales[1] il s'avère difficile de synthétiser des ADNc contenant l'intégralité de la séquence des ARNm correspondants. Par conséquent, des techniques ont été mises au point pour améliorer la production de tels ADNc (Maruyama et Sugano, 1994; Carninci *et al.*, 1996, 1997; Kato *et al.*, 1994; Edery *et al.*, 1995; Carninci et Hayashizaki, 1999; Zhu *et al.*, 2001), appelés ADNc "pleine longueur" ou ADNc complets.

En 2002, par l'application d'une de ces méthodes (Carninci *et al.*, 1996, 1997) l'institut de recherche japonais RIKEN isole et caractérise près de 15000 ADNc distincts d'*Arabidopsis thaliana* qualifiés de "pleine longueur" (*"full-length cDNA"*). Ces ADNc complets représentent une occasion d'améliorer la qualité de l'annotation de référence du génome de l'arabette, car la garantie qu'un ADNc est complet apporte une information supplémentaire considérable par rapport à un transcrit partiel de type EST.

C'est dans ce contexte que le programme de recherche en génomique végétale GÉNOPLANTE[2] (Samson *et al.*, 2003) lance le projet CATMA[3] (Crowe *et al.*, 2003), pour *"Complete* Arabidopsis *Transcriptome MicroArray"*. L'objectif est de construire une puce à ADN (*"microarray"*) pour permettre une étude du transcriptome complet d'*Arabidopsis thaliana*. Sans entrer dans les détails, cela revient à créer un outil de mesure du niveau d'expression des gènes. Un tel outil demande une connaissance de la position et de la structure de l'ensemble des gènes dans le génome de l'espèce, et donc une annotation de la meilleure qualité possible.

C'est pour cette raison que les laboratoires de biologie impliqués dans le projet se sont tournés vers le logiciel EUGÈNE, et l'ont choisi pour assurer la partie annotation *in silico* du projet. Or, la disponibilité des ADNc complets produits par l'institut RIKEN permettait d'envisager la prise en compte d'un nouveau type d'information précieuse. C'est pour répondre à cette opportunité que nous avons mis au point une méthode d'intégration spécifique des données de type ADNc "pleine longueur" ou complets.

[1] principalement à cause de décrochement précoce de la transcriptase reverse qui ne parvient donc pas à l'extrémité de l'ARNm, combiné à l'influence de la taille de l'ARNm sur la RT-PCR et le clonage.

[2] http ://www.genoplante.com/htm/prehome.html

[3] http ://genoplante-info.infobiogen.fr/projects/CT_Nouveaux_Outils/NO2001040/index.php

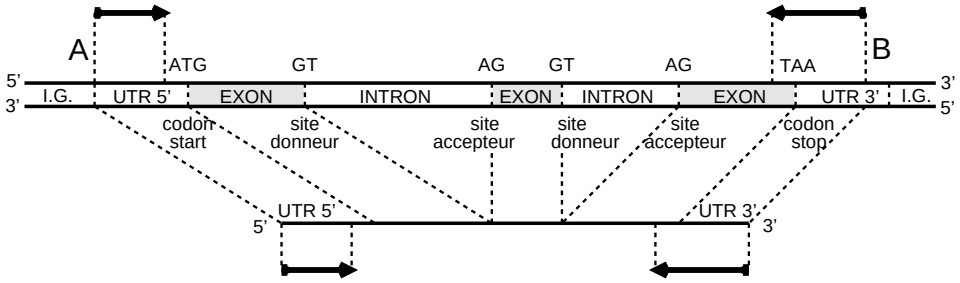

FIG. 6.1 – Exemple d'ADNc complet, avec la position des séquences de ses extrémités sur la séquence génomique. Le gène représenté est le même que celui de la Fig. 1.3 page 5. L'ADNc "pleine longueur" correspondant est schématisé dessous. Les flèches correspondent à ses parties terminales séquencées. Leur alignement avec la séquence génomique est représenté dessus. Dans cet exemple, l'ADNc n'est pas tout à fait complet, car son extrémité 3' n'arrive pas au bout de l'UTR. A et B indiquent les positions terminales de la paire alignée. I.G. = intergénique.

2/ Comment intégrer cette information ?

a. Comment se présente-t-elle ?

Au moment où cette intégration était envisagée, les ADNc n'étaient pas tous séquencés entièrement. En revanche, pour chacun d'eux, on disposait de la séquence des deux extrémités sur une longueur de quelques centaines de bases, à la façon des EST. L'annotation d'une séquence génomique passait donc par une recherche préalable de similarité avec ces paires d'extrémités d'ADNc complets. Lorsqu'une extrémité s'aligne sur la séquence génomique, cela signifie qu'un gène est présent, et qu'une des extrémités de sa partie transcrite correspond à une des extrémités de l'alignement (Fig. 6.1). Les positions correspondantes ainsi que le nom de l'ADNc (pour pouvoir identifier les deux extrémités correspondant au même gène) sont notés dans un fichier que peut analyser EUGÈNE. Pour résumer, pour chacun des 15000 gènes de l'étude, on dispose de l'alignement génomique des deux EST provenant des extrémités de son ADNc complet.

Après une analyse attentive (et de nombreuses interactions avec les utilisateurs) il s'est avéré que les ADNc n'étaient pas tous parfaitement complets, c'est-à-dire que parfois l'extrémité de la partie transcrite du gène n'était pas comprise dans l'alignement génomique des extrémités de l'ADNc (Sébastien Aubourg, INRA Evry et Carine Serizet, Université de Gent, communication personnelle). Nous faisons l'hypothèse que si les UTR ne sont pas toujours complètes, la séquence codante du gène (CDS, du start au stop) est comprise dans son intégralité entre les positions terminales.

b. L'intégration dans le graphe d'EUGÈNE

i) L'information apportée

Lorsqu'une paire de fragments terminaux d'un ADNc "pleine longueur" est alignée sur la séquence génomique, on identifie les deux positions termi-

nales : puisque chaque fragment aligné est localisé par deux positions sur la séquence génomique, les deux positions définissant l'intervalle le plus grand sont appelées terminales (Fig. 6.1 page précédente).

Un tel alignement indique qu'un gène, et un seul, se trouve entre ses deux positions. De plus, il est supposé être complet, au moins concernant sa séquence codante.

ii) Pondération des pistes

Soient A et B les positions terminales respectivement de gauche et de droite. Le traitement opéré sur le graphe est le suivant :

- entre A et B : interdiction de la piste intergénique. Ceci est obtenu par une pénalisation d'un poids infini de tous les arcs *contenu* de la piste intergénique entre A et B. Puisque la nature du graphe d'EuGène l'oblige a passer par de l'intergénique pour séparer deux gènes distincts (Fig. 3.1 page 61), la conséquence de cette interdiction est la garantie qu'un seul gène peut être prédit entre les positions terminales.
- à la position A : interdiction de toutes les pistes sauf celle de l'UTR5'F (F pour *"Forward"*, cas d'un gène dans le sens direct de gauche à droite) et celle de l'UTR3'R (R pour *"Reverse"*, cas d'un gène dans le sens inverse de droite à gauche). Comme ci-dessus, l'interdiction d'une piste s'effectue par pénalisation infinie de l'arc *contenu* de la position d'intérêt.
- à la position B : de façon comparable, interdiction de toutes les pistes exceptées celles de l'UTR5'R et de l'UTR3'F.

Le premier point empêche la coupure du gène et les deux autres sa fusion avec un gène voisin (car la seule piste qui peut succéder aux UTR de part et d'autre des positions terminales est la piste intergénique). Ainsi, bien que l'on force EuGèneà prédire un gène unique dans la région impliquée, une certaine souplesse est conservée dans la mesure où le choix de la structure précise de ce gène (exons, introns, start, stop) est laissés à "l'appréciation" du logiciel, c'est-à-dire en fonction des autres sources d'information.

3/ Conclusion

Grâce à la souplesse et au potentiel d'évolution du graphe d'EuGène, nous avons pu facilement prendre en compte un nouveau type d'information spécifique. Le développement de cette méthode a demandé de nombreuses interactions enrichissantes avec les utilisateurs d'EuGène du côté de l'annotation. Le résultat obtenu a été concrétisé et valorisé à travers la réannotation du génome d'*Arabidopsis thaliana* qui a servi de support pour la réalisation de puces à ADN destinées à mesurer des niveaux d'expression génique dans le cadre du projet CATMA (Crowe *et al.*, 2003).

La possibilité d'incorporer spécifiquement des données de transcrits de type "pleine longueur" n'est proposée à notre connaissance dans aucun autre logiciel de prédiction de gènes. Elle est cependant destinée à être exploitée à l'avenir, car d'autres espèces font également l'objet de programmes d'iso-

lation et de séquençage d'ADNc complets, ce qui représente l'opportunité d'améliorer les annotations génomiques produites jusqu'alors.

Cet exemple simple d'exploitation de données met en évidence l'importance capitale dans le domaine de la prédiction de gènes d'un modèle souple et évolutif permettant une intégration aisée d'informations nouvelles.

II Extension du modèle à l'épissage alternatif

Cette seconde partie du chapitre traite de la prise en compte du phénomène de l'épissage alternatif dans le domaine de la prédiction de gènes. Après un rappel sur le processus biologique, nous présenterons les raisons de l'intégration de ce phénomène d'actualité, un état de l'art sur son influence dans le domaine de la prédiction de gènes, et comment nous l'avons incorporé au sein du logiciel EuGène.

1/ Problématique

a. Rappel

L'épissage alternatif est un phénomène biologique qui intervient au cours de l'expression génique et qui permet à la cellule de produire des protéines différentes à partir du même gène. On constate pour certains gènes plusieurs formes d'épissage possibles à partir du même pré-ARNm, chacune étant caractérisée par un ensemble distinct de sites d'épissage. Il en résulte une production possible de différents ARNm (appelés isoformes ou variants d'épissage, voir la Fig. 6.2 ci-contre), permettant par traduction la synthèse de protéines différentes.

Initialement considéré comme un évènement exceptionnel, l'épissage alternatif apparaît aujourd'hui comme un acteur majeur de la régulation de l'expression génique et de la diversité des transcriptomes et protéomes (Modrek et Lee, 2002a). La principale raison de ce changement d'égard est l'augmentation considérable ces dernières années de données de type transcriptome, qui ont permis par comparaisons des séquences des ARNm de mettre en évidence de nombreux cas d'épissage alternatif. On estime par exemple que chez *H. sapiens*, près de 2 à 3 gènes épissés sur 4 sont concernés par le phénomène (Modrek et Lee, 2002a; International Human Genome Sequencing Consortium, 2001; Johnson *et al.*, 2003; Leipzig *et al.*, 2004; Kim *et al.*, 2004). Le plus célèbre exemple du potentiel de diversité résultant de l'épissage alternatif est le gène Dscam (*"Down syndrome cell adhesion molecule"*) de la drosophile[4], à partir duquel plus de 38000 ARNm différents pourraient être produits (Schmucker *et al.*, 2000).

L'épissage alternatif, dont les mécanismes responsables de la régulation ne sont pas encore bien identifiés, bénéficie actuellement d'un intérêt considérable de la part de la communauté, et fait donc l'objet de nombreuses études (Itoh *et al.*, 2004; Gupta *et al.*, 2004; Miriami *et al.*, 2004; Roca *et al.*, 2003; Zhu *et al.*, 2003a; de la Mata *et al.*, 2003; Zavolan *et al.*, 2003). Les comparaisons à grande échelle de séquences transcrites ont permis en outre la création de nombreuses bases de données. Une liste est présentée dans la table 6.1. De plus amples informations sur l'épissage alternatif sont disponibles dans les publications de Black (2003); Modrek et Lee (2002a);

[4] *Drosophila melanogaster* : la mouche du vinaigre (du grec *drosos*, rosée, et *philos*, qui aime).

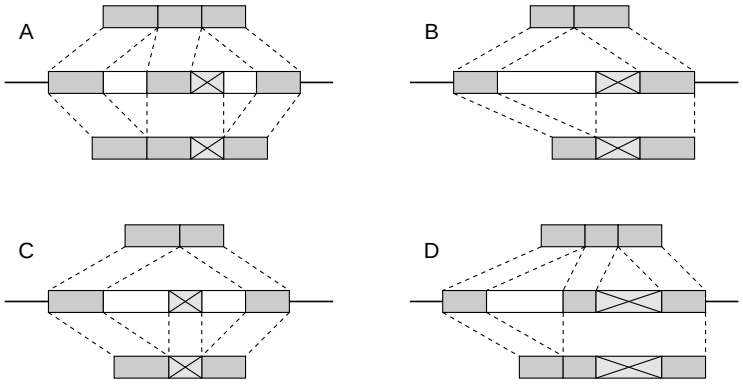

FIG. 6.2 – Schéma des différentes catégories d'épissage alternatif. A : site donneur alternatif. B : site accepteur alternatif. C : exclusion/inclusion d'exon. D : excision/rétention d'intron. Pour chaque cas, la séquence génomique est représentée par le trait horizontal, la structure du gène par des rectangles : foncés pour les exons, clairs pour les introns, et marqués d'une croix pour la partie responsable de la différence entre les deux variants d'épissage, schématisés au-dessus et au-dessous de la séquence génomique. À noter que ne sont représentés ici que les cas simples, mais que les combinaisons possibles peuvent augmenter le nombre de variants.

Maniatis et Tasic (2002); Singh (2002); Woodley et Valcarcel (2002); Lorkovic *et al.* (2000).

b. Quel rapport avec la prédiction de gènes ?

La reconsidération à la hausse de la fréquence et de l'importance de l'épissage alternatif soulève un sérieux problème concernant l'annotation des génomes, et plus particulièrement pour le processus de prédiction de gènes. En effet, les logiciels de localisation des séquences codantes produisent en général une seule structure exon-intron par séquence génomique alors que l'épissage alternatif permet à la cellule de construire plusieurs ARNm distincts par gène.

Pour avoir une idée de l'importance du problème et de l'intérêt que porte communauté à son égard, il suffit simplement de considérer les nombreux articles de la littérature dans lesquels l'épissage alternatif est évoqué dans les parties "discussion" ou "perspective", que ce soit dans les publications relatives à des logiciels de prédiction, en faisant généralement référence à de futurs développements (Snyder et Stormo, 1993; Chuang et Roth, 2001; Xu et Uberbacher, 1997; Yeh *et al.*, 2001; Meyer et Durbin, 2004), ou dans les articles indépendants d'évaluation ou de revue du domaine, typiquement dans la liste des problèmes d'avenir non résolus (Claverie, 1997; Wang *et al.*, 2003; Rogic *et al.*, 2001; Zhang, 2002; Mathé *et al.*, 2002; Brent et Guigó, 2004).

Suite à la prise de conscience que l'épissage alternatif est devenu incontournable pour tout projet d'annotation de génome, le domaine de la prédiction de gènes doit faire face à un défi capital. Comment intégrer l'éventualité

ASDB	Alternative Splicing DataBase (Gelfand *et al.*, 1999)
AEDB	Alternative-Exon DataBase (Stamm *et al.*, 2000)
HASDB	Human Alternative Splicing DataBase (Modrek *et al.*, 2001)
PALS	Putative Alternative Splicing database (Huang *et al.*, 2002)
SpliceNest	(Krause *et al.*, 2002)
AltExtron	(Clark et Thanaraj, 2002)
ProSplicer	(Huang *et al.*, 2003)
ASAP	Alternative Splicing Annotation Project (Lee *et al.*, 2003)
ASD	Alternative Splicing Database (Thanaraj *et al.*, 2004)
EASED	Extended Alternatively Spliced EST Database (Pospisil *et al.*, 2004)
ASG	Alternative Splicing Gallery (Leipzig *et al.*, 2004)

Tab. 6.1 – Bases de données dédiées à l'épissage alternatif.

d'épissage alternatif pour permettre la détection et l'identification de plusieurs variants par gène prédit ? C'est à cette question que nous tentons d'apporter des éléments de réponse dans la suite de ce chapitre.

2/ Etat de l'art

Différentes méthodes ont été développées pour répondre à ce problème. On distingue d'une part les méthodes *ab initio* ou intrinsèques, et d'autre part les méthodes par similarité ou extrinsèques.

a. L'approche *ab initio* (intrinsèque)

i) Méthodes et logiciels

Certains prédicteurs de gènes de type *ab initio* (n'utilisant pas d'information de type *similarité*) ont abordé le problème de l'épissage alternatif. On peut citer GenScan (Burge et Karlin, 1997), HMMgene (Krogh, 2000) et SLAM[5] (Alexandersson *et al.*, 2003). Pour ces prédicteurs, la structure de gène prédite correspond à la prédiction optimale, soit la plus probable selon le modèle sous-jacent (comme avec EuGène, à ceci près qu'il s'agit alors d'une optimalité de score et non de probabilité). Dans le cas d'épissage alternatif, une seule prédiction ne suffit pas puisque plusieurs structures de gènes coexistent. Chercher parmi les prédictions sous-optimales s'avère par conséquent intéressant, comme évoqué très tôt par Snyder et Stormo (1993) avec GeneParser, première application de la programmation dynamique pour la prédiction de gènes.

Pour un classique prédicteur basé sur un modèle de type HMM, il est possible par une simple modification de l'algorithme de Viterbi d'accéder

[5]ce dernier n'est pas exactement de type *ab initio* car il utilise les similarités entre 2 séquences génomiques ; il figure toutefois dans ce paragraphe car la méthode est liée au modèle probabiliste et non aux informations utilisées.

non pas uniquement à la solution (succession d'états cachés caractérisant une annotation) optimale, mais à l'ensemble des n meilleures solutions. On trouve cette approche initialement dans GeneParser (Snyder et Stormo, 1995)[6], puis par exemple HMMgene (Krogh, 2000) ou Fgenes-M (non publié).

Une autre façon d'obtenir des solutions sous-optimales est de faire de l'échantillonnage dans les . Cette méthode, proposée sous le nom de *"HMM sampling"* par Cawley et Pachter (2003) a été testée dans le logiciel SLAM (Alexandersson *et al.*, 2003). Cela consiste à générer aléatoirement des successions d'états (et donc des prédictions) en suivant les probabilités conditionnelles du modèle et de la séquence. En pratique, un nombre impressionnant d'échantillonnages est nécessaire pour obtenir une prédiction qui diffère de l'optimale.

GenScan adopte une approche différente, et cherche des exons qui sont probables mais absents de la prédiction optimale. Pour cela, un algorithme de type *"forward-backward"*—voir par exemple (Durbin *et al.*, 1999)—, permet d'identifier les exons dont la probabilité *a posteriori* est supérieure à un certain seuil, car ils appartiennent à un grand nombre de solutions sous-optimales. Toutefois, déterminer la valeur de ce seuil reste un problème délicat. Plus simplement, Zhang (1997) évoquait la possibilité que les exons de scores sous optimaux proposés par Mzef puisse faire partie de variants d'épissage.

ii) Limitations

Deux problèmes majeurs affectent ces méthodes *ab initio*, un qui concerne la sensibilité et l'autre la spécificité.

Défaut de sensibilité : Toutes ces méthodes font l'hypothèse qu'une solution caractérisant un variant d'épissage jouit d'une probabilité proche de l'optimale d'après le modèle sous-jacent. Cette supposition est discutable, surtout si l'on considère les variants dont la structure exon-intron diffère de façon significative de la structure optimale. Par exemple, dans le cas d'un événement d'exclusion d'exon (Fig. 6.2 page 113), si le variant le plus long dispose de la probabilité optimale, le plus court est complètement privé du caractère codant de l'exon manquant, et sa probabilité risque fort d'en pâtir. Un cas similaire concerne le variant dont les sites d'épissages le caractérisant sont dits "faibles", ce qui signifie que leur séquence n'est pas en bonne adéquation avec le modèle chargé de représenter ces signaux. La probabilité associée à ces variants souffre par conséquent de ce contraste, il est donc peu vraisemblable qu'elle soit proche de l'optimale. Or, des analyses à grande échelle montrent que de tels sites faibles sont fréquemment impliqués dans le phénomène d'épissage alternatif (Clark et Thanaraj, 2002; Itoh *et al.*, 2004), ce qui nuit également à ce type d'approche *ab initio*. Les signaux impliqués dans l'épissage font toujours l'objet de recherches, comme par exemple les ESE *"Exonic Splicing Enhancers"* (Cartegni *et al.*, 2003; Fairbrother *et al.*,

[6] où bien qu'il ne s'agisse pas de l'algorithme de Viterbi le principe est semblable.

2002), mais aucun outil spécifique et efficace de prédiction de sites d'épissage alternatif n'est disponible à ce jour.

Puisque les structures de gènes correspondant aux variants d'épissage risquent de se voir attribuer une probabilité bien inférieure à celle de la structure optimale, il semble indispensable pour les identifier de recourir à davantage d'information que n'en apporte la seule probabilité donnée par un modèle général.

Défaut de spécificité : Le problème de spécificité est évident. Dès lors que de nombreuses prédictions peuvent être produites de façon systématique pour chaque séquence génomique, comment distinguer celles qui correspondent à d'authentiques variants d'épissage de celles qui ne reflètent que des faux positifs virtuels ?

Aussi longtemps qu'un prédicteur fiable de sites d'épissage alternatif fera défaut, une méthode purement intrinsèque ne pourra se révéler satisfaisante.

b. L'approche par similarité (extrinsèque)

i) Méthodes et logiciels

Le principe de l'approche extrinsèque est de détecter les épissages alternatifs en analysant directement leurs produits, c'est-à-dire les transcrits, généralement alignés au préalable sur la séquence génomique. Des logiciels ont été créés pour réaliser et/ou exploiter des alignements génomiques de transcrits dans le but d'identifier la structure génique, avec ou sans variants d'épissage. Parmi de tels outils d'annotation par similarité, on peut citer GeneSeqer (Usuka *et al.*, 2000; Brendel *et al.*, 2004), ASPic (Bonizzoni *et al.*, 2003), TAP (Kan *et al.*, 2001, 2002), et PASA (Haas *et al.*, 2003). Excepté pour GeneSeqer, plutôt destiné à l'alignement, les autres adoptent la même stratégie : à partir d'alignements génomiques de transcrits, leur objectif est d'indentifier la (ou les) structure(s) exon-intron compatible(s) avec le maximum d'alignements. Dans le même esprit mais plus récemment, ClusterMerge (Eyras *et al.*, 2004) a été utilisé au sein du système d'annotation d'Ensembl (Curwen *et al.*, 2004) pour identifier les structures géniques, toujours à partir d'alignements de transcrits. Contrairement à l'approche *ab initio*, ces méthodes prennent en compte des informations de données d'expression. Cependant, elles sont également confrontées à divers problèmes.

ii) Limitations

Le principal inconvénient de ces méthodes par similarité est qu'elles sont exclusivement extrinsèques (à part quelques exceptions, comme pour TAP qui utilise une recherche de signal de poly-adénylation pour déterminer les frontières entre gènes adjacents, ou comme GeneSeqer, qui contient un modèle pour attribuer un score aux site d'épissage). Il découle de cette restriction une grande dépendance vis-à-vis des données de transcriptome, concernant aussi bien leur quantité que leur qualité. En effet, ces logiciels ne peuvent détecter un site d'épissage qu'à partir des alignements de transcrits disponibles. Par conséquent, tout gène qui ne bénéficie pas d'une couver-

ture totale de transcrits avec toutes ses frontières exon-intron ne peut être correctement prédit par ces outils, ce qui constitue une limitation majeure. Rappelons que les données de transcrits les plus abondantes sont de type EST, qui ne contiennent qu'une partie de la séquence de l'ARNm, et que les gènes bénéficiant dans les bases de données d'un ADNc de type "pleine taille" sont une minorité.

De plus, les logiciels basés uniquement sur les données de transcriptome ne fournissent que la structure exon-intron au sens biologique, c'est-à-dire qu'elles localisent les parties de la séquence génomiques contenues dans celle de l'ARNm. En revanche, la distinction précise entre la séquence codante et les UTR n'est pas réalisée, car l'identification de la CDS passe par la localisation du codon start, qui n'est pas évidente (Pedersen et Nielsen, 1997; Hatzigeorgiou, 2002; Liu *et al.*, 2004). Aucune garantie n'est faite que la prédiction respecte les contraintes classiques sur la structure des gènes, comme le fait de contenir une CDS complète avec un codon start et un codon stop dans la même phase de lecture.

Enfin, il arrive parfois que des transcrits chevauchants soient fusionnés pour construire une prédiction les incorporant. L'hypothèse sous-jacente est qu'ils proviennent du même variant d'épissage, ce qui peut très bien ne pas être le cas. Le risque est alors pris d'assembler deux transcrits provenant de variants d'épissage incompatibles, et de forcer la prédiction à suivre une structure chimérique virtuelle.

3/ La méthode développée dans EuGène

a. Philosophie

Puisque chaque approche comporte ses limitations propres, il est souhaitable de faire appel aux deux, ce qui est généralement le cas dans les pipe-lines des grands projets d'annotations (Haas *et al.*, 2003; Curwen *et al.*, 2004; Yuan *et al.*, 2003). Or, si elles sont réalisées de façon indépendante, les deux approches peuvent produire des prédictions totalement différentes, voire incompatibles, et qui nécessitent pour les fusionner proprement une expertise attentive de la part d'annotateurs (et donc des moyens humains), comme réalisé par Haas *et al.* (2003). Une approche fédératrice qui prend en compte toutes les informations en amont de l'étape de décision semble donc idéale. À notre connaissance, seul le logiciel GrailEXP (Xu et Uberbacher, 1997) semble aller dans cette direction. Cependant, il ne considère que les cas d'épissage alternatif de type exclusion/inclusion d'exon (Fig. 6.2 page 113), ignorant ainsi près de la moitié des cas (Xu *et al.*, 2002; Haas *et al.*, 2003). De plus, la méthode utilisée n'est pas publiée.

Pour étendre le domaine de la prédiction de gènes à la prédiction de gènes avec leurs variants d'épissage alternatif, notre objectif était de développer une méthode d'annotation satisfaisant les exigences suivantes :

- Pour une séquence génomique donnée, produire une structure de gènes optimale.
- En plus de cette prédiction, fournir pour chaque transcrit considéré

comme une preuve d'épissage alternatif la structure de gène optimale parmi celles qui sont en accord avec ce transcrit.
- Toute prédiction supplémentaire doit être justifiée par la séquence d'un transcrit.
- Un gène doit être identifiable même s'il ne bénéficie pas d'une couverture totale de transcrits.
- Toute prédiction produite doit respecter les contraintes classiques sur la structure des gènes (typiquement définie par un ou plusieurs exons séparés par des introns bordés par des sites d'épissage, formant une CDS complète sans codon stop en phase dans les exons).
- Et tout ceci de façon efficace, afin de pouvoir faire face en pratique à de longues séquences génomiques et de nombreux transcrits dans des temps raisonnables.

Pour atteindre ce but, nous avons choisi de combiner les avantages des approches *ab initio* et par similarité au sein d'un système intégratif, naturellement développé dans le logiciel EuGène, qui fournit un cadre propice à l'intégration d'informations.

La méthode passe par différentes étapes, détaillées ci-dessous, qui comprennent la recherche de transcrits révélant des cas d'épissage alternatif, l'extension du graphe d'EuGène pour prendre en compte ces transcrits, et la modification de l'algorithme de programmation dynamique pour produire autant de prédictions que de transcrits considérés.

b. Détection des transcrits *"alternatifs"*

La première étape du processus est l'identification des transcrits qui mettent en évidence des cas d'épissage alternatif. Le principe est de détecter les transcrits provenant de différents variants d'épissage sur la base de l'incompatibilité entre leur alignement génomique.

i) Recherche et alignement des transcrits

Tout d'abord, il faut récupérer les transcrits susceptibles d'être issus des gènes éventuellement présents dans la séquence génomique.

Alignement Puisque la séquence de tels transcrits doit être contenue dans celle que l'on souhaite annoter, on réalise pour cela une recherche dans les bases de données contenant des transcrits lesquels s'alignent avec la séquence génomique.

Par exemple, nous avons choisi dans la version testée la base de données dbEST (Boguski *et al.*, 1993), section *A. thaliana*, version du 12/2003 (190708 séquences). Comme logiciel d'alignement, nous avons utilisé Sim4 (Florea *et al.*, 1998) dans un premier temps, logiciel efficace pour traiter un grand nombre de séquences, et dans un second temps GeneSeqer (Usuka *et al.*, 2000), plus précis sur les frontières exon-intron, pour affiner les alignements de Sim4 ayant passé les étapes de filtrage suivantes.

FILTRAGE Puisque l'on recherche uniquement les transcrits issus des gènes éventuels de la séquence génomique, on réalise une première étape de filtrage sur la base de la qualité des séquences et des alignements. Tout transcrit ne remplissant pas les conditions de ce filtre n'est pas conservé pour la suite du processus. Pour *Arabidopsis thaliana*, les paramètres par défaut du filtre sont les suivants : longueur du transcrit entre 30 et 10000 pb, pourcentage minimum de la longueur du transcrit aligné avec le génomique = 95%, pourcentage minimum d'identité sur la partie alignée = 95%, longueur maximum des régions de brèches = 5000 pb, longueur maximum des régions d'appariements = 4000 pb. Par défaut, pour éviter d'éventuelles contaminations génomiques[7], les alignements "non épissés" (ne contenant pas de régions de brèches) sont écartés de l'analyse. De plus, puisque les séquences des transcrits contiennent fréquemment des erreurs dans les parties terminales, leurs extrémités sont raccourcies de 15 pb.

La deuxième étape de filtrage se base sur les comparaisons entre les alignements de transcrits, pour détecter des traces d'épissage alternatif. L'idée sous-jacente est que deux variants d'épissage (comme par exemple les couples présentées dans la figure 6.2 page 113) ne présentent pas des alignements génomique compatibles.

La procédure passe par l'analyse de toutes les paires d'alignements chevauchants, c'est-à-dire d'alignements qui impliquent une région génomique commune. On envisage deux types particuliers de relation pour un couple d'alignements chevauchants :

- Un alignement A est considéré comme *inclus* dans un alignement B si et seulement si pour chaque position génomique de A, B porte la même information (*gap* ou *match* pour les deux). Par défaut, tout alignement *inclus* dans un autre est immédiatement écarté de l'analyse, car il n'apporte aucune information supplémentaire sur une structure de transcrit par rapport à celui qui l'englobe.
- Deux alignements A et B sont tous deux considérés comme *incompatibles* si et seulement si il existe une position génomique à laquelle est associée une information *gap* dans l'un et *match* dans l'autre (incompatibilité entre exon et intron). On les considère comme des indices d'épissage alternatif.

Au terme des comparaisons deux à deux, puisque l'on s'intéresse aux épissages alternatifs, seuls les alignements de transcrits notés *incompatibles* sont conservés.

Remarque 18 *Du fait de leur faible proportion dans les alignements retenus, les mésappariements sont ignorés et considérés comme des appariements dans ces définitions ; on estime en effet qu'ils sont dûs à des erreurs ponctuelles de séquençage expérimental des transcrits.*

Un exemple de résultat de la procédure de filtrage est illustré dans la Fig. 6.3 page suivante.

[7] il arrive en effet que les bases de données de transcrits contiennent par erreur des séquences génomiques, parfois de grande taille.

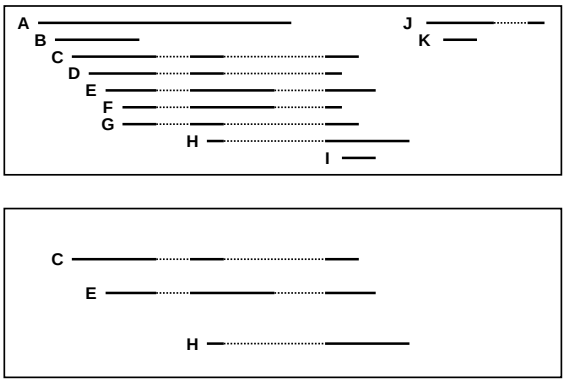

FIG. 6.3 – Exemple illustrant la procédure de filtrage d'un ensemble de transcrits alignés (rectangle du haut). Trait plein = appariements ; trait pointillé = brèches. A, B, I et K sont éliminés car ils ne sont pas épissés, D, F et G car ils sont *inclus* dans C ou E, et les autres car ils ne sont pas *incompatibles* avec un autre transcrit. Sont retenus C, E et H (rectangle du bas).

c. Intégration des alignements

Une fois que les transcrits indiquant des positions d'épissages alternatifs sont obtenus, il faut intégrer ces informations dans le graphe d'EUGÈNE. L'objectif est de produire pour chaque transcrit la meilleure prédiction parmi celles dont la structure exon-intron est compatible avec l'alignement correspondant.

i) Première approche

Une façon de procéder très simple serait de "lancer" EUGÈNE classiquement une première fois pour obtenir la prédiction optimale, puis de le relancer pour chaque transcrit en intégrant à chaque fois l'information spécifique portée par l'alignement correspondant.

En effet, pour un transcrit donné, il est facile d'obtenir la prédiction optimale parmi toutes celles qui fournissent une structure de gènes en accord avec son alignement. Il suffit d'injecter l'information fournie par l'alignement dans le graphe sous forme d'interdiction de pistes (pénalisation infinie), d'une façon comparable à ce qui est réalisé dans la première partie du chapitre pour les ADNc "pleine longueur". Puisque l'alignement d'un transcrit mature respecte la structure exon-intron du gène correspondant, il suffit d'interdire les pistes exoniques aux positions génomiques de brèches, et d'interdire les pistes introniques pour les positions d'appariements. De plus, si l'on considère le transcrit comme la preuve de la présence d'un gène, on peut interdire également la piste intergénique tout le long de l'alignement. La Fig. 6.4 ci-contre illustre la modification du graphe résultante d'une telle prise en compte. Ainsi, tout chemin de poids fini traversant le graphe ne viole pas les contraintes imposée par l'alignement. L'algorithme classique d'EUGÈNE qui permet d'identifier la prédiction la moins pénalisée fournit donc la structure génique à la fois optimale et compatible avec l'alignement de transcrit.

L'inconvénient de cette méthode est qu'elle demanderait de procéder ainsi pour chacun des transcrits, c'est-à-dire par une modification spécifique du graphe et par une exécution de l'algorithme d'EUGÈNE sur toute la longueur du graphe résultant. La complexité résultante en temps de calcul serait de

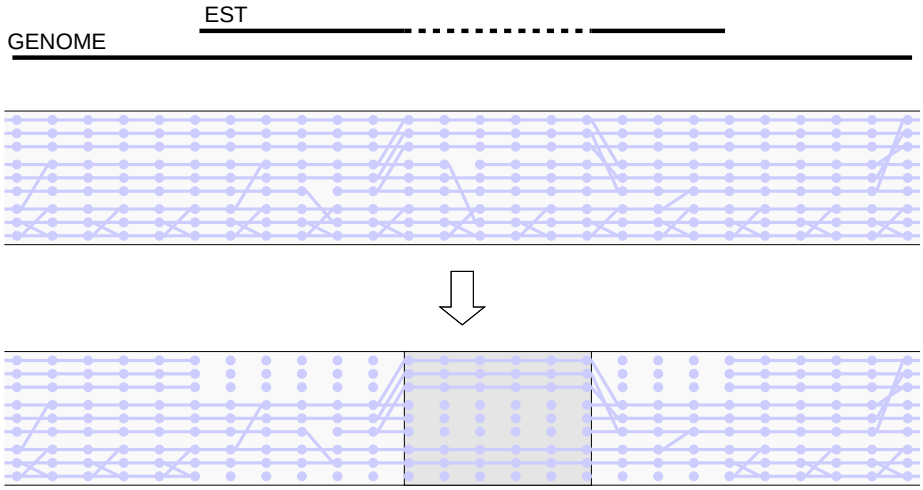

FIG. 6.4 – Exemple d'intégration possible de l'information apportée par l'alignement d'un transcrit (EST, trait plein = appariement, pointillé = brèche). Graphe du haut : avant intégration, graphe du bas : après intégration. Ne sont représentées que les pistes correspondant au brin sens, à savoir de haut en bas : introns (x3), exons (x3) UTR (x2) et intergénique. L'intégration entraîne une suppression d'arcs pour les pistes incohérentes avec la structure de l'alignement. La partie foncée représente la région de brèches.

$O(ln)$ si l est la longueur de la séquence et n le nombre de transcrits considérés. Or, sans aller chercher les 38000 variants de Dscam, il se peut que le nombre de transcrits soit élevé, une séquence pouvant de plus contenir plusieurs gènes. Dans le même temps, les séquences génomiques analysées font typiquement plusieurs dizaines ou centaines de milliers de paires de bases, et les transcrits typiquement quelques centaines. Cette première approche ne paraît donc pas envisageable en pratique.

Comment permettre alors une réduction de la complexité de résolution du problème ? La méthode suivante propose une réponse à cette question.

ii) L'extension du graphe

Le problème peut se formuler de cette façon : étant donné un espace de recherche (ici le graphe) et un ensemble de n modifications distinctes et locales de cet espace (les informations des transcrits), comment produire l'ensemble des n solutions globales correspondantes (une par modification) sans balayer n fois l'intégralité de l'espace de recherche ? L'idéal étant de conserver une complexité linéaire en temps/espace avec la quantité d'information.

PRINCIPE L'idée générale est de confiner les perturbations résultant des modifications locales dans des parties séparées de l'espace de recherche créées de façon *ad hoc*, afin de ne pas affecter le déroulement de l'algorithme de recherche global sur les parties communes. Plus précisément, il s'agit de dupliquer chaque partie du graphe localement impliquée dans un alignement de transcrits et de la relier par ses extrémités au graphe principal, à la fa-

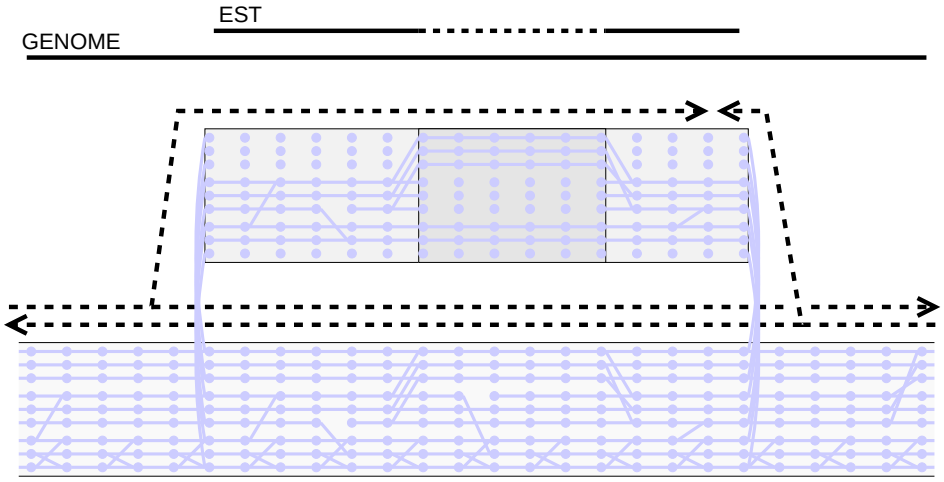

FIG. 6.5 – Extension du modèle de graphe d'EUGÈNE par une déviation locale intégrant l'information d'un alignement de transcrit. La région concernée par l'alignement est dupliquée (en haut) et connectée par ses extrémités au graphe principal (en bas). Seuls les arcs de la déviation peuvent faire l'objet de suppressions. Les flèches pointillées représentent les deux passes algorithmiques.

çon d'un branchement en parallèle. Des déviations locales sont ainsi créées. Puis, chaque déviation reçoit spécifiquement l'information apportée par l'alignement du transcrit correspondant, par le système d'interdiction de pistes vu précédemment. De cette façon, tout chemin passant par une déviation donnée produit une prédiction en accord avec l'alignement associé. Par une modification de l'algorithme d'EUGÈNE, il est possible d'identifier l'ensemble de ces chemins en 2 passes au lieu de n.

DUPLICATION ET BRANCHEMENT Pour un alignement de transcrit donné impliquant les positions génomiques entre e et f, la section entière (arcs et sommets) du graphe comprise entre e et f est dupliquée, comme illustré dans la Fig. 6.5. La copie résultante, formant un graphe auxiliaire, est ensuite connectée par ses extrémités au graphe principal par des arcs d'un nouveau type (s'ajoutant aux arcs de type *contenu* et aux arcs de type *signal*) : les arcs de types *déviation*, qui ne portent aucun poids (pondération nulle). Plus précisément, pour chaque piste j le sommet g_e^j du graphe principal est relié à sa copie du graphe auxiliaire (extrémité gauche) par un arc *déviation*. Il en va de même pour l'extrémité droite de la déviation, connectée aux sommets d_f^j du graphe principal. Tous les arcs de déviation sont orientés du graphe principal vers la déviation parallèle.

PONDÉRATION Initialement, les arcs composant les déviations (graphes auxiliaires de déviation) disposent des mêmes poids que ceux du graphe principal. Pour intégrer les informations fournies par les transcrits, les arcs de chaque déviation sont pondérés en fonction de l'alignement du transcrit correspondant. La procédure de pénalisation est la même que décrite pré-

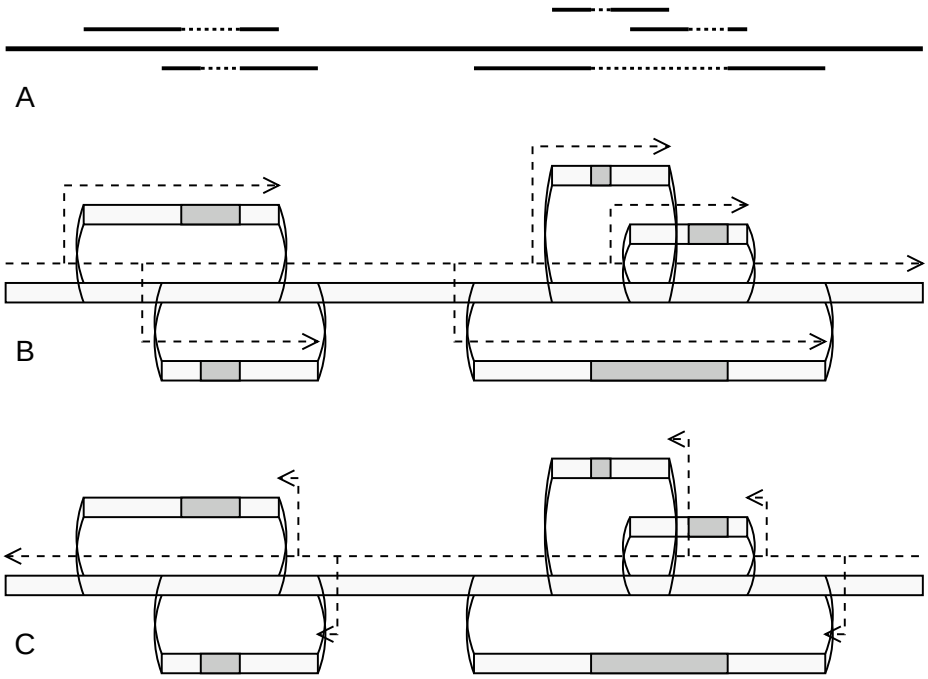

FIG. 6.6 – Illustration de la méthode avec plusieurs transcrits. A : représentation des alignements des transcrits (pointillés = brèches) sur la séquence génomique (long trait plein central). Ces alignements sont gardés car ils sont tous *incompatibles* avec au moins un autre. B : premier passage de l'algorithme (flèches pointillées) à travers le graphe résultant du branchement des déviations locales correspondantes. La formule de récurrence est appliquée dans les déviations. C : second passage de l'algorithme, de droite à gauche dans le graphe principal en connectant chaque déviation (voir texte).

cédemment pour la première approche simpliste de complexité trop élevée, c'est-à-dire par interdiction de pistes (Fig. 6.5 ci-contre). Ainsi, les chemins possibles (de poids non infini) qui traversent le graphe et qui passent par une déviation donnée caractérisent des structures de gènes toutes compatibles avec l'information contenue dans l'alignement du transcrit correspondant.

d. Modification de l'algorithme de programmation dynamique

Pour obtenir l'ensemble des prédictions optimales chacune pour un transcrit, il faut identifier pour chaque déviation le chemin le moins pénalisé qui la traverse. Ceci est réalisé par un algorithme de programmation dynamique bidirectionnel.

i) Premier passage

La même formule de récurrence que dans la version classique d'EUGÈNE — formule (3.3) page 69 — est appliquée dans l'ensemble du graphe pour toutes les pistes j et les positions génomiques i, de l'extrémité gauche G à l'extrémité droite D. Ainsi, pour chaque sommet d_i^j, on mémorise quel som-

met $d_{i-1}^{j'}$ de provenance possible permet de minimiser le poids du chemin partant de G arrivant à d_i^j. Ce poids W_i^j est également mémorisé, comme pour la version précédente. De même, quand l'algorithme parvient à l'extrémité droite D du graphe , la prédiction optimale est identifiée par la procédure de retour.

La différence caractérisant cette version est que l'algorithme est également appliqué dans les déviations parallèles du graphe jusqu'à leur extrémité droite, comme représenté dans la Fig. 6.6 B. Ce premier passage s'effectue avec une complexité linéaire en temps/espace avec le nombre de sommets d_i^j, déterminé par la taille du graphe total, et donc avec la longueur totale de séquences nucléiques (génome + transcrits).

ii) Second passage

Dans un second temps, tous les arcs changent de sens, exceptés les arcs de déviation. Puis, une formule de récurrence similaire est également appliquée à travers le graphe , mais dans le sens inverse, de l'extrémité droite D à l'extrémité gauche G et sans passer dans les déviations. Pour chaque position génomique i elle calcule de façon récurrente pour toutes les pistes j le poids V_i^j du chemin optimal partant de D et arrivant au sommet d_i^j (Fig. 6.6 C). Ce poids est donné par l'expression

$$\boxed{V_i^j = \min_{j'}(S_i^{j,j'} + C_{i+1}^{j'} + V_{i+1}^{j'})} \tag{6.1}$$

Le sommet $d_{i+1}^{j'}$ faisant partie de ce chemin optimal est mémorisé dans une variable τ_i^j pour l'étape de retour.

iii) Connexion des chemins

L'objectif est d'identifier pour chaque déviation le chemin optimal qui la traverse, ce qui est réalisé en connectant les chemins identifiés lors des deux passages précédents :

Pour une déviation δ donnée, si l'on considère les sommets r_f^j de son extrémité droite, le poids U du chemin optimal reliant G à D et passant par δ peut être calculé par

$$U(\delta) = \min_j(W_f^j + V_f^j)$$

À partir de la piste j ainsi identifiée, une procédure de retour appliquée de part et d'autre de d_f^j (en suivant d'un côté π_f^j et de l'autre τ_f^j) permet d'identifier le chemin optimal traversant la déviation, qui représente la meilleure prédiction possible en accord avec l'alignement du transcrit correspondant.

e. Résultats

i) Temps de calcul

L'étape des alignements de transcrits n'est pas considérée ici car elle n'est pas prise en charge par EuGène, et dépend donc des logiciels utilisés.

L'étape initiale d'analyse et de filtrage des transcrits pour ne conserver que ceux qui sont issus de phénomènes d'épissages alternatifs demande une complexité de $O(n^2)$ pour n transcrits initiaux. En effet, tous les couples doivent être considérés pour comparer les transcrits. Une simple comparaison est cependant rapide, car linéaire en nombre d'introns, typiquement borné par une constante (par exemple 50), d'où une complexité générale en $O(n^2)$.

L'étape correspondant aux deux passages de programmation dynamique demande une complexité linéaire en temps et en espace avec la quantité de données en entrée. Si L est la longueur totale de séquences nucléiques (séquence génomique + transcrits conservés), les poids de tous les chemins sont obtenus en $O(L)$

Pour l'étape de retour et de présentation des résultats, il n'est pas possible de garantir une linéarité avec la taille des données en entrée. En effet, une modification locale sur un graphe peut avoir une influence de longue portée sur la nature du chemin optimal, et donc rien ne garantit que la procédure de retour ne doive pas se réaliser sur l'intégralité du graphe pour chaque transcrit[8]. Il est cependant possible d'atteindre une complexité linéaire avec la taille de la sortie (les prédictions des variants) en n'affichant pour les prédictions alternatives que les parties qui diffèrent de la prédiction optimale, et en modifiant la procédure de retour pour lui éviter de poursuivre le parcours du graphe lorsqu'elle "rejoint" la prédiction optimale. Cette dernière modification ne figure pas dans la version courante du logiciel.

Pour donner un ordre de grandeur, une exécution d'EuGène avec un processeur AMD Athlon 1,7 GHz prend 47 secondes sur une séquence génomique de 500 kb, avec 945 transcrits conservés après l'étape de filtrage sur la qualité des alignements dont 24 *incompatibles*.

ii) Tests, vérifications et interprétations

Exemple spl7 Pour évaluer si la méthode représente une amélioration par rapport aux approches existantes, nous avons testé cette nouvelle version d'EuGène sur le gène spl7 d'*Arabidopsis thaliana* (pour *"squamosa promoter-binding protein-like 7"*). Ce gène contient 10 exons et présente deux variants d'ARNm connus identifié grâce à deux ADNc "pleine longueur" distincts (AY063815 et AF367355, voir Fig. 6.7 page suivante). Les alignements génomiques de ces ADNc permettent d'identifier de façon fiable les deux structure exon-intron que nous considérons comme annotation de référence, qui ne diffèrent que par l'extrémité 3' du 9^{e} exon (cas de site donneur alternatif, comme dans la Fig. 6.2 A). Cependant, sans ces deux ADNc complets, seuls le premier et les deux derniers exons du gènes sont "couverts" par des transcrits disponibles dans la base de données dbEST (Boguski *et al.*, 1993) comme le montre la Fig. 6.7. Cet exemple est intéressant car du fait de la couverture partielle, les approches exclusivement extrinsèques classiques ne peuvent produire une annotation correcte de ce gène.

[8] ceci semble toutefois peu probable en pratique. En effet, la propagation a plutôt tendance à se borner au voisinage de la région altérée, même si des études plus poussées seraient nécessaires pour évaluer ce genre de phénomène.

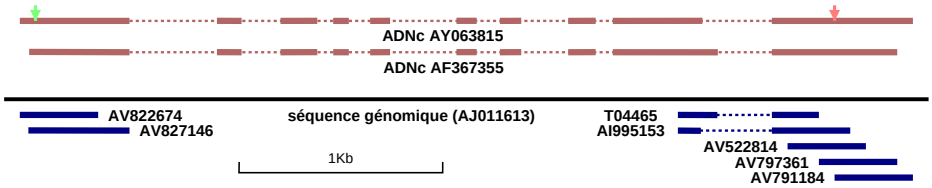

FIG. 6.7 – Alignements des transcrits (ADNc et EST) sur la séquence génomique contenant le gène spl7. Au dessus de la séquence génomique, les 2 ADNc alignés fournissent les deux structures de variants de référence (flèches = start et stop). Les EST T04465 et AI995153 présentent des alignements marqués *incompatibles*.

À partir des alignements génomiques des EST, nous avons donc "lancé" EUGÈNE sur la séquence génomique contenant le gène spl7. Puisque deux alignements d'EST (T04465 and AI995153) sont mutuellement *incompatibles*, EUGÈNE produit en plus de la prédiction optimale deux prédictions supplémentaires. L'une est identique à la prédiction optimale, qui identifie correctement l'un des deux variants de référence, et l'autre correspond à l'autre variant.

APPLICATION SUR ARASET Pour réaliser un test de plus grande envergure, nous avons appliqué cette méthode à l'intégralité du jeu de données ARASET (Pavy *et al.*, 1999a), qui a récemment été utilisé avec le même but pour l'évaluation du logiciel GENESEQER (Brendel *et al.*, 2004). EUGÈNE ayant déjà été évalué sur ce jeu, l'objectif ici est d'estimer une fréquence d'épissage alternatif pour un ensemble de gènes de référence d'*A. thaliana*. Sur les 168 gènes d'ARASET, 9 présentent au moins deux variants distincts, et furent l'objet d'une analyse approfondie. Les résultats sont présentés dans le tableau 6.2 ci-contre. Toutes ces prédictions semblent correspondre à de réels cas d'épissage alternatif sauf deux : l'une est dûe à une incompatibilité entre deux transcrits dont l'un correspond à un cas d'inclusion d'intron ; il est possible qu'il s'agisse d'un ARN partiellement épissé (il arrive en effet qu'au cours de l'extraction expérimentale d'ARNm cellulaires de tels intermédiaires du processus de maturation soient récupérés par erreur). L'autre est causé par deux transcrits provenant de gènes distincts en orientation opposée mais partiellement chevauchants concernant leur extrémité 3'. Cette configuration de gènes chevauchants, relativement rare et donc ignorée par les modèles des prédicteurs existants est ici traitée par EUGÈNE qui produit donc les deux prédictions correspondantes.

TAUX D'ÉPISSAGE ALTERNATIF Si l'on fait l'hypothèse que les 7 autres gènes produisent effectivement plusieurs variants, on peut estimer à partir de ce jeu de données un taux de gènes sujets à l'épissage alternatif voisin de $4,2\%$, un rapport du même ordre qu'estimé par ailleurs dans la littérature pour *A. thaliana*[9]. Bien que ce taux soit bien inférieur à celui mesuré

[9] de $1,5\%$ (Zhu *et al.*, 2003b) à $6,5\%$ (calculé à partir des données de Haas *et al.* (2003)).

sequence d'AraSet	n° ID du gène	n° ID de l'EST	type	notes
seq14	At2g47640	AI998209	ACC	réduction de 3 nt
seq16	At2g39780	AV832175	-EX	réduction de 36 nt
seq50	At5g46290	CF652136	-IN	sites d'épissage atypiques
seq53	At3g51800	AV544387	ACC	incorporation de 27 nt
seq62	At4g37070	AU236122	DON	incorporation de 33 nt
seq62	At4g37050	AV542276	FP	gènes chevauchants
seq65	At2g44100	BE521212	FP	épissage partiel probable
seq65	At2g44120	BE524396	ACC	réduction de 33 nt
seq69	At4g14350	AV547538	-IN	réduction de 105 nt

Tab. 6.2 – Analyse des cas d'épissage alternatif détectés par EuGène sur le jeu de données AraSet. n° ID = identifiant de la donnée. Types d'épissage alternatif : ACC = site donneur alternatif, -EX = exclusion d'exon, -IN = excision d'intron, FP = faux positif probable, nt = nucléotide. Des EST du gène At2g39780 (seq16) ne sont pas correctement alignés : Sim4 et GeneSeqer avec les options par défaut ratent un exon de 4 nt (qui n'est toutefois pas le responsable de l'épissage alternatif). La partie excisée de l'EST CF652136 pour le gène seq50 est flanquée par les nucléotides `GC-CT` au lieu de `GT-AG`. Dans seq62, l'EST AV542276 du gène At4g37040 chevauche un intron de l'EST AV562725 du gène voisin At4g37050. L'EST BE521212 de seq65 n'est pas épissé entre les exons 5 et 6 (cas de rétention d'intron ou probablement d'épissage incomplet).

actuellement chez *H. sapiens*[10], il est fort probable qu'il s'agisse d'une sous-estimation dûe à un manque de données : il a été démontré en effet que les gènes bénéficiant d'une abondante couverture en transcrits sont ceux sur lesquels on observe des cas d'épissage alternatif de façon prédominante, en raison de la probabilité de détecter des évènements rares dans des échantillons de grande taille (Kan *et al.*, 2002). La représentation en transcrits des gènes de l'arabette étant encore bien en deçà de celle des gènes humains, nous pensons que l'estimation du taux de gènes concernés par l'épissage alternatif chez *A. thaliana* ne manquera pas d'augmenter avec l'accumulation progressive des données de transcriptomes.

Comparaison avec les données de GeneSeqer Une comparaison des résultats obtenus sur AraSet avec ceux de la récente évaluation du logiciel GeneSeqer sur le même jeu est intéressante. En effet, seulement trois cas d'épissage alternatifs furent mentionnés par les auteurs. Cela s'explique par le fait que contrairement à nous, ils n'ont pas cherché à faire une analyse exhaustive (et n'ont considéré que les cas détectés par comparaison entre certaines des prédictions de leur logiciel avec les annotations de référence). Nous avons alors vérifié que nos alignements étaient identiques aux leurs, en analysant les données disponibles sur le site de l'AtGDB, *"Arabidopsis thaliana Genome Database"* (Zhu *et al.*, 2003b)[11] Un seul alignement

[10] de 30% à 65% (Modrek et Lee, 2002b; Kim *et al.*, 2004; Leipzig *et al.*, 2004).

[11] http ://www.plantgdb.org/AtGDB/prj/ZSB03PP/alternativeSplicing (*"Alternative splicing on AtGDB"*)

diffère entre les deux études : celui de l'EST CF652136, absent de la base AtGDB en raison probablement de sa date de dépôt dans la base dbEST (octobre 2003). Tous les autres alignements peuvent être visualisés sur le site de l'AtGDB. Enfin, nous avons cherché à savoir si les cas d'épissage alternatifs détectés par EuGène étaient déjà répertoriés dans les sections dédiées à l'épissage alternatif de deux bases de données de référence : la section AtGDB de la *"Plant Genome DataBase"* (Dong *et al.*, 2004) et la TIGRdb (Haas *et al.*, 2003)[12]. Seulement 3 de nos cas sont présents dans les deux bases (At2g47640, At3g51800 et At4g37070), et 3 ne figurent dans aucune (At2g39780, At5g46290 et At4g37050), ce qui confirme que l'application de notre méthode peut permettre de découvrir de façon automatique de nouveaux candidats d'épissage alternatif, même sur un jeu de données bien étudié.

III Conclusion

Dans ce chapitre consacré à l'intégration d'informations provenant d'études de transcriptomes, nous avons présenté deux applications distinctes du potentiel d'évolution du logiciel EuGène.

Nous avons pu dans un premier temps mettre en évidence la souplesse du modèle de graphe utilisé et la facilité d'incorporation d'un nouveau type d'information spécifique avec l'exemple des ADNc de type "pleine longueur". Il nous semble important de souligner que le besoin de cette nouvelle fonctionnalité fut suscité par la production et la disponibilité de données particulières issues d'un projet de séquençage massif de transcriptome. Cet exemple confirme la thèse que nous avons proposée dans le chapitre 2 selon laquelle le domaine de la prédiction de gène doit s'adapter aux avancées de celui de la génomique, et que la capacité d'intégration de nouvelles informations est un élément capital du potentiel d'évolution nécessaire.

Cette partie du travail de thèse fut notamment valorisée par une collaboration et de nombreuses interactions enrichissantes avec les utilisateurs du logiciel impliqués dans le projet CATMA (Crowe *et al.*, 2003).

L'intégration de la possibilité de l'épissage alternatif fut également amenée par les progrès de la génomique : c'est en effet à l'augmentation conjointe du nombre de transcrits disponibles et de l'estimation de sa fréquence que l'épissage alternatif doit la reconsidération de son importance.

L'exploitation de la flexibilité du logiciel EuGène, par l'extension de son modèle de gènes et de son algorithme de programmation dynamique, a permis de combiner les avantages des approches intrinsèques et des approches extrinsèques pour intégrer la détection d'épissages alternatifs dans le processus de prédiction de gènes (Foissac et Schiex, 2004). Il s'agit à notre

[12] http ://www.tigr.org/tdb/e2k1/ath1/altsplicing/splicing_variations.shtml (*"Arabidopsis splicing variations on TIGRdb"*)

connaissance de la première approche d'annotation automatique proposant plusieurs structures exon-intron par gène telles que chacune soit supportée par une séquence de transcrit, mais pas forcément sur toute la longueur. La méthode développée apporte une contribution originale et efficace à la résolution du problème crucial de l'intégration de l'épissage alternatif dans le domaine de la prédiction de gènes. Elle serait également capable éventuellement d'intégrer d'éventuelles prédictions *ab initio* de paires de sites d'épissage alternatif si elles existent un jour.

L'homme de Science le sait bien, lui, que, sans la Science, l'homme ne serait qu'un stupide animal sottement occupé à s'adonner aux vains plaisirs de l'amour dans les folles prairies de l'insouciance, alors que la Science, et la Science seule, a pu, patiemment, au fil des siècles, lui apporter l'horloge pointeuse et le parcmètre automatique sans lesquels il n'est pas de bonheur terrestre possible.

— Pierre Desproges

(Vivons heureux en attendant la mort, 1983)

Je sens que je vais conclure.

— Jean-Claude Dusse

Conclusion & perspectives

Ces dernières décennies ont vu la recherche scientifique bouleversée par l'explosion de la biologie moléculaire, puis de la génomique. L'exploitation des données produites par les projets internationaux de séquençage des génomes représente un enjeu colossal et les retombées potentielles qui en découlent concentrent de la part des chercheurs mais aussi du public une multitude d'attentes, d'espoirs, mais aussi d'inquiétudes. Une part majeure de cette tâche revient à la bioinformatique, concernant notamment l'étape d'annotation des séquences génomiques produites et la localisation des gènes qu'elles contiennent.

L'objectif du travail réalisé dans le cadre de cette thèse consistait à développer à appliquer de nouvelles méthodes d'intégration d'information pour améliorer les performances et le champ d'application des logiciels de détection de gènes.

Dans un premier temps, à travers une étude approfondie de l'état de l'art de la prédiction de structure génique, nous avons pu mettre en évidence les limitations des méthodes existantes et proposer un axe de travail pour apporter une contribution originale au domaine de recherche :

- L'analyse de l'évolution des informations utilisées par les méthodes existantes révèle que, du fait des progrès de la génomique, de nouvelles sources d'information apparaissent régulièrement. Bien que chaque type d'information permette une amélioration des performances des prédicteurs, l'intégration de l'ensemble des informations semble rester problématique.
- L'analyse des modèles et des algorithmes utilisés par les logiciels actuels confirme l'impression qu'ils sont difficiles à faire évoluer pour qu'ils s'adaptent rapidement aux progrès de la génomiques. Les raisons en sont d'une part la rigidité des modèles, qui ne favorise pas les évolutions, et d'autre part des méthodes d'intégration d'informations qui semblent difficilement généralisables.

La thèse qui se dégage de cette étude est qu'un logiciel de prédiction de gènes doit pour garantir une efficacité et une durée de vie satisfaisante disposer d'un modèle de structure génique souple et évolutif qui permette une intégration aisée de sources d'information diverses et d'une procédure

d'estimation de ses paramètres fiable et réutilisable à chaque tentative d'intégration d'information.

Certaines perspectives de recherche possibles apparaissent également à la suite de cette étude concernant certains aspects techniques, comme l'amélioration de la prise en compte des variations en $GC\%$ le long des génomes ou la construction de modèles représentant les régions intergéniques de façon appropriée.

Après avoir présenté le logiciel EuGène, qui a servi de base de développement pour les travaux accomplis, son modèle de graphe et les informations qu'il exploite, nous nous sommes intéressés à un processus clef pour l'intégration de nouveaux types d'information : l'estimation des paramètres de pondération par optimisation des performances du logiciel.

Ce processus d'optimisation représente une étape capitale pour le potentiel d'adaptation du logiciel :

- à de nouvelles intégrations d'information, qui passent par des pondérations d'arcs en fonction de paramètres spécifiques à estimer.
- à l'application du logiciel sur de nouveaux génomes, chaque espèce demandant des valeurs de paramètres spécifiques.
- à de nouvelles explorations méthodologiques.

Par une approche pragmatique originale dans le domaine combinant un algorithme génétique et une recherche linéaire, nous avons développé, testé et mis en application un processus d'estimation des paramètres du logiciel par optimisation de ses performances globales mesurées sur un jeu d'apprentissage. Ce travail a rendu possibles les développements qui ont suivi.

Là aussi, certaines voies de recherche sont envisageables. Il est clairement établi que l'influence des jeux de données de test sur l'évaluation des performances des logiciels est considérable, et c'est également le cas naturellement pour l'estimation des paramètres en fonction des jeux de données d'apprentissage. Il serait intéressant et probablement enrichissant pour la communauté de disposer d'une étude rigoureuse de l'influence des jeux de données sur les logiciels, afin de dégager les caractéristiques d'importance prépondérantes qui devraient guider la construction des jeux d'apprentissage. On peut tester par exemple l'influence du contenu en gènes des séquences composant les jeux (nombre, densité, taille, $GC\%$ ...), ou de leur organisation (introduction de gènes partiels dans les jeux, taille du contexte de part et d'autre des gènes ...).

Afin de tester par une mise en application ce potentiel d'évolution, nous avons cherché à améliorer les performances du logiciel en développant une version capable d'intégrer des informations d'homologies inter et intra-génomiques. Les résultats obtenus sur un ensemble de gènes de plusieurs espèces confirment l'efficacité du processus d'optimisation et l'intérêt d'une intégration massive d'informations de différentes natures. La méthode développée caractérise un outil d'annotation efficace, relativement générique et prêt à bénéficier des avancées de la génomique en termes de séquences génomiques produites.

Comme perspectives concernant cette partie, on peut imaginer intégrer de récents travaux de mise en évidence de propriétés statistiques permettant de distinguer les régions codantes du reste du génome (type *contenu*, basés sur la périodicité), qui, de la même façon que notre modèle protéique, semblent être peu spécifiques d'une espèce particulière (Kotlar et Lavner, 2003; Gao et Zhang, 2004).

Enfin, la dernière partie du travail, exploitant des données de transcriptomes, a mis à profit la souplesse du modèle pour proposer une approche novatrice au problème de l'épissage alternatif dans la prédiction de gènes. Bien que représentant à ce jour un défi capital pour les projets d'annotation, ce thème n'est traité de façon satisfaisante par aucun des logiciels existants. En intégrant au sein d'un même modèle une méthode de détection extrinsèque d'épissages alternatifs et le modèle *ab initio* de prédiction de structure génique, nous avons ouvert le champ d'application de la localisation de gènes à la localisation de gènes et variants, que nous estimons incoutournables dans un avenir proche.

Les perspectives concernant cette partie concernent tout d'abord l'intégration d'une récente amélioration de l'algorithme d'EuGène qui permet la prise en compte de distributions explicites de longueurs associées aux différentes annotations possibles. Cette extension ressemble dans l'esprit à l'apport des GHMM par rapport aux HMM (voir le chapitre 2) excepté qu'elle n'est pas restreinte aux exons pour garantir une complexité linéaire. Le problème est qu'elle n'est pas pour l'instant exploitable dans la version d'EuGène traitant l'épissage alternatif car elle brise la propriété de Bellman nécessaire notamment à l'étape de jonction des chemins optimaux. Des travaux complémentaires seraient nécessaires pour déterminer dans quelle mesure une extension de notre algorithme est possible sans détérioration de la complexité en temps de calcul.

D'autres travaux très récents peuvent être intégrés facilement dans cette version d'EuGène, concernant la prédiction de cas d'épissage alternatif de type exclusion/inclusion d'exons. Basée sur une anlayse comparative de gènes homologues, cette étude identifie certaines propriétés communes aux exons impliqués dans ce type d'épissage alternatif, et propose une méthode de classification applicable à tout exon (Sorek *et al.*, 2004). Intégrer ce type d'information dans EuGène semble relativement facile (par construction de déviations appropriées) et permettrait d'étendre encore son potentiel.

Autres développements

Toujours plus d'intégrations...

L'intégration massive d'informations n'a *a priori* aucune raison de s'arrêter en si bon chemin ; de nombreuses sources d'information feraient probablement le bonheur des utilisateurs d'EuGène, d'autant que très peu de prédicteurs peuvent prétendre à un tel potentiel d'intégration.

Les autres transcrits

Lors de la présentations des types de séquences transcrites en page 31, nous avons rapidement évoqués deux classes de molécules particulières, construites à partir d'ARNm :

Les SAGE (pour *"Serial Analysis of Gene Expression"*) ou *"tag"* SAGE sont des courts fragments générés par des procédés de biologie moléculaire à partir de transcrits cellulaires (Velculescu *et al.*, 1995). Sans décrire la méthode de construction, on peut dire que malgré sa courte taille (de l'ordre de la douzaine de pb — plusieurs méthodes existent) la séquence d'un *"tag"* SAGE est supposée être plus ou moins spécifique de la région génomique contenant le gène dont il est issu (en général la partie 3' terminale du gène). Ces séquences étant disponibles dans les bases de données, on peut imaginer intégrer cette information spécifique dans le graphe d'EuGène.

Les ORESTES (pour *"Open reading frame expressed sequence tags"*) sont des transcrits qui présentent eux aussi par construction une caractéristique particulière (Dias Neto *et al.*, 2000) ; ils correspondent généralement à une partie centrale de l'ARNm (l'idée étant de capturer la CDS dans cette séquence, alors que la majorité des EST ne couvrent que les UTR). Des projets d'analyse de transcriptome à grande échelle produisant ce genre de données (Camargo *et al.*, 2001), il peut être intéressant d'intégrer de façon spécifique l'information que représenterait un alignement entre un ORESTE et une région génomique (comme ce que nous avons réalisé pour les RIKEN en début de chapitre 6).

Du transcriptome au protéome

Un vaste domaine d'étude a fait son apparition depuis peu, et représente un enjeu d'avenir : il s'agit de la protéomique. La caractérisation de protéomes (ensemble de protéines) prend de l'ampleur avec des approches basée sur l'utilisation à grande échelle de la spectrométrie de masse (Aebersold et Mann, 2003).

Expérimentalement, par une digestion enzymatique ("découpage") des protéines contenues dans un ensemble donné suivie d'une analyse par spectrométrie qui détermine la masse des fragments ainsi obtenus, il est possible de caractérisation une "signature" globale pour un protéome donné (d'une façon comparable à une mesure de transcriptome par puce à ADN) sans avoir besoin d'en séquencer toutes les protéines. Les protéines étant par définition issues de régions codantes, la prédiction de gènes a probablement un intérêt dans ce nouveau champ d'investigation. Par exemple, s'il est facile de traduire virtuellement une séquence génomique donnée, il est également possible de chercher dans la séquence protéique virtuelle obtenue des sites possibles de coupure enzymatique connus (d'une façon comparable à la recherche d'un signal, grossièrement), de calculer le poids moléculaire des fragments virtuels, et de les comparer à ceux de fragments obtenus expérimentalement (Reguer *et al.*, 2003).

On peut discerner au sein des protéines des domaines fonctionnels, c'est-

à-dire des parties caractérisées par une fonction biochimique donnée (fixation d'un substrat, catalyse d'une réaction chimique...). De nombreuses bases de données publiques recensent les domaines protéiques connus et/ou possibles (Mulder *et al.*, 2003; Servant *et al.*, 2002). Ces informations peuvent être exploitées en recherchant la présence de domaines connus dans une séquence génomique donnée (traduite virtuellement), ce qui peut permettre d'identifier des régions à caractère codant.

Enfin, des projets ont pour objectif de caractériser non seulement les protéines, mais les interactions qu'elles peuvent avoir entre elles. On parle alors d'études d'interactome par exemple, ou de métabolome pour les protéines impliquées dans les voies métaboliques. Des procédés expérimentaux permettent d'identifier des paires de gènes dont on pense que les protéines interagissent, comme les expériences de double-hybride (Osman, 2004). Un intérêt pour la prédiction de gènes réside dans le fait que si dans un génome à annoter on localise un gène dont l'orthologue chez une espèce voisine est impliqué dans une interaction par paire, on peut s'aider de la séquence de l'autre protéine de la paire pour chercher si l'orthologue correspondant est présent dans le génome à annoter. Étendre ce raisonnement peut permettre d'envisager une intégration de réseaux entiers de gènes dans un processus d'annotation génomique de grande envergure.

Prédicteurs en réseaux

Lorsque l'on anticipe sur le nombre de génomes séquencés dans les années à venir, force est de constater que l'avenir des approches traitant de génomique comparative paraît assuré. Cependant, il y a fort à parier que les méthodes qui tireront leur épingle du lot sont celles qui permettront l'intégration massive d'informations provenant de plusieurs génomes simultanément. Une étape ultime d'EuGène pourrait constituer à long terme un système d'annotation en réseau contenant autant de graphe s qu'il y a de génomes à annoter, ou en cours d'annotation. L'objectif d'une multitude de connexions entre les graphe s (reflétant les homologies) serait que les avancées des connaissances sur un génome, un ensemble de protéines ou une source d'information quelconque permettent par le biais des connexions d'améliorer les annotations des autres génomes.

Ce système demande un protocole de "communication" entre les différents graphe , ou du moins de transmission d'informations. Or, il se trouve qu'une des sources d'information actuellement exploitable par EuGène se présente sous la forme d'un fichier d'annotation, pouvant décrire certaines structures de gènes. Ces informations peuvent être intégrées par EuGène, qui peut également produire ses prédictions de structure génique sous le même format. Ainsi on peut imaginer plusieurs graphe s interconnectés de façon dynamique se "communicant" leurs prédictions.

Un avantage considérable d'un tel projet fédérateur de grande envergure est qu'il proposerait un cadre d'intégration des efforts d'annotation de la communauté internationale. En effet, il est regrettable que certains génomes soient annotés plusieurs fois par différents groupes séparément, produisant

des résultats délicats à exploiter et de qualité inférieure à ce qu'aurait produit une collaboration. EuGène peut également à l'heure actuelle prendre en compte des informations arbitraires spécifiées par l'utilisateur (position et score d'un signal, d'une région codante,...). Étendre ce principe au système de graphe s en réseau permettrait à chacun de contribuer à l'effort collectif, comme cela est fait d'une certaine façon dans les bases de données de séquences biologiques utilisées au quotidien par l'ensemble de la communauté.

Il y a les inventeurs lumineux dont la gloire fracassante résonne longtemps après eux dans les plaines de la connaissance humaine, et puis il y a les inventeurs obscurs, les génies de l'ombre qui traversent la vie sans bruit et s'effacent à jamais sans que la moindre reconnaissance posthume vienne apaiser les tourments éternels de leur âme errante qui gémit aux vents mauvais de l'infernal séjour, sa désespérance écorchée aux griffes glacées d'ingratitude d'un monde au ventre mou sans chaleur ni tendresse.

— Pierre Desproges

(Les réquisitoires)

Bibliographie

Aebersold, R. et Mann, M. (2003). Mass spectrometry-based proteomics. *Nature*, **422**(6928), 198–207.

Alexandersson, M., Cawley, S., et Pachter, L. (2003). SLAM : cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res*, **13**(3), 496–502.

Allen, J. E., Pertea, M., et Salzberg, S. L. (2004). Computational gene prediction using multiple sources of evidence. *Genome Res*, **14**(1), 142–8.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., et Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, **215**(3), 403–10.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., et Lipman, D. J. (1997). Gapped blast and psi-blast : a new generation of protein database search programs. *Nucleic Acids Res*, **25**(17), 3389–402.

Asai, K., Itou, K., Ueno, Y., et Yada, T. (1998). Recognition of human genes by stochastic parsing. *Pac Symp Biocomput*, pages 228–39.

Ashburner, M. (2000). A biologist's view of the drosophila genome annotation assessment project. *Genome Res*, **10**(4), 391–3.

Azad, R. K. et Borodovsky, M. (2004). Effects of choice of dna sequence model structure on gene identification accuracy. *Bioinformatics*, **20**(7), 993–1005.

Bafna, V. et Huson, D. (2000). The conserved exon method for gene finding. In *Proc Int Conf Intell Syst Mol Biol.*, volume 8, pages 3–12.

Barker, W. C., Garavelli, J. S., Huang, H., McGarvey, P. B., Orcutt, B. C., Srinivasarao, G. Y., Xiao, C., Yeh, L. S., Ledley, R. S., Janda, J. F., Pfeiffer, F., Mewes, H. W., Tsugita, A., et Wu, C. (2000). The protein information resource (pir). *Nucleic Acids Res*, **28**(1), 41–4.

Batzoglou, S., Pachter, L., Mesirov, J. P., Berger, B., et Lander, E. S. (2000). Human and mouse gene structure : comparative analysis and application to exon prediction. *Genome Res*, **10**(7), 950–8.

Bejerano, G. (2004). Algorithms for variable length markov chain modeling. *Bioinformatics*, **20**(5), 788–9.

Bellman, R. (1957). *Dynamic Programming*. Princeton Univ. Press, Princeton, New Jersey.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., et Wheeler, D. L. (2004). Genbank : update. *Nucleic Acids Res*, **32 Database issue**, D23–6.

Birney, E. et Durbin, R. (1997). Dynamite : a flexible code generating language for dynamic programming methods used in sequence comparison. *Proc Int Conf Intell Syst Mol Biol*, **5**, 56–64.

Birney, E., Clamp, M., et Durbin, R. (2004). Genewise and genomewise. *Genome Res*, **14**(5), 988–95.

Black, D. L. (2003). Mechanisms of alternative pre-messenger rna splicing. *Annu Rev Biochem*, **72**, 291–336.

Blayo, P., Rouzé, P., et Sagot, M.-F. (2003). Orphan gene finding - an exon assembly approach. *Theoretical Computer Science*, **290**, 1407–1431.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., et Schneider, M. (2003). The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Res*, **31**(1), 365–70.

Boguski, M. S., Lowe, T. M., et Tolstoshev, C. M. (1993). dbEST-database for expressed sequence tags. *Nat Genet*, **4**(4), 332–3.

Boheler, K. R. et Stern, M. D. (2003). The new role of sage in gene discovery. *Trends Biotechnol*, **21**(2), 55–7 ; discussion 57–8.

Bonizzoni, P., Pesole, G., et Rizzi, R. (2003). A method to detect gene structure and alternative splice sites by agreeing ESTs to a genomic sequence. In G. Benson et R. Page, editors, *Algorithms in Bioinformatics, 3rd International Workshop (WABI)*, LNCS, pages 63–77. Springer Verlag,.

Borodovsky, M. et McIninch, J. (1993). GENMARK : Parallel gene recognition for both DNA strands. *Computers and Chemistry*, **17**(2), 123–33.

Borodovsky, M., Rudd, K. E., et Koonin, E. V. (1994). Intrinsic and extrinsic approaches for detecting genes in a bacterial genome. *Nucleic Acids Res*, **22**(22), 4756–67.

Bourguignon, P. et Robelin, D. (2004). Modèles de markov parcimonieux : sélection de modèle et estimation. In *JOBIM 2004*.

Brendel, V. et Kleffe, J. (1998). Prediction of locally optimal splice sites in plant pre-mRNA with application to gene identification in *Arabidopsis thaliana* genomic DNA. *Nucleic Acids Res.*, **26**, 4749–4757.

Brendel, V., Xing, L., et Zhu, W. (2004). Gene structure prediction from consensus spliced alignment of multiple ests matching the same genomic locus. *Bioinformatics*, **20**(7), 1157–69.

Brent, M. R. et Guigó, R. (2004). Recent advances in gene structure prediction. *Curr Opin Struct Biol*, **14**(3), 264–72.

Burge, C. (1997). *Identification of genes in human genomic DNA*. Ph.D. thesis, Stanford University.

Burge, C. et Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J Mol Biol.*, **268**(1), 78–94.

Burset, M. et Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.

Cai, Y. et Bork, P. (1998). Homology-based gene prediction using neural nets. *Anal Biochem*, **265**(2), 269–74.

Camargo, A. A., Samaia, H. P., Dias-Neto, E., Simao, D. F., Migotto, I. A., Briones, M. R., Costa, F. F., Nagai, M. A., Verjovski-Almeida, S., Zago, M. A., Andrade, L. E., Carrer, H., El-Dorry, H. F., Espreafico, E. M., Habr-Gama, A., Giannella-Neto, D., Goldman, G. H., Gruber, A., Hackel, C., Kimura, E. T., Maciel, R. M., Marie, S. K., Martins, E. A., Nobrega, M. P., Paco-Larson, M. L., Pardini, M. I., Pereira, G. G., Pesquero, J. B., Rodrigues, V., Rogatto, S. R., da Silva, I. D., Sogayar, M. C., Sonati, M. F., Tajara, E. H., Valentini, S. R., Alberto, F. L., Amaral, M. E., Aneas, I., Arnaldi, L. A., de Assis, A. M., Bengtson, M. H., Bergamo, N. A., Bombonato, V., de Camargo, M. E., Canevari, R. A., Carraro, D. M., Cerutti, J. M., Correa, M. L., Correa, R. F., Costa, M. C., Curcio, C., Hokama, P. O., Ferreira, A. J., Furuzawa, G. K., Gushiken, T., Ho, P. L., Kimura, E., Krieger, J. E., Leite, L. C., Majumder, P., Marins, M., Marques, E. R., Melo, A. S., Melo, M. B., Mestriner, C. A., Miracca, E. C., Miranda, D. C., Nascimento, A. L., Nobrega, F. G., Ojopi, E. P., Pandolfi, J. R., Pessoa, L. G., Prevedel, A. C., Rahal, P., Rainho, C. A., Reis, E. M., Ribeiro, M. L., da Ros, N., de Sa, R. G., Sales, M. M., Sant'anna, S. C., dos Santos, M. L., da Silva, A. M., da Silva, N. P., Silva, W. A. J., da Silveira, R. A., Sousa, J. F., Stecconi, D., Tsukumo, F., Valente, V., Soares, F., Moreira, E. S., Nunes, D. N., Correa, R. G., Zalcberg, H., Carvalho, A. F., Reis, L. F., Brentani, R. R., Simpson, A. J., de Souza, S. J., et Melo, M. (2001). The contribution of 700,000 orf sequence tags to the definition of the human transcriptome. *Proc Natl Acad Sci U S A*, **98**(21), 12103–8.

Carninci, P. et Hayashizaki, Y. (1999). High-efficiency full-length cdna cloning. *Methods Enzymol*, **303**, 19–44.

Carninci, P., Kvam, C., Kitamura, A., Ohsumi, T., Okazaki, Y., Itoh, M., Kamiya, M., Shibata, K., Sasaki, N., Izawa, M., Muramatsu, M., Hayashizaki, Y., et Schneider, C. (1996). High-efficiency full-length cdna cloning by biotinylated cap trapper. *Genomics*, **37**(3), 327–36.

Carninci, P., Westover, A., Nishiyama, Y., Ohsumi, T., Itoh, M., Nagaoka, S., Sasaki, N., Okazaki, Y., Muramatsu, M., Schneider, C., et Hayashizaki, Y. (1997). High efficiency selection of full-length cdna by improved biotinylated cap trapper. *DNA Res*, **4**(1), 61–6.

Cartegni, L., Wang, J., Zhu, Z., Zhang, M. Q., et Krainer, A. R. (2003). ESEfinder : A web resource to identify exonic splicing enhancers. *Nucleic Acids Res*, **31**(13), 3568–71.

Castelo, R. et Guigó, R. (2004). Splice site identification by idlbns. *Bioinformatics*, **20 Suppl 1**, I69–I76.

Cawley, S., Pachter, L., et Alexandersson, M. (2003). Slam web server for comparative gene finding and alignment. *Nucleic Acids Res*, **31**(13), 3507–9.

Cawley, S. L. et Pachter, L. (2003). HMM sampling and applications to gene finding and alternative splicing. *Bioinformatics*, **19**(Suppl 2), II36–II41.

Cerf, R. (1994). *Une Théorie Asymptotique des Algorithmes Génétiques*. Ph.D. thesis, Université Montpellier II (France).

Chen, T., Lu, C., et Li, W. (2004). Prediction of splice sites with dependency graphs and their expanded bayesian networks. *Bioinformatics*.

Chuang, J. et Roth, D. (2001). Gene recognition based on dag shortest paths. *Bioinformatics*, **17 Suppl 1**, S56–64.

Clark, F. et Thanaraj, T. A. (2002). Categorization and characterization of transcript-confirmed constitutively and alternatively spliced introns and exons from human. *Hum Mol Genet*, **11**(4), 451–64.

Claverie, J. M. (1997). Computational methods for the identification of genes in vertebrate genomic sequences. *Hum Mol Genet*, **6**(10), 1735–44.

Claverie, J. M. et Bougueleret, L. (1986). Heuristic informational analysis of sequences. *Nucleic Acids Res*, **14**(1), 179–96.

Cormen, T. H., Leiserson, C. E., et Rivest, R. L. (1990). *Introduction to algorithms*. MIT Press. ISBN : 0-262-03141-8.

Crowe, M. L., Serizet, C., Thareau, V., Aubourg, S., Rouze, P., Hilson, P., Beynon, J., Weisbeek, P., van Hummelen, P., Reymond, P., Paz-Ares, J., Nietfeld, W., et Trick, M. (2003). Catma : a complete arabidopsis gst database. *Nucleic Acids Res*, **31**(1), 156–8.

Curwen, V., Eyras, E., Andrews, T. D., Clarke, L., Mongin, E., Searle, S. M. J., et Clamp, M. (2004). The ensembl automatic gene annotation system. *Genome Res*, **14**(5), 942–50.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection : or the Preservation of Favoured Races in the Struggle for Life*. John Murray, London.

de la Mata, M., Alonso, C. R., Kadener, S., Fededa, J. P., Blaustein, M., Pelisch, F., Cramer, P., Bentley, D., et Kornblihtt, A. R. (2003). A slow rna polymerase ii affects alternative splicing in vivo. *Mol Cell*, **12**(2), 525–32.

Degroeve, S., De Baets, B., Van De Peer, Y., et Rouze, P. (2002). Feature subset selection for splice site prediction. *Bioinformatics*, **18 Suppl 2**, S75–S83.

Delcher, A. L., Harmon, D., Kasif, S., White, O., et Salzberg, S. L. (1999). Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.*, **27**(23), 4636–4641.

Dewey, C., Wu, J. Q., Cawley, S., Alexandersson, M., Gibbs, R., et Pachter, L. (2004). Accurate identification of novel human genes through simultaneous gene prediction in human, mouse, and rat. *Genome Res*, **14**(4), 661–4.

Dias Neto, E., Correa, R. G., Verjovski-Almeida, S., Briones, M. R., Nagai, M. A., da Silva, W. J., Zago, M. A., Bordin, S., Costa, F. F., Goldman, G. H., Carvalho, A. F., Matsukuma, A., Baia, G. S., Simpson, D. H., Brunstein, A., de Oliveira, P. S., Bucher, P., Jongeneel, C. V., O'Hare, M. J., Soares, F., Brentani, R. R., Reis, L. F., de Souza, S. J., et Simpson, A. J. (2000). Shotgun sequencing of the human transcriptome with orf expressed sequence tags. *Proc Natl Acad Sci U S A*, **97**(7), 3491–6.

Dong, Q., Schlueter, S. D., et Brendel, V. (2004). PlantGDB, plant genome database and analysis tools. *Nucleic Acids Res*, **32**, D354–9.

Dong, S. et Searls, D. B. (1994). Gene structure prediction by linguistic methods. *Genomics*, **23**(3), 540–51.

Durand, N. et Alliot, J. M. (1999). A combined nelder mead simplex and genetic algorithm. In *GECCO 99*.

Durbin, R., Eddy, S. R., Krogh, A., et Mitchison, G. (1999). *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Edery, I., Chu, L. L., Sonenberg, N., et Pelletier, J. (1995). An efficient strategy to isolate full-length cdnas based on an mrna cap retention procedure (capture). *Mol Cell Biol*, **15**(6), 3363–71.

Eyras, E., Caccamo, M., Curwen, V., et Clamp, M. (2004). Estgenes : alternative splicing from ests in ensembl. *Genome Res*, **14**(5), 976–87.

Fairbrother, W. G., Yeh, R.-F., Sharp, P. A., et Burge, C. B. (2002). Predictive identification of exonic splicing enhancers in human genes. *Science*, **297**(5583), 1007–13.

Fichant, G. et Gautier, C. (1987). Statistical method for predicting protein coding regions in nucleic acid sequences. *Comput Appl Biosci*, **3**(4), 287–95.

Fickett, J. W. (1982). Recognition of protein coding regions in dna sequences. *Nucleic Acids Res*, **10**(17), 5303–18.

Fickett, J. W. (1995). The gene identification problem : an overview for developers. *Computers Chem*, **20**, 103–118.

Fickett, J. W. et Tung, C. S. (1992). Assessment of protein coding measures. *Nucleic Acids Res*, **20**(24), 6441–50.

Fickett, J. W., Torney, D. C., et Wolf, D. R. (1992). Base compositional structure of genomes. *Genomics*, **13**(4), 1056–64.

Fields, C. A. et Soderlund, C. A. (1990). Gm : a practical tool for automating dna sequence analysis. *Comput Appl Biosci*, **6**(3), 263–70.

Flicek, P., Keibler, E., Hu, P., Korf, I., et Brent, M. R. (2003). Leveraging the mouse genome for gene prediction in human : from whole-genome shotgun reads to a global synteny map. *Genome Res*, **13**(1), 46–54.

Florea, L., Hartzell, G., Zhang, Z., Rubin, G., et Miller, W. (1998). A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res*, **8**(9), 967–974.

Foissac, S. et Schiex, T. (2004). Integrating anternative splicing detection into gene prediction. *BMC Bioinformatics*. À paraître.

Foissac, S., Bardou, P., Moisan, A., Cros, M.-J., et Schiex, T. (2003). Eugene'hom : A generic similarity-based gene finder using multiple homologous sequences. *Nucleic Acids Res*, **31**(13), 3742–5.

Gao, F. et Zhang, C.-T. (2004). Comparison of various algorithms for recognizing short coding sequences of human genes. *Bioinformatics*, **20**(5), 673–81.

Gelfand, M. S. (1990). Computer prediction of the exon-intron structure of mammalian pre-mrnas. *Nucleic Acids Res*, **18**(19), 5865–9.

Gelfand, M. S., Mironov, A. A., et Pevzner, P. A. (1996). Gene recognition via spliced sequence alignment. *Proc Natl Acad Sci U S A*, **93**(17), 9061–6.

Gelfand, M. S., Dubchak, I., Dralyuk, I., et Zorn, M. (1999). ASDB : database of alternatively spliced genes. *Nucleic Acids Res*, **27**(1), 301–2.

Gish, W. et States, D. J. (1993). Identification of protein coding regions by database similarity search. *Nat Genet*, **3**(3), 266–72.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimisation and machine learning*. Reading, MA : Addison-Wesley.

Gotoh, O. (2000). Homology-based gene structure prediction : simplified matching algorithm using a translated codon (tron) and improved accuracy by allowing for long gaps. *Bioinformatics*, **16**(3), 190–202.

Grantham, R., Gautier, C., Gouy, M., Mercier, R., et Pave, A. (1980). Codon catalog usage and the genome hypothesis. *Nucleic Acids Res*, **8**(1), r49–r62.

Gribskov, M., Devereux, J., et Burgess, R. R. (1984). The codon preference plot : graphic analysis of protein coding sequences and prediction of gene expression. *Nucleic Acids Res*, **12**(1 Pt 2), 539–49.

Guigó, R. (1998). Assembling genes from predicted exons in linear time with dynamic programming. *Journal of Computational Biology*, **5**(4), 681–702.

Guigó, R. (1999). Dna composition, codon usage and exon prediction. In M. J. Bishop, editor, *Genetic Databases*, pages 53–80. Academic Press, San Diego.

Guigó, R., Knudsen, S., Drake, N., et Smith, T. (1992). Prediction of gene structure. *J Mol Biol*, **226**(1), 141–57.

Guigó, R., Agarwal, P., Abril, J. F., Burset, M., et Fickett, J. W. (2000). An assessment of gene prediction accuracy in large dna sequences. *Genome Res*, **10**(10), 1631–42.

Gupta, S., Zink, D., Korn, B., Vingron, M., et Haas, S. (2004). Genome wide identification and classification of alternative splicing based on est data. *Bioinformatics*.

Haas, B., Delcher, A., Mount, S., Wortman, J., Smith, R. J., Hannick, L., Maiti, R., Ronning, C., Rusch, D., Town, C., Salzberg, S., et White, O. (2003). Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. *Nucleic Acids Res*, **31**(19), 5654–66.

Hardison, R. C., Oeltjen, J., et Miller, W. (1997). Long human-mouse sequence alignments reveal novel regulatory elements : a reason to sequence the mouse genome. *Genome Res*, **7**(10), 959–66.

Hatzigeorgiou, A. G. (2002). Translation initiation start prediction in human cdnas with high accuracy. *Bioinformatics*, **18**(2), 343–50.

Henderson, J., Salzberg, S., et Fasman, K. H. (1997). Finding genes in dna with a hidden markov model. *J Comput Biol*, **4**(2), 127–41.

Hogenesch, J. B., Ching, K. A., Batalov, S., Su, A. I., Walker, J. R., Zhou, Y., Kay, S. A., Schultz, P. G., et Cooke, M. P. (2001). A comparison of the celera and ensembl predicted gene sets reveals little overlap in novel genes. *Cell*, **106**(4), 413–5.

Holland, J. H. (1975). *Adaptation in Natural and Artifial Systems*. University of Michigan Press.

Hooper, P. M., Zhang, H., et Wishart, D. S. (2000). Prediction of genetic structure in eukaryotic dna using reference po int logistic regression and sequence alignment. *Bioinformatics*, **16**(5), 425–38.

Howe, K. L., Chothia, T., et Durbin, R. (2002). Gaze : a generic framework for the integration of gene-prediction data by dynamic programming. *Genome Res*, **12**(9), 1418–27.

Huang, H.-D., Horng, J.-T., Lee, C.-C., et Liu, B.-J. (2003). ProSplicer : a database of putative alternative splicing information derived from protein, mRNA and expressed sequence tag sequence data. *Genome Biol*, **4**(4), R29.

Huang, Y.-H., Chen, Y.-T., Lai, J.-J., Yang, S.-T., et Yang, U.-C. (2002). PALS db : Putative Alternative Splicing database. *Nucleic Acids Res*, **30**(1), 186–90.

Hutchinson, G. B. et Hayden, M. R. (1992). The prediction of exons through an analysis of spliceable open reading frames. *Nucleic Acids Res*, **20**(13), 3453–62.

Imanishi, T., Itoh, T., Suzuki, Y., O'Donovan, C., Fukuchi, S., Koyanagi, K. O., Barrero, R. A., Tamura, T., Yamaguchi-Kabata, Y., Tanino, M., Yura, K., Miyazaki, S., Ikeo, K., Homma, K., Kasprzyk, A., Nishikawa, T., Hirakawa, M., Thierry-Mieg, J., Thierry-Mieg, D., Ashurst, J., Jia, L., Nakao, M., Thomas, M. A., Mulder, N., Karavidopoulou, Y., Jin, L., Kim, S., Yasuda, T., Lenhard, B., Eveno, E., Suzuki, Y., Yamasaki, C., Takeda, J.-i., Gough, C., Hilton, P., Fujii, Y., Sakai, H., Tanaka, S., Amid, C., Bellgard, M., Bonaldo, M. d. F., Bono, H., Bromberg, S. K., Brookes, A. J., Bruford, E., Carninci, P., Chelala, C., Couillault, C., de Souza, S. J., Debily, M.-A., Devignes, M.-D., Dubchak, I., Endo, T., Estreicher, A., Eyras, E., Fukami-Kobayashi, K., Gopinath,

G. R., Graudens, E., Hahn, Y., Han, M., Han, Z.-G., Hanada, K., Hanaoka, H., Harada, E., Hashimoto, K., Hinz, U., Hirai, M., Hishiki, T., Hopkinson, I., Imbeaud, S., Inoko, H., Kanapin, A., Kaneko, Y., Kasukawa, T., Kelso, J., Kersey, P., Kikuno, R., Kimura, K., Korn, B., Kuryshev, V., Makalowska, I., Makino, T., Mano, S., Mariage-Samson, R., Mashima, J., Matsuda, H., Mewes, H.-W., Minoshima, S., Nagai, K., Nagasaki, H., Nagata, N., Nigam, R., Ogasawara, O., Ohara, O., Ohtsubo, M., Okada, N., Okido, T., Oota, S., Ota, M., Ota, T., Otsuki, T., Piatier-Tonneau, D., Poustka, A., Ren, S.-X., Saitou, N., Sakai, K., Sakamoto, S., Sakate, R., Schupp, I., Servant, F., Sherry, S., Shiba, R., Shimizu, N., Shimoyama, M., Simpson, A. J., Soares, B., Steward, C., Suwa, M., Suzuki, M., Takahashi, A., Tamiya, G., Tanaka, H., Taylor, T., Terwilliger, J. D., Unneberg, P., Veeramachaneni, V., Watanabe, S., Wilming, L., Yasuda, N., Yoo, H.-S., Stodolsky, M., Makalowski, W., Go, M., Nakai, K., Takagi, T., Kanehisa, M., Sakaki, Y., Quackenbush, J., Okazaki, Y., Hayashizaki, Y., Hide, W., Chakraborty, R., Nishikawa, K., Sugawara, H., Tateno, Y., Chen, Z., Oishi, M., Tonellato, P., Apweiler, R., Okubo, K., Wagner, L., Wiemann, S., Strausberg, R. L., Isogai, T., Auffray, C., Nomura, N., Gojobori, T., et Sugano, S. (2004). Integrative annotation of 21,037 human genes validated by full-length cdna clones. *PLoS Biol*, **2**(6), 856–75.

International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**(6822), 860–921.

Itoh, H., Washio, T., et Tomita, M. (2004). Computational comparative analyses of alternative splicing regulation using full-length cdna of various eukaryotes. *RNA*, **10**(7), 1005–18.

Jaillon, O., Aury, J.-M., Brunet, F., Petit, J.-L., Stange-Thomann, N., Mauceli, E., Bouneau, L., Fischer, C., Ozouf-Costaz, C., Bernot, A., Nicaud, S., Jaffe, D., Fisher, S., Lutfalla, G., Dossat, C., Segurens, B., Dasilva, C., Salanoubat, M., Levy, M., Boudet, N., Castellano, S., Anthouard, V., Jubin, C., Castelli, V., Katinka, M., Vacherie, B., Biemont, C., Skalli, Z., Cattolico, L., Poulain, J., De Berardinis, V., Cruaud, C., Duprat, S., Brottier, P., Coutanceau, J.-P., Gouzy, J., Parra, G., Lardier, G., Chapple, C., McKernan, K. J., McEwan, P., Bosak, S., Kellis, M., Volff, J.-N., Guigo, R., Zody, M. C., Mesirov, J., Lindblad-Toh, K., Birren, B., Nusbaum, C., Kahn, D., Robinson-Rechavi, M., Laudet, V., Schachter, V., Quetier, F., Saurin, W., Scarpelli, C., Wincker, P., Lander, E. S., Weissenbach, J., et Roest Crollius, H. (2004). Genome duplication in the teleost fish tetraodon nigroviridis reveals the early vertebrate proto-karyotype. *Nature*, **431**(7011), 946–57.

Jareborg, N., Birney, E., et Durbin, R. (1999). Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Res*, **9**(9), 815–24.

Johnson, J., Castle, J., Garrett-Engele, P., Kan, Z., Loerch, P., Armour, C., Santos, R., Schadt, E., Stoughton, R., et Shoemaker, D. (2003). Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. *Science*, **302**(5653), 2141–4.

Kan, Z., Rouchka, E., Gish, W., et States, D. (2001). Gene structure prediction and alternative splicing analysis using genomically aligned ESTs. *Genome Res*, **11**(5), 889–900.

Kan, Z., States, D., et Gish, W. (2002). Selecting for functional alternative splices in ESTs. *Genome Res*, **12**(12), 1837–45.

Kato, S., Sekine, S., Oh, S. W., Kim, N. S., Umezawa, Y., Abe, N., Yokoyama-Kobayashi, M., et Aoki, T. (1994). Construction of a human full-length cdna bank. *Gene*, **150**(2), 243–50.

Keibler, E. et Brent, M. R. (2003). Eval : a software package for analysis of genome annotations. *BMC Bioinformatics*, **4**(1), 50.

Kent, W. J. et Zahler, A. M. (2000). Conservation, regulation, synteny, and introns in a large-scale c. briggsae-c. elegans genomic alignment. *Genome Res*, **10**(8), 1115–25.

Kim, H., Klein, R., Majewski, J., et Ott, J. (2004). Estimating rates of alternative splicing in mammals and invertebrates. *Nat Genet*, **36**(9), 915–6 ; author reply 916–7.

Kleffe, J., Hermann, K., Vahrson, W., Wittig, B., et Brendel, V. (1998). Genegenerator–a flexible algorithm for gene prediction and its application to maize sequences. *Bioinformatics*, **14**(3), 232–43.

Korf, I. (2004). Gene finding in novel genomes. *BMC Bioinformatics*, **5**(1), 59.

Korf, I., Flicek, P., Duan, D., et Brent, M. (2001). Integrating genomic homology into gene structure prediction. *Bioinformatics*, **17**(Suppl 1), S140–8.

Korning, P., Hebsgaard, S., Rouzé, P., et Brunak, S. (1996). Cleaning the genbank arabidopsis thaliana data set. *Nucleic Acids Res.*, **24**, 316–320.

Kotlar, D. et Lavner, Y. (2003). Gene prediction by spectral rotation measure : a new method for ident ifying protein-coding regions. *Genome Res*, **13**(8), 1930–7.

Kraemer, E., Wang, J., Guo, J., Hopkins, S., et Arnold, J. (2001). An analysis of gene-finding programs for neurospora crassa. *Bioinformatics*, **17**(10), 901–12.

Krause, A., Haas, S. A., Coward, E., et Vingron, M. (2002). Systers, genenest, splicenest : exploring sequence space from genome to protein. *Nucleic Acids Res*, **30**(1), 299–300.

Krogh, A. (1997). Two methods for improving performance of an hmm and their application for gene finding. *Proc Int Conf Intell Syst Mol Biol*, **5**, 179–86.

Krogh, A. (2000). Using database matches with for HMMGene for automated gene detection in drosophila. *Genome Res.*, **10**(4), 391–7.

Krogh, A., Mian, I. S., et Haussler, D. (1994). A hidden markov model that finds genes in e. coli dna. *Nucleic Acids Res*, **22**(22), 4768–78.

Kulp, D., Haussler, D., Reese, M. G., et Eeckman, F. H. (1996). A generalized hidden markov model for the recognition of human genes in dna. *Proc Int Conf Intell Syst Mol Biol*, **4**, 134–42.

Kulp, D., Haussler, D., Reese, M. G., et Eeckman, F. H. (1997). Integrating database homology in a probabilistic gene structure model. *Pac Symp Biocomput*, pages 232–44.

Lee, C., Atanelov, L., Modrek, B., et Xing, Y. (2003). ASAP : the alternative splicing annotation project. *Nucleic Acids Res*, **31**(1), 101–5.

Leipzig, J., Pevzner, P., et Heber, S. (2004). The alternative splicing gallery (asg) : bridging the gap between genome and transcriptome. *Nucleic Acids Res*, **32**(13), 3977–83.

Liu, H., Han, H., Li, J., et Wong, L. (2004). Dnafsminer : a web-based software toolbox to recognize two types of functional sites in dna sequences. *Bioinformatics*.

Lorkovic, Z. J., Wieczorek Kirk, D. A., Lambermon, M. H., et Filipowicz, W. (2000). Pre-mrna splicing in higher plants. *Trends Plant Sci*, **5**(4), 160–7.

Majoros, W. H., Pertea, M., Antonescu, C., et Salzberg, S. L. (2003). Glimmerm, exonomy and unveil : three ab initio eukaryotic genefinders. *Nucleic Acids Res*, **31**(13), 3601–4.

Maniatis, T. et Tasic, B. (2002). Alternative pre-mrna splicing and proteome expansion in metazoans. *Nature*, **418**(6894), 236–43.

Maruyama, K. et Sugano, S. (1994). Oligo-capping : a simple method to replace the cap structure of eukaryotic mrnas with oligoribonucleotides. *Gene*, **138**(1-2), 171–4.

Mathé, C., Sagot, M.-F., Schiex, T., et Rouzé, P. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.*, **30**(19), 4103–4117.

McCaldon, P. et Argos, P. (1988). Oligopeptide biases in protein sequences and their use in predicting protein coding regions in nucleotide sequences. *Proteins*, **4**(2), 99–122.

Meyer, I. M. et Durbin, R. (2002). Comparative ab initio prediction of gene structures using pair hmms. *Bioinformatics*, **18**(10), 1309–18.

Meyer, I. M. et Durbin, R. (2004). Gene structure conservation aids similarity based gene prediction. *Nucleic Acids Res*, **32**(2), 776–83.

Miriami, E., Sperling, R., Sperling, J., et Motro, U. (2004). Regulation of splicing : the importance of being translatable. *RNA*, **10**(1), 1–4.

Modrek, B. et Lee, C. (2002a). A genomic view of alternative splicing. *Nat Genet*, **30**(1), 13–9.

Modrek, B. et Lee, C. (2002b). A genomic view of alternative splicing. *Nat Genet*, **30**(1), 13–9.

Modrek, B., Resch, A., Grasso, C., et Lee, C. (2001). Genome-wide detection of alternative splicing in expressed sequences of human genes. *Nucleic Acids Res*, **29**(13), 2850–9.

Morgenstern, B., Rinner, O., Abdeddaim, S., Haase, D., Mayer, K. F. X., Dress, A. W. M., et Mewes, H.-W. (2002). Exon discovery by genomic sequence alignment. *Bioinformatics*, **18**(6), 777–87.

Mouse Genome Sequencing Consortium (2002). Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**(6915), 520–62.

Mulder, N. J., Apweiler, R., Attwood, T. K., Bairoch, A., Barrell, D., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P., Copley, R. R., Courcelle, E., Das, U., Durbin, R., Falquet, L., Fleischmann, W., Griffiths-Jones, S., Haft, D., Harte, N., Hulo, N., Kahn, D., Kanapin, A., Krestyaninova, M., Lopez, R., Letunic, I., Lonsdale, D., Silventoinen, V., Orchard, S. E., Pagni, M., Peyruc, D., Ponting, C. P., Selengut, J. D., Servant, F., Sigrist, C. J. A., Vaughan, R., et Zdobnov, E. M. (2003). The interpro database, 2003 brings increased coverage and new features. *Nucleic Acids Res*, **31**(1), 315–8.

Murakami, K. et Takagi, T. (1998). Gene recognition by combination of several gene-finding programs. *Bioinformatics*, **14**(8), 665–75.

Needleman, S. et Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**(3), 443–453.

Nelder, J. A. et Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, **7**(4), 308–313.

Nicolas, P., Bize, L., Muri, F., Hoebeke, M., Rodolphe, F., Ehrlich, S. D., Prum, B., et Bessieres, P. (2002). Mining bacillus subtilis chromosome heterogeneities using hidden markov models. *Nucleic Acids Res*, **30**(6), 1418–26.

Novichkov, P., Gelfand, M., et Mironov, A. (2001). Gene recognition in eukaryotic DNA by comparison of genomic sequences. *Bioinformatics*, **17**(11), 1011–8.

Okazaki, Y., Furuno, M., Kasukawa, T., Adachi, J., Bono, H., Kondo, S., Nikaido, I., Osato, N., Saito, R., Suzuki, H., Yamanaka, I., Kiyosawa, H., Yagi, K., Tomaru, Y., Hasegawa, Y., Nogami, A., Schonbach, C., Gojobori, T., Baldarelli, R., Hill, D. P., Bult, C., Hume, D. A., Quackenbush, J., Schriml, L. M., Kanapin, A., Matsuda, H., Batalov, S., Beisel, K. W., Blake, J. A., Bradt, D., Brusic, V., Chothia, C., Corbani, L. E., Cousins, S., Dalla, E., Dragani, T. A., Fletcher, C. F., Forrest, A., Frazer, K. S., Gaasterland, T., Gariboldi, M., Gissi, C., Godzik, A., Gough, J., Grimmond, S., Gustincich, S., Hirokawa, N., Jackson, I. J., Jarvis, E. D., Kanai, A., Kawaji, H., Kawasawa, Y., Kedzierski, R. M., King, B. L., Konagaya, A., Kurochkin, I. V., Lee, Y., Lenhard, B., Lyons, P. A., Maglott, D. R., Maltais, L., Marchionni, L., McKenzie, L., Miki, H., Nagashima, T., Numata, K., Okido, T., Pavan, W. J., Pertea, G., Pesole, G., Petrovsky, N., Pillai, R., Pontius, J. U., Qi, D., Ramachandran, S., Ravasi, T., Reed, J. C., Reed, D. J., Reid, J., Ring, B. Z., Ringwald, M., Sandelin, A., Schneider, C., Semple, C. A. M., Setou, M., Shimada, K., Sultana, R., Takenaka, Y., Taylor, M. S., Teasdale, R. D., Tomita, M., Verardo, R., Wagner, L., Wahlestedt, C., Wang, Y., Watanabe, Y., Wells, C., Wilming, L. G., Wynshaw-Boris, A., Yanagisawa, M., Yang, I., Yang, L., Yuan, Z., Zavolan, M., Zhu, Y., Zimmer, A., Carninci, P., Hayatsu, N., Hirozane-Kishikawa, T., Konno, H., Nakamura, M., Sakazume, N., Sato, K., Shiraki, T., Waki, K., Kawai, J., Aizawa, K., Arakawa, T., Fukuda, S., Hara, A., Hashizume, W., Imotani, K., Ishii, Y., Itoh, M., Kagawa, I., Miyazaki, A., Sakai, K., Sasaki, D., Shibata, K., Shinagawa, A., Yasunishi, A., Yoshino, M., Waterston, R., Lander, E. S., Rogers, J., Birney, E., et Hayashizaki, Y. (2002). Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cdnas. *Nature*, **420**(6915), 563–73.

Osman, A. (2004). Yeast two-hybrid assay for studying protein-protein interactions. *Methods Mol Biol*, **270**, 403–22.

Pachter, L., Alexandersson, M., et Cawley, S. (2002). Applications of generalized pair hidden markov models to alignment and gene finding problems. *J Comput Biol*, **9**(2), 389–99.

Parra, G., Blanco, E., et Guigó, R. (2000). Geneid in drosophila. *Genome Res.*, **10**(4), 391–7.

Parra, G., Agarwal, P., Abril, J., Wiehe, T., Fickett, J., et Guigó, R. (2003). Comparative gene prediction in human and mouse. *Genome Res*, **13**(1), 108–17.

Pavlovic, V., Garg, A., et Kasif, S. (2002). A bayesian framework for combining gene predictions. *Bioinformatics*, **18**(1), 19–27.

Pavy, N., Rombauts, S., Déhais, P., Mathé, C., Ramana, D., Leroy, P., et Rouzé, P. (1999a). Evaluation of gene prediction software using a genomic data set : application to *Arabidopsis thaliana* sequences. *Bioinformatics*, **15**(11), 887–99.

Pavy, N., Rombauts, S., Déhais, P., Mathé, C., Ramana, D., Leroy, P., et Rouzé, P. (1999b). Evaluation of gene prediction software using a genomic dataset : application to *Arabidopsis thaliana* sequences. In *Proc. of 2^d Georgia Tech conference on BioInformatics*, Atlanta.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, San Mateo.

Pedersen, A. et Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes : prespectives for EST and genome analysis. In *Proc. of ISMB'97*, pages 226–233. AAAI Press.

Pedersen, J. S. et Hein, J. (2003). Gene finding with a hidden markov model of genome structure and evolution. *Bioinformatics*, **19**(2), 219–27.

Pospisil, H., Herrmann, A., Bortfeldt, R. H., et Reich, J. G. (2004). EASED : Extended alternatively spliced EST database. *Nucleic Acids Res*, **32**, D70–4.

Rabiner, L. (1989). A tutorial on hidden markov models and selected application in speech recognition. *Proc. IEEE*, **77**, 257–286.

Raftery, A. et Berchtold, A. (2002). The mixture transition distribution model for high-order markov chains and non-gaussian time series. *Statistical Science*, (17), 328–356.

Reese, M., Kulp, D., Tammana, H., et Haussler, D. (2000a). Genie–gene finding in drosophila melanogaster. *Genome Res.*, **10**(4), 529–38.

Reese, M. G., Eeckman, F. H., Kulp, D., et Haussler, D. (1997). Improved splice site detection in genie. *J Comput Biol*, **4**(3), 311–23.

Reese, M. G., Hartzell, G., Harris, N. L., Ohler, U., Abril, J. F., et Lewis, S. E. (2000b). Genome annotation assessment in drosophila melanogaster. *Genome Res*, **10**(4), 483–501.

Reguer, E., Nugues, E., Cahuzac, R., Ferro, M., Vermat, T., Mouton, E., et Garin, J. (2003). A software pipeline dedicated to automatic ms/ms data analysis. In *Proc. ECCB 2003 (European Conference on Computational Biology)*, Paris.

Rinner, O. et Morgenstern, B. (2002). Agenda : gene prediction by comparative sequence analysis. *In Silico Biol*, **2**(3), 195–205.

Roca, X., Sachidanandam, R., et Krainer, A. R. (2003). Intrinsic differences between authentic and cryptic 5' splice sites. *Nucleic Acids Res*, **31**(21), 6321–33.

Rogic, S., Mackworth, A. K., et Ouellette, F. B. (2001). Evaluation of gene-finding programs on mammalian sequences. *Genome Res*, **11**(5), 817–32.

Rogic, S., Ouellette, B. F. F., et Mackworth, A. K. (2002). Improving gene recognition accuracy by combining predictions from two gene-finding programs. *Bioinformatics*, **18**(8), 1034–45.

Saeys, Y., Degroeve, S., Aeyels, D., Rouze, P., et Van de Peer, Y. (2004). Feature selection for splice site prediction : a new method using eda-based feature ranking. *BMC Bioinformatics*, **5**(1), 64.

Salamov, A. A. et Solovyev, V. V. (2000). Ab initio gene finding in drosophila genomic dna. *Genome Res*, **10**(4), 516–22.

Salzberg, S. (1997). A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput Appl Biosci.*, **13**(4), 365–76.

Salzberg, S., Delcher, A. L., Fasman, K. H., et Henderson, J. (1998a). A decision tree system for finding genes in dna. *J Comput Biol*, **5**(4), 667–80.

Salzberg, S. L., Delcher, A. L., Kasif, S., et White, O. (1998b). Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, **26**, 544–548.

Salzberg, S. L., Pertea, M., Delcher, A. L., Gardner, M. J., et Tettelin, H. (1999). Interpolated markov models for eukaryotic gene finding. *Genomics*, **59**(1), 24–31.

Samson, D., Legeai, F., Karsenty, E., Reboux, S., Veyrieras, J.-B., Just, J., et Barillot, E. (2003). Genoplante-info (gpi) : a collection of databases and bioinformatics resources for plant genomics. *Nucleic Acids Res*, **31**(1), 179–82.

Schiex, T., Moisan, A., Duret, L., et Rouzé, P. (1999). EuGène : A simple yet effective gene finder for eucaryotic organisms (*Arabidopsis thaliana*). In *Proc. of the 2^nd Georgia Tech International Conference on Bioinformatics - In silico Biology*, Atlanta.

Schiex, T., Moisan, A., et Rouzé, P. (2001). EuGène, an eukaryotic gene finder that combines several type of evidence. In *Computational Biology, selected papers from JOBIM'2000*, number 2066 in LNCS, pages 118–133. Springer Verlag,.

Schmucker, D., Clemens, J. C., Shu, H., Worby, C. A., Xiao, J., Muda, M., Dixon, J. E., et Zipursky, S. L. (2000). Drosophila dscam is an axon guidance receptor exhibiting extraordinary molecular diversity. *Cell*, **101**(6), 671–84.

Servant, F., Bru, C., Carrere, S., Courcelle, E., Gouzy, J., Peyruc, D., et Kahn, D. (2002). Prodom : automated clustering of homologous domains. *Brief Bioinform*, **3**(3), 246–51.

Siepel, A. et Haussler, D. (2004). Combining phylogenetic and hidden markov models in biosequence analysis. *J Comput Biol*, **11**(2-3), 413–28.

Singh, R. (2002). Rna-protein interactions that regulate pre-mrna splicing. *Gene Expr*, **10**(1-2), 79–92.

Snyder, E. E. et Stormo, G. D. (1993). Identification of coding regions in genomic dna sequences : an application of dynamic programming and neural networks. *Nucleic Acids Res*, **21**(3), 607–13.

Snyder, E. E. et Stormo, G. D. (1995). Identification of protein coding regions in genomic dna. *J Mol Biol*, **248**(1), 1–18.

Solovyev, V. et Salamov, A. (1997). The Gene-Finder computer tools for analysis of human and model organisms genome sequences. *Proc Int Conf Intell Syst Mol Biol*, **5**, 294–302.

Solovyev, V. V., Salamov, A. A., et Lawrence, C. B. (1994). Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Res*, **22**(24), 5156–63.

Sorek, R., Shemesh, R., Cohen, Y., Basechess, O., Ast, G., et Shamir, R. (2004). A non-est-based method for exon-skipping prediction. *Genome Res*, **14**(8), 1617–23.

Staden, R. (1984a). Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res*, **12**(1 Pt 2), 505–19.

Staden, R. (1984b). Measurements of the effects that coding for a protein has on a dna sequence and their use for finding genes. *Nucleic Acids Res*, **12**(1 Pt 2), 551–67.

Staden, R. et McLachlan, A. D. (1982). Codon preference and its use in identifying protein coding regions in long dna sequences. *Nucleic Acids Res*, **10**(1), 141–56.

Stamm, S., Zhu, J., Nakai, K., Stoilov, P., Stoss, O., et Zhang, M. Q. (2000). An alternative-exon database and its statistical analysis. *DNA Cell Biol*, **19**(12), 739–56.

Stanke, M. et Waack, S. (2003). Gene prediction with a hidden markov model and a new intron submodel. *Bioinformatics*, **19 Suppl 2**, II215–II225.

Stormo, G. D. et Haussler, D. (1994). Optimally parsing a sequence into different classes based on multiple types of evidence. *Proc Int Conf Intell Syst Mol Biol*, **2**, 369–75.

Taher, L., Rinner, O., Garg, S., Sczyrba, A., Brudno, M., Batzoglou, S., et Morgenstern, B. (2003). Agenda : homology-based gene prediction. *Bioinformatics*, **19**(12), 1575–7.

Taher, L., Rinner, O., Garg, S., Sczyrba, A., et Morgenstern, B. (2004a). Agenda : gene prediction by cross-species sequence comparison. *Nucleic Acids Res*, **32**(Web Server issue), W305–8.

Taher, L., Rinner, O., Garg, S., Sczyrba, A., et Morgenstern, B. (2004b). Agenda : gene prediction by cross-species sequence comparison. *Nucleic Acids Res*, **32**(Web Server issue), W305–8.

Tech, M. et Merkl, R. (2003). Yacop : Enhanced gene prediction obtained by a combination of existing methods. *In Silico Biol*, **3**(4), 441–51.

Thanaraj, T. A., Stamm, S., Clark, F., Riethoven, J.-J., Le Texier, V., et Muilu, J. (2004). ASD : the Alternative Splicing Database. *Nucleic Acids Res*, **32**, D64–9.

Thébault, P. (2004). *Formalisme CSP et localisation de motifs structurés dans les textes génomiques.* Ph.D. thesis, Université P. Sabatier, Toulouse 3, France.

Tiwari, S., Ramachandran, S., Bhattacharya, A., Bhattacharya, S., et Ramaswamy, R. (1997). Prediction of probable genes by fourier analysis of genomic sequences. *Comput Appl Biosci*, **13**(3), 263–70.

Tolstrup, N. *et al.* (1997). A branch-point consensus from *Arabidopsis* found by non circular analysis allows for better prediction of acceptor sites. *Nucleic Acids Res.*, **25**, 3159–3163.

Tuteja, R. et Tuteja, N. (2004). Serial analysis of gene expression (sage) : unraveling the bioinformatics tools. *Bioessays*, **26**(8), 916–22.

Uberbacher, E. C. et Mural, R. J. (1991). Locating protein-coding regions in human dna sequences by a multiple sensor-neural network approach. *Proc Natl Acad Sci U S A*, **88**(24), 11261–5.

Usuka, J., Zhu, W., et Brendel, V. (2000). Optimal spliced alignment of homologous cDNA to a genomic DNA template. *Bioinformatics*, **16**(3), 203–211.

Velculescu, V. E., Zhang, L., Vogelstein, B., et Kinzler, K. W. (1995). Serial analysis of gene expression. *Science*, **270**(5235), 484–7.

Wang, J., Li, S., Zhang, Y., Zheng, H., Xu, Z., Ye, J., Yu, J., et Wong, G. K.-S. (2003). Vertebrate gene predictions and the problem of large genes. *Nat Rev Genet*, **4**(9), 741–9.

Watson, J. et Crick, F. (1953). A structure for deoxyribose nucleic acid. *Nature*, **171**, 737.

Wiehe, T., Gebauer-Jung, S., Mitchell-Olds, T., et Guigó, R. (2001). SGP-1 : prediction and validation of homologous genes based on sequence alignments. *Genome Res.*, **11**(9), 1574–83.

Woodley, L. et Valcarcel, J. (2002). Regulation of alternative pre-mrna splicing. *Brief Funct Genomic Proteomic*, **1**(3), 266–77.

Xu, Q., Modrek, B., et Lee, C. (2002). Genome-wide detection of tissue-specific alternative splicing in the human transcriptome. *Nucleic Acids Res*, **30**(17), 3754–66.

Xu, Y. et Uberbacher, E. (1997). Automated gene identification in large-scale genomic sequences. *J Comput Biol*, **4**(3), 325–38.

Xu, Y., Mural, R. J., et Uberbacher, E. C. (1994). Constructing gene models from accurately predicted exons : an application of dynamic programming. *Comput Appl Biosci*, **10**(6), 613–23.

Yada, T., Takagi, T., Totoki, Y., Sakaki, Y., et Takaeda, Y. (2003). Digit : a novel gene finding program by combining gene-finders. *Pac Symp Biocomput*, pages 375–87.

Yamada, K., Lim, J., Dale, J. M., Chen, H., Shinn, P., Palm, C. J., Southwick, A. M., Wu, H. C., Kim, C., Nguyen, M., Pham, P., Cheuk, R., Karlin-Newmann, G., Liu, S. X., Lam, B., Sakano, H., Wu, T., Yu, G., Miranda, M., Quach, H. L., Tripp, M., Chang, C. H., Lee, J. M., Toriumi, M., Chan, M. M. H., Tang, C. C., Onodera, C. S., Deng, J. M., Akiyama, K., Ansari, Y., Arakawa, T., Banh, J., Banno, F., Bowser, L., Brooks, S., Carninci, P., Chao, Q., Choy, N., Enju, A., Goldsmith, A. D., Gurjal, M., Hansen, N. F., Hayashizaki, Y., Johnson-Hopson, C., Hsuan, V. W., Iida, K., Karnes, M., Khan, S., Koesema, E., Ishida, J., Jiang, P. X., Jones, T., Kawai, J., Kamiya, A., Meyers, C., Nakajima, M., Narusaka, M., Seki, M., Sakurai, T., Satou, M., Tamse, R., Vaysberg, M., Wallender, E. K., Wong, C., Yamamura, Y., Yuan, S., Shinozaki, K., Davis, R. W., Theologis, A., et Ecker, J. R. (2003). Empirical analysis of transcriptional activity in the arabidopsis genome. *Science*, **302**(5646), 842–6.

Yeh, R., Lim, L., et Burge, C. (2001). Computational inference of homologous gene structures in the human genome. *Genome Res.*, **11**(5), 803–16.

Yeo, G. et Burge, C. B. (2004). Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. *J Comput Biol*, **11**(2-3), 377–94.

Yuan, Q., Ouyang, S., Liu, J., Suh, B., Cheung, F., Sultana, R., Lee, D., Quackenbush, J., et Buell, C. R. (2003). The tigr rice genome annotation resource : annotating the rice genome and creating resources for plant biologists. *Nucleic Acids Res*, **31**(1), 229–33.

Zavolan, M., Kondo, S., Schonbach, C., Adachi, J., Hume, D. A., Hayashizaki, Y., et Gaasterland, T. (2003). Impact of alternative initiation, splicing, and termination on the diversity of the mrna transcripts encoded by the mouse transcriptome. *Genome Res*, **13**(6B), 1290–300.

Zdobnov, E. M., von Mering, C., Letunic, I., Torrents, D., Suyama, M., Copley, R. R., Christophides, G. K., Thomasova, D., Holt, R. A., Subramanian, G. M., Mueller, H.-M., Dimopoulos, G., Law, J. H., Wells, M. A., Birney, E., Charlab, R., Halpern, A. L., Kokoza, E., Kraft, C. L., Lai, Z., Lewis, S., Louis, C., Barillas-Mury, C., Nusskern, D., Rubin, G. M., Salzberg, S. L., Sutton, G. G., Topalis, P., Wides, R., Wincker, P., Yandell, M., Collins, F. H., Ribeiro, J., Gelbart, W. M., Kafatos, F. C., et Bork, P. (2002). Comparative genome and proteome analysis of anopheles gambiae and drosophila melanogaster. *Science*, **298**(5591), 149–59.

Zhang, L., Pavlovic, V., Cantor, C. R., et Kasif, S. (2003). Human-mouse gene identification by comparative evidence integration and evolutionary analysis. *Genome Res*, **13**(6A), 1190–202.

Zhang, M. et Marr, T. (1993). A weight array method for splicing signal analysis. *Comput Appl Biosci.*, **9**(5), 499–509.

Zhang, M. Q. (1997). Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proc Natl Acad Sci U S A*, **94**(2), 565–8.

Zhang, M. Q. (2002). Computational prediction of eukaryotic protein-coding genes. *Nat Rev Genet*, **3**(9), 698–709.

Zhang, R. et Zhang, C. T. (1994). Z curves, an intutive tool for visualizing and analyzing the dna sequences. *J Biomol Struct Dyn*, **11**(4), 767–82.

Zhu, J., Shendure, J., Mitra, R. D., et Church, G. M. (2003a). Single molecule profiling of alternative pre-mrna splicing. *Science*, **301**(5634), 836–8.

Zhu, W., Schlueter, S., et Brendel, V. (2003b). Refined annotation of the Arabidopsis genome by complete expressed sequence tag mapping. *Plant Physiol.*, **132**(2), 469–84.

Zhu, Y. Y., Machleder, E. M., Chenchik, A., Li, R., et Siebert, P. D. (2001). Reverse transcriptase template switching : a smart approach for full-length cdna library construction. *Biotechniques*, **30**(4), 892–7.

Index